



**TESIS DOCTORAL**

**ESTRATEGIAS LÚDICAS PARA LA  
ENSEÑANZA DE LA PROGRAMACIÓN:  
UN ANÁLISIS COMPARATIVO DE SU EFICACIA  
EN LA EDUCACIÓN SUPERIOR**

**DIEGO PABLO CORSI**

**PROGRAMA DE DOCTORADO EN INNOVACIÓN EN FORMACIÓN DEL  
PROFESORADO.ASESORAMIENTO ANÁLISIS DE LA PRÁCTICA  
EDUCATIVA Y TIC EN EDUCACIÓN.**

**2019**





TESIS DOCTORAL

**ESTRATEGIAS LÚDICAS PARA LA  
ENSEÑANZA DE LA PROGRAMACIÓN:  
UN ANÁLISIS COMPARATIVO DE SU EFICACIA  
EN LA EDUCACIÓN SUPERIOR**

**DIEGO PABLO CORSI**

**PROGRAMA DE DOCTORADO EN INNOVACIÓN EN FORMACIÓN DEL  
PROFESORADO.ASESORAMIENTO ANÁLISIS DE LA PRÁCTICA  
EDUCATIVA Y TIC EN EDUCACIÓN.**

**Conformidad de los Directores:**

**Fdo: Dr. Francisco Ignacio Revuelta Domínguez**  
**Director**

**Fdo: Dra. María Inmaculada Pedrera Rodríguez**  
**Codirectora**

**2019**



*A Adriana*



# AGRADECIMIENTOS

---

Quisiera a través de estas líneas expresar mi agradecimiento a las personas que con su dedicación, apoyo y orientación me han acompañado durante la extraordinaria experiencia formativa que fue llevar a cabo las actividades del programa de doctorado y elaborar esta tesis.

Al Dr. Francisco I. Revuelta Domínguez y a la Dra. María Inmaculada Pedrera Rodríguez, por comprometerse con la dirección de este trabajo, aportando su inestimable asesoramiento y dedicándole su tiempo.

A mi esposa Adriana, quien me alentó y me ayudó durante todo el proceso.

A mi madre y mis hermanos, por su apoyo permanente, en especial a Julieta, quien desde España me dio una mano con los trámites administrativos de la Universidad de Extremadura.

A mi padre y mis abuelos, por seguir acompañándome y guiándome siempre.

A todos los estudiantes que participaron de esta investigación, ya sea cursando las materias que dicto en el Instituto Nacional Superior del Profesorado Técnico (Universidad Tecnológica Nacional), programando videojuegos, jugándolos, dando entrevistas o completando cuestionarios y encuestas.

A mis compañeros docentes e investigadores, por compartir conmigo sus opiniones y conocimientos. E incluyo aquí a todos aquellos que han sido mis maestros y profesores en algún momento de mi vida.

A las autoridades de las instituciones donde trabajo, por facilitar mi participación en eventos relacionados con la temática de esta investigación.

A todos ustedes, de todo corazón, ¡muchas gracias!





<b>Introducción .....</b>	<b>1</b>
Contextualización y planteamiento del problema de investigación .....	1
Justificación y objetivo del estudio .....	3
Estructura de la tesis .....	4

## **Primera Parte MARCO TEÓRICO**

<b>Capítulo 1: El Aprendizaje de la Programación .....</b>	<b>9</b>
1.1. Historia general de la programación y su aprendizaje .....	10
1.1.1. Décadas de 1950 y 1960: La programación llega a las universidades .....	10
1.1.2. Década de 1970: La programación llega a los hogares .....	15
1.1.3. Década de 1980: La programación llega a las escuelas .....	19
1.1.4. Década de 1990: Disminuye el entusiasmo por la programación .....	28
1.1.5. Década de 2000: Se priorizan otros usos de las computadoras .....	36
1.1.6. Década de 2010: La programación retorna a las aulas .....	43
1.2. La programación orientada a objetos .....	49
1.3. Razones para la importancia de aprender programación .....	53
1.3.1. El mercado laboral demanda programadores .....	53
1.3.2. La informática forma parte de nuestras vidas .....	54
1.3.3. Al programar se ejercita el pensamiento computacional .....	54
1.3.4. Aprender programación forma parte de la alfabetización computacional .....	55
1.3.5. Saber programación puede aumentar la productividad .....	55
1.3.6. Aprender programación puede ayudar a reducir las brechas digitales .....	56
1.4. Modelos didácticos para la programación .....	57
1.4.1. Escalera semiótica .....	57

1.4.2. Taxonomía de objetivos cognitivos .....	59
1.4.3. Resolución de problemas.....	61
1.5. Evaluación del aprendizaje de la programación.....	61
1.5.1. Pruebas objetivas .....	61
1.5.2. Trabajos para hacer en casa.....	62
1.5.3. Exámenes de respuestas cortas .....	63
1.5.4. Miniproyectos de laboratorio.....	63
1.6. Requisitos cognitivos para el aprendizaje de la programación.....	64
1.6.1. Capacidad matemática .....	64
1.6.2. Capacidad de procesamiento.....	65
1.6.3. Capacidad de aplicar el razonamiento analógico .....	65
1.6.4. Capacidad de aplicar el razonamiento condicional.....	66
1.6.5. Capacidad de aplicar el pensamiento procedimental.....	66
1.6.6. Capacidad de aplicar el razonamiento temporal.....	67
1.7. Obstáculos para el aprendizaje de la programación .....	68
1.7.1. Motivación inadecuada del estudiante.....	68
1.7.2. Estilo de aprendizaje inapropiado.....	69
1.7.3. Creencia del estudiante en su falta de aptitud .....	70
1.7.4. Ritmo del curso .....	72
1.7.5. Dificultad de la disciplina .....	72
<b>Capítulo 2: Las Relaciones Interpersonales en las Aulas.....</b>	<b>75</b>
2.1. La importancia de estudiar las relaciones humanas en las aulas .....	75
2.2. El test sociométrico.....	76
2.2.1. Trabajo preparatorio I: Elaboración del protocolo o ficha técnica .....	77
2.2.2. Trabajo preparatorio II: Elaboración del cuestionario.....	78
2.2.3. Trabajo preparatorio III: Motivación del grupo .....	79
2.2.4. Aplicación del cuestionario .....	81
2.2.5. Análisis de las respuestas I: Elaboración de la matriz sociométrica.....	82
2.2.6. Análisis de las respuestas II: Cálculo de los valores e índices sociométricos ...	83
2.2.7. Síntesis de los resultados I: Elaboración de los sociogramas .....	85

2.2.8. Síntesis de los resultados II: Elaboración del informe sociométrico.....	86
2.3. Validez del test sociométrico.....	86
<b>Capítulo 3: La Inteligencia Emocional .....</b>	<b>87</b>
3.1. La importancia de evaluar la inteligencia emocional .....	88
3.2. Algunos modelos teóricos de la inteligencia emocional.....	90
3.2.1. Modelos de Salovey y Mayer .....	90
3.2.2. Modelo de Petrides y Furnham .....	94
3.2.3. Otros modelos.....	99
3.3. Las competencias emocionales.....	101
<b>Capítulo 4: El Juego .....</b>	<b>103</b>
4.1. La naturaleza del juego .....	103
4.2. El estado de <i>flujo</i> .....	105
4.3. Los videojuegos .....	107
4.3.1. Definición del término <i>videojuego</i> .....	108
4.3.2. El origen de los videojuegos.....	112
4.3.3. Géneros de videojuegos.....	116
4.3.4. Los videojuegos como arte popular.....	122
4.4. Una taxonomía de los jugadores .....	124
4.5. La ludología .....	125
<b>Capítulo 5: El Aprendizaje mediante la Utilización de Videojuegos .....</b>	<b>131</b>
5.1. El <i>Aprendizaje Basado en Juegos Digitales</i> .....	133
5.1.1. Definición del <i>Aprendizaje Basado en Juegos Digitales</i> .....	134
5.1.2. Razones para la eficacia del <i>Aprendizaje Basado en Juegos Digitales</i> .....	136
5.1.3. Características de los buenos juegos digitales .....	138
5.2. Los <i>Juegos Serios</i> .....	140
5.2.1. Definición de <i>Juegos Serios</i> .....	141
5.2.2. Una taxonomía de los <i>Juegos Serios</i> .....	142
5.3. Tipos de juegos utilizados para aprender programación.....	144
5.3.1. Juegos no digitales.....	144

5.3.2. Juegos con componentes digitales y no digitales.....	144
5.3.3. Minijuegos orientados al aprendizaje de conceptos específicos.....	145
5.3.4. Videojuegos con personajes programables.....	146
5.3.5. Videojuegos del género <i>mundo abierto</i> .....	148
5.4. Impacto de la utilización de juegos en el aprendizaje de la programación.....	150
<b>Capítulo 6: El Aprendizaje mediante el Desarrollo de Videojuegos.....</b>	<b>151</b>
6.1. El construccionismo.....	152
6.2. Obstáculos para el aprendizaje mediante el desarrollo de videojuegos.....	153
6.2.1. Preferencia por el instruccionismo.....	154
6.2.2. Dificultades técnicas.....	155
6.2.3. Falta de interés de la industria.....	155
6.3. Beneficios del aprendizaje mediante el desarrollo de videojuegos.....	156
6.3.1. Aprendizaje de la programación.....	157
6.3.2. Aprendizaje de contenidos.....	158
6.3.3. Aprendizaje sobre el aprendizaje.....	159
6.3.4. Promoción del trabajo colaborativo.....	160
6.3.5. Reducción de las brechas de género y racial.....	161
6.4. Desarrollo de videojuegos sin escribir código.....	162
6.5. Formas de aprender programación mediante el desarrollo de videojuegos.....	164
6.5.1. Libros que enseñan lenguajes a través del desarrollo de videojuegos.....	164
6.5.2. Desarrollo de videojuegos usando lenguajes de propósito general.....	165
6.5.3. Herramientas de programación que facilitan la creación de videojuegos.....	168
<b>Capítulo 7: La Ludificación del Aprendizaje (<i>Gamification</i>).....</b>	<b>173</b>
7.1. Definición de la ludificación o <i>gamification</i> .....	174
7.2. Objetivos de la ludificación.....	176
7.2.1. Cambio de conductas.....	176
7.2.2. Desarrollo de habilidades.....	177
7.2.3. Impulso de la innovación.....	177
7.3. Elementos usados en la ludificación.....	178

7.4. Un marco de diseño para la ludificación .....	182
7.5. Tipos de ludificación.....	184
7.5.1. Ludificación interna vs. ludificación externa vs. ludificación social.....	184
7.5.2. Ludificación BLAP vs. ludificación significativa.....	185
7.6. Posibles problemas de la ludificación.....	186
7.6.1. Tendencia a la puntificación ( <i>pointsification</i> ).....	186
7.6.2. Uso como medio de explotación ( <i>exploitationware</i> ).....	187
7.7. Implementación de la ludificación para el aprendizaje .....	188
7.8. La ludificación del aprendizaje de la programación.....	189

## **Segunda Parte**

### **DESARROLLO EMPÍRICO**

<b>Capítulo 8: Metodología y Diseño de la Investigación .....</b>	<b>197</b>
8.1. Método.....	197
8.2. Diseño de la investigación.....	200
8.2.1. Primer tratamiento: Desarrollo de un videojuego serio .....	202
8.2.2. Segundo tratamiento: Uso de videojuegos serios .....	202
8.2.3. Tercer tratamiento: Ludificación del aprendizaje.....	204
8.3. Variables.....	205
8.4. Hipótesis .....	208
8.5. Muestras.....	209
8.5.1. Muestras para recolectar datos cuantitativos.....	209
8.5.2. Muestras para recolectar datos cualitativos .....	210
8.6. Instrumentos .....	212
8.6.1. Instrumentos para la recolección de datos .....	212
8.6.2. Instrumentos para el análisis de datos.....	216
<b>Capítulo 9: Resultados.....</b>	<b>217</b>
9.1. Análisis cuantitativo: Estadística descriptiva .....	217
9.2. Análisis cuantitativo: Contraste de hipótesis.....	222

9.2.1. Primera hipótesis (H1).....	223
9.2.2. Segunda hipótesis (H2).....	238
9.2.3. Tercera hipótesis (H3) .....	247
9.2.4. Cuarta hipótesis (H4) .....	250
9.2.5. Quinta hipótesis (H5).....	310
9.3. Análisis cualitativo: Categorización de percepciones.....	317
9.3.1. Adquisición de conocimientos de programación orientada a objetos.....	317
9.3.2. Relaciones interpersonales en el aula .....	323
9.3.3. Inteligencia emocional de los estudiantes .....	325
<b>Capítulo 10: Discusión y Conclusiones .....</b>	<b>329</b>
10.1. Discusión.....	329
10.1.1. Discusión sobre <i>el aprendizaje mediante la utilización de videojuegos</i> .....	329
10.1.2. Discusión sobre <i>el aprendizaje mediante el desarrollo de videojuegos</i> .....	330
10.1.3. Discusión sobre <i>la ludificación del aprendizaje/Gamification</i> .....	330
10.2. Conclusiones .....	331
10.2.1. Primera conclusión.....	331
10.2.2. Segunda conclusión.....	332
10.2.3. Tercera conclusión .....	333
10.2.4. Conclusión general del estudio .....	335
10.3. Principales aportaciones .....	335
10.4. Limitaciones del estudio .....	336
10.5. Prospectiva .....	338
<b>Referencias.....</b>	<b>341</b>
<b>ANEXOS</b>	
A1. TEIQue ( <i>Trait Emotional Intelligence Questionnaire</i> ).....	379
A2. Transcripción de las entrevistas con los estudiantes.....	383
<b>Índice .....</b>	<b>405</b>

# LISTA DE FIGURAS

<i>Figura 1:</i>	Diagrama de secuencia para un sistema de inscripción a un posgrado .....	50
<i>Figura 2:</i>	La clase <i>Boton</i> en UML y su utilización para crear los objetos <i>btnAbrir</i> , <i>btnGuardar</i> y <i>btnSalir</i> en Java .....	51
<i>Figura 3:</i>	Principales relaciones entre clases .....	52
<i>Figura 4:</i>	Tipos primitivos (izquierda), sus rangos y las clases de envoltorio correspondientes en Java (derecha) .....	53
<i>Figura 5:</i>	Escalera semiótica para analizar lenguajes de programación. ....	59
<i>Figura 6:</i>	Escala de la Distancia Social en el Aula.....	80
<i>Figura 7:</i>	Matriz sociométrica con los puntajes otorgados según la <i>Escala de la Distancia Social en el Aula</i> de Cunningham et al. (1951) .....	83
<i>Figura 8:</i>	Modelo de inteligencia emocional según Salovey y Mayer (1990).....	90
<i>Figura 9:</i>	Ejemplos de ítems del instrumento TMMS.....	91
<i>Figura 10:</i>	Modelo de inteligencia emocional según Mayer y Salovey (1997).....	92
<i>Figura 11:</i>	Ejemplos de ítems de percepción emocional del MSCEIT .....	93
<i>Figura 12:</i>	Ejemplos de ítems de manejo emocional del MSCEIT .....	94
<i>Figura 13:</i>	Modelo de inteligencia emocional según Petrides y Furnham .....	95
<i>Figura 14:</i>	El canal de flujo.....	107
<i>Figura 15:</i>	Una representación imprimible del modelo G/P/S .....	143
<i>Figura 16:</i>	Ciclo de sobreexpectación para las tecnologías emergentes para el año 2011 .....	173
<i>Figura 17:</i>	El marco de diseño 6D .....	183
<i>Figura 18:</i>	Diseño de la investigación.....	200
<i>Figura 19:</i>	Algunos de los videojuegos serios desarrollados por el primer grupo experimental y utilizados por el segundo grupo experimental .....	203
<i>Figura 20:</i>	Tabla de clasificación de los 12 equipos que constituyeron el tercer grupo experimental.....	204

<i>Figura 21:</i> Características de los estudiantes que participaron de la fase de recolección de datos cuantitativos .....	210
<i>Figura 22:</i> Características de los estudiantes que participaron de la fase de recolección de datos cualitativos .....	211
<i>Figura 23:</i> Escala de la Distancia Social en el Aula.....	213
<i>Figura 24:</i> Pretest de conceptos de programación orientada a objetos.....	214
<i>Figura 25:</i> Postest de conceptos de programación orientada a objetos .....	215
<i>Figura 26:</i> Guía de preguntas para estructurar las entrevistas .....	216
<i>Figura 27:</i> Adquisición de la capacidad de identificar las relaciones entre clases en UML (POO1).....	225
<i>Figura 28:</i> Adquisición de la capacidad de implementar las relaciones entre clases en Java (POO2).....	227
<i>Figura 29:</i> Adquisición de la capacidad de ordenar por rango los tipos primitivos en Java (POO3).....	229
<i>Figura 30:</i> Adquisición de la capacidad de nombrar las clases de envoltorio (POO4) .....	231
<i>Figura 31:</i> Adquisición de la capacidad de ordenar los calificadores de acceso que regulan el encapsulamiento en Java (POO5) .....	232
<i>Figura 32:</i> Adquisición de la capacidad de usar la abstracción para diseñar en UML un diagrama de clases que aplique la herencia (POO6) .....	234
<i>Figura 33:</i> Adquisición de la capacidad de describir la utilidad de la sobrecarga de constructores y métodos en Java (POO7) .....	236
<i>Figura 34:</i> Adquisición de la capacidad de reconocer e interpretar el polimorfismo en Java (POO8).....	238
<i>Figura 35:</i> Medias marginales estimadas de <i>Rango de POS-DSG</i> .....	247
<i>Figura 36:</i> Diagrama de dispersión general de <i>POS-Faceta Adaptabilidad</i> contra <i>PRE-Faceta Adaptabilidad</i> .....	252
<i>Figura 37:</i> Diagramas de dispersión de <i>POS-Faceta Adaptabilidad</i> contra <i>PRE-Faceta Adaptabilidad</i> separados según <i>Estrategia Lúdica</i> .....	252



<i>Figura 38:</i>	Diagrama de dispersión general de <i>POS-Faceta Asertividad</i> contra <i>PRE-Faceta Asertividad</i> .....	255
<i>Figura 39:</i>	Diagramas de dispersión de <i>POS-Faceta Asertividad</i> contra <i>PRE-Faceta Asertividad</i> separados según <i>Estrategia Lúdica</i> .....	255
<i>Figura 40:</i>	Diagrama de dispersión general de <i>POS-Faceta Autoestima</i> contra <i>PRE-Faceta Autoestima</i> .....	258
<i>Figura 41:</i>	Diagramas de dispersión de <i>POS-Faceta Autoestima</i> contra <i>PRE-Faceta Autoestima</i> separados según <i>Estrategia Lúdica</i> .....	258
<i>Figura 42:</i>	Diagrama de dispersión general de <i>POS-Faceta Automotivación</i> contra <i>PRE-Faceta Automotivación</i> .....	261
<i>Figura 43:</i>	Diagramas de dispersión de <i>POS-Faceta Automotivación</i> contra <i>PRE-Faceta Automotivación</i> separados según <i>Estrategia Lúdica</i> .....	261
<i>Figura 44:</i>	Diagrama de dispersión general de <i>POS-Faceta Conciencia Social</i> contra <i>PRE-Faceta Conciencia Social</i> .....	264
<i>Figura 45:</i>	Diagramas de dispersión de <i>POS-Faceta Conciencia Social</i> contra <i>PRE-Faceta Conciencia Social</i> separados según <i>Estrategia Lúdica</i> .....	264
<i>Figura 46:</i>	Medias marginales estimadas de <i>POS-Faceta Conciencia Social</i> .....	267
<i>Figura 47:</i>	Diagrama de dispersión general de <i>POS-Faceta Empatía</i> contra <i>PRE-Faceta Empatía</i> .....	271
<i>Figura 48:</i>	Diagramas de dispersión de <i>POS-Faceta Empatía</i> contra <i>PRE-Faceta Empatía</i> separados según <i>Estrategia Lúdica</i> .....	271
<i>Figura 49:</i>	Diagrama de dispersión general de <i>POS-Faceta Gestión de la Emoción</i> contra <i>PRE-Faceta Gestión de la Emoción</i> .....	278
<i>Figura 50:</i>	Diagramas de dispersión de <i>POS-Faceta Gestión de la Emoción</i> contra <i>PRE-Faceta Gestión de la Emoción</i> separados según <i>Estrategia Lúdica</i> .....	278
<i>Figura 51:</i>	Diagrama de dispersión general de <i>POS-Faceta Gestión del Estrés</i> contra <i>PRE-Faceta Gestión del Estrés</i> .....	281
<i>Figura 52:</i>	Diagramas de dispersión de <i>POS-Faceta Gestión del Estrés</i> contra <i>PRE-Faceta Gestión del Estrés</i> separados según <i>Estrategia Lúdica</i> .....	281

<i>Figura 53:</i>	Diagrama de dispersión general de <i>POS-Faceta Percepción Emocional</i> contra <i>PRE-Faceta Percepción Emocional</i> .....	286
<i>Figura 54:</i>	Diagramas de dispersión de <i>POS-Faceta Percepción Emocional</i> contra <i>PRE-Faceta Percepción Emocional</i> separados según <i>Estrategia Lúdica</i> .....	286
<i>Figura 55:</i>	Diagrama de dispersión general de <i>POS-Faceta Regulación Emocional</i> contra <i>PRE-Faceta Regulación Emocional</i> .....	289
<i>Figura 56:</i>	Diagramas de dispersión de <i>POS-Faceta Regulación Emocional</i> contra <i>PRE-Faceta Regulación Emocional</i> separados según <i>Estrategia Lúdica</i> .....	289
<i>Figura 57:</i>	Diagrama de dispersión general de <i>POS-Faceta Relaciones</i> contra <i>PRE-Faceta Relaciones</i> .....	292
<i>Figura 58:</i>	Diagramas de dispersión de <i>POS-Faceta Relaciones</i> contra <i>PRE-Faceta Relaciones</i> separados según <i>Estrategia Lúdica</i> .....	292
<i>Figura 59:</i>	Diagrama de dispersión general de <i>POS-Factor Autocontrol</i> contra <i>PRE-Factor Autocontrol</i> .....	295
<i>Figura 60:</i>	Diagramas de dispersión de <i>POS-Factor Autocontrol</i> contra <i>PRE-Factor Autocontrol</i> separados según <i>Estrategia Lúdica</i> .....	295
<i>Figura 61:</i>	Diagrama de dispersión general de <i>POS-Factor Bienestar</i> contra <i>PRE-Factor Bienestar</i> .....	298
<i>Figura 62:</i>	Diagramas de dispersión de <i>POS-Factor Bienestar</i> contra <i>PRE-Factor Bienestar</i> separados según <i>Estrategia Lúdica</i> .....	298
<i>Figura 63:</i>	Diagrama de dispersión general de <i>POS-Factor Emocionalidad</i> contra <i>PRE-Factor Emocionalidad</i> .....	301
<i>Figura 64:</i>	Diagramas de dispersión de <i>POS-Factor Emocionalidad</i> contra <i>PRE-Factor Emocionalidad</i> separados según <i>Estrategia Lúdica</i> .....	301
<i>Figura 65:</i>	Diagrama de dispersión general de <i>POS-Factor Sociabilidad</i> contra <i>PRE-Factor Sociabilidad</i> .....	304
<i>Figura 66:</i>	Diagramas de dispersión de <i>POS-Factor Sociabilidad</i> contra <i>PRE-Factor Sociabilidad</i> separados según <i>Estrategia Lúdica</i> .....	304
<i>Figura 67:</i>	Diagrama de dispersión general de <i>POS-Factor Global de IE</i> contra <i>PRE-Factor Global de IE</i> .....	307

<i>Figura 68:</i> Diagramas de dispersión de <i>POS-Factor Global de IE</i> contra <i>PRE-Factor Global de IE</i> separados según <i>Estrategia Lúdica</i> .....	307
<i>Figura 69:</i> Percepción de los conocimientos adquiridos en función de la estrategia lúdica empleada.....	321
<i>Figura 70:</i> Videojuego serio <i>The Java Mania Game</i> .....	322
<i>Figura 71:</i> Percepción de las relaciones en el grupo en función de la estrategia lúdica empleada.....	324
<i>Figura 72:</i> Percepción de las facetas de la IE en función de la estrategia lúdica empleada .....	327



## LISTA DE TABLAS

Tabla 1:	<i>Modalidades de uso de las computadoras en el aula</i> .....	22
Tabla 2:	<i>Estructura del Dominio Cognitivo (taxonomía de 1956) y de la dimensión Proceso Cognitivo (taxonomía de 2001)</i> .....	60
Tabla 3:	<i>Un conjunto de datos sobre el grupo que se deberían incluir en el protocolo de un test sociométrico</i> .....	77
Tabla 4:	<i>Partes del protocolo de un test sociométrico</i> .....	78
Tabla 5:	<i>Recomendaciones para la aplicación del cuestionario de un test sociométrico</i> .....	81
Tabla 6:	<i>Los principales valores sociométricos</i> .....	83
Tabla 7:	<i>Las partes del informe sociométrico</i> .....	86
Tabla 8:	<i>Principales instrumentos disponibles para la evaluación de la inteligencia emocional</i> .....	89
Tabla 9:	<i>Dominio muestral del modelo de Petrides y Furnham</i> .....	96
Tabla 10:	<i>Ítems característicos del TEIQue</i> .....	97
Tabla 11:	<i>Versiones del TEIQue disponibles</i> .....	99
Tabla 12:	<i>Modelos de inteligencia emocional mixtos</i> .....	100
Tabla 13:	<i>Principales teorías que buscan explicar la naturaleza, la génesis y el sentido del juego</i> .....	104
Tabla 14:	<i>Géneros de videojuegos según su interactividad</i> .....	117
Tabla 15:	<i>Una tipología multidimensional de juegos</i> .....	119
Tabla 16:	<i>Géneros de videojuegos según sus requerimientos para el éxito</i> .....	120
Tabla 17:	<i>Géneros de videojuegos utilizados en Google Play, Metacritic y GameSpot</i> .....	121
Tabla 18:	<i>Tipos de jugadores (Taxonomía de Bartle)</i> .....	125
Tabla 19:	<i>Carreras de formación en el desarrollo de videojuegos dictadas en la República Argentina</i> .....	128
Tabla 20:	<i>Líneas de trabajo de la Fundación Argentina de Videojuegos</i> .....	129
Tabla 21:	<i>Funciones esenciales del juego</i> .....	132

Tabla 22:	<i>Razones para la eficacia del Aprendizaje Basado en Juegos Digitales desde el punto de vista de los videojugadores</i> .....	137
Tabla 23:	<i>Elementos que siempre se encuentran presentes en los buenos videojuegos para un único jugador</i> .....	138
Tabla 24:	<i>Elementos que siempre se encuentran presentes en los buenos videojuegos para más de un jugador</i> .....	139
Tabla 25:	<i>Características de un buen juego</i> .....	140
Tabla 26:	<i>Principales herramientas actualmente disponibles para desarrollar videojuegos sin escribir código</i> .....	163
Tabla 27:	<i>Elementos del diseño de juegos y motivos</i> .....	179
Tabla 28:	<i>Elementos del diseño de juegos y descripciones según Ramírez Cogollor (2014)</i> .....	181
Tabla 29:	<i>Elementos del diseño de juegos y descripciones según Werbach y Hunter (2012)</i> .....	182
Tabla 30:	<i>RECIPE: Una receta para lograr la ludificación significativa</i> .....	186
Tabla 31:	<i>Niveles de implementación de la ludificación para el aprendizaje</i> .....	188
Tabla 32:	<i>Justificaciones y razonamientos para el uso de los métodos mixtos</i> .....	199
Tabla 33:	<i>Variables independientes utilizadas en este trabajo</i> .....	206
Tabla 34:	<i>Variables dependientes utilizadas en este trabajo</i> .....	206
Tabla 35:	<i>Codificación de las variables dependientes utilizadas en este trabajo</i> .....	207
Tabla 36:	<i>Características de los estudiantes que participaron de la fase de recolección de datos cuantitativos</i> .....	209
Tabla 37:	<i>Características de los estudiantes que participaron de la fase de recolección de datos cualitativos</i> .....	211
Tabla 38:	<i>Estadísticos descriptivos de las variables dependientes (categóricas dicotómicas)</i> .....	217
Tabla 39:	<i>Estadísticos descriptivos de las variables dependientes (cuantitativas continuas)</i> .....	218
Tabla 40:	<i>Tabla cruzada Estrategia Lúdica * POO1</i> .....	224
Tabla 41:	<i>Resumen de procesamiento de casos Estrategia Lúdica * POO1</i> .....	224

Tabla 42:	<i>Descriptores del tamaño del efecto según la V de Cramér.....</i>	225
Tabla 43:	<i>Tabla cruzada Estrategia Lúdica * POO2 .....</i>	226
Tabla 44:	<i>Tabla cruzada Estrategia Lúdica * POO3 .....</i>	228
Tabla 45:	<i>Resumen de procesamiento de casos Estrategia Lúdica * POO3.....</i>	228
Tabla 46:	<i>Tabla cruzada Estrategia Lúdica * POO4 .....</i>	230
Tabla 47:	<i>Resumen de procesamiento de casos Estrategia Lúdica * POO4.....</i>	230
Tabla 48:	<i>Tabla cruzada Estrategia Lúdica * POO5 .....</i>	232
Tabla 49:	<i>Tabla cruzada Estrategia Lúdica * POO6 .....</i>	233
Tabla 50:	<i>Resumen de procesamiento de casos Estrategia Lúdica * POO6.....</i>	234
Tabla 51:	<i>Tabla cruzada Estrategia Lúdica * POO7 .....</i>	235
Tabla 52:	<i>Tabla cruzada Estrategia Lúdica * POO8 .....</i>	237
Tabla 53:	<i>Resumen de procesamiento de casos Estrategia Lúdica * POO8.....</i>	237
Tabla 54:	<i>Supuestos para la prueba ANCOVA.....</i>	239
Tabla 55:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-DSP) .....</i>	241
Tabla 56:	<i>Estadísticos descriptivos de las variables Rango de PRE-DSP y Rango de POS-DSP .....</i>	242
Tabla 57:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS- DSP; Covariable: Rango de PRE-DSP) .....</i>	242
Tabla 58:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-DSG).....</i>	243
Tabla 59:	<i>Estadísticos descriptivos de las variables Rango de PRE-DSG y Rango de POS-DSG.....</i>	244
Tabla 60:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS- DSG; Covariable: Rango de PRE-DSG) .....</i>	244
Tabla 61:	<i>Descriptores del tamaño del efecto según el índice f de Cohen.....</i>	245
Tabla 62:	<i>Medias marginales de Rango de POS-DSG estimadas con base en la covariable Rango de PRE-DSG.....</i>	245
Tabla 63:	<i>Comparación de las medias marginales de Rango de POS-DSG entre las parejas formadas según Estrategia Lúdica .....</i>	246

Tabla 64:	<i>Prueba de Shapiro-Wilk para las diferencias entre el pretest y el posttest (relaciones interpersonales en el aula) .....</i>	248
Tabla 65:	<i>Contraste de las hipótesis referidas a las diferencias entre el pretest y el posttest (relaciones interpersonales en el aula).....</i>	248
Tabla 66:	<i>Descriptores del tamaño del efecto según los índices r y d de Cohen .....</i>	249
Tabla 67:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Adaptabilidad).....</i>	251
Tabla 68:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Adaptabilidad; Covariable: PRE-Faceta Adaptabilidad).....</i>	253
Tabla 69:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Asertividad).....</i>	254
Tabla 70:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Asertividad; Covariable: PRE-Faceta Asertividad).....</i>	256
Tabla 71:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Autoestima).....</i>	257
Tabla 72:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Autoestima; Covariable: PRE-Faceta Autoestima) .....</i>	259
Tabla 73:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Automotivación).....</i>	260
Tabla 74:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Automotivación; Covariable: PRE-Faceta Automotivación).....</i>	262
Tabla 75:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Conciencia Social).....</i>	263
Tabla 76:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Conciencia Social; Covariable: PRE-Faceta Conciencia Social) .....</i>	265
Tabla 77:	<i>Medias marginales de POS-Faceta Conciencia Social estimadas con base en la covariable PRE-Faceta Conciencia Social.....</i>	266
Tabla 78:	<i>Comparación de las medias marginales de POS-Faceta Conciencia Social entre las parejas formadas según Estrategia Lúdica.....</i>	267



Tabla 79:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Control de la Impulsividad).....</i>	268
Tabla 80:	<i>Estadísticos de Rango de PRE-Faceta Control de la Impulsividad y Rango de POS-Faceta Control de la Impulsividad.....</i>	269
Tabla 81:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Control de la Impulsividad; Covariable: Rango de PRE-Faceta Control de la Impulsividad) .....</i>	269
Tabla 82:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Empatía) .....</i>	270
Tabla 83:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Empatía; Covariable: PRE-Faceta Empatía) .....</i>	272
Tabla 84:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Expresividad Emocional) .....</i>	273
Tabla 85:	<i>Estadísticos de Rango de PRE-Faceta Expresividad Emocional y Rango de POS-Faceta Expresividad Emocional .....</i>	274
Tabla 86:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Expresividad Emocional; Covariable: Rango de PRE-Faceta Expresividad Emocional).....</i>	274
Tabla 87:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Felicidad).....</i>	275
Tabla 88:	<i>Estadísticos de Rango de PRE-Faceta Felicidad y Rango de POS-Faceta Felicidad.....</i>	276
Tabla 89:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Felicidad; Covariable: Rango de PRE-Faceta Felicidad) .....</i>	276
Tabla 90:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Gestión de la Emoción) .....</i>	277
Tabla 91:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Gestión de la Emoción; Covariable: PRE-Faceta Gestión de la Emoción) .....</i>	279
Tabla 92:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Gestión del Estrés).....</i>	280

Tabla 93:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Gestión del Estrés; Covariable: PRE-Faceta Gestión del Estrés)</i> .....	282
Tabla 94:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Optimismo)</i> .....	283
Tabla 95:	<i>Estadísticos de Rango de PRE-Faceta Optimismo y Rango de POS-Faceta Optimismo</i> .....	284
Tabla 96:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Optimismo; Covariable: Rango de PRE-Faceta Optimismo)</i> .....	284
Tabla 97:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Percepción Emocional)</i> .....	285
Tabla 98:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Percepción Emocional; Covariable: PRE-Faceta Percepción Emocional)</i> .....	287
Tabla 99:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Regulación Emocional)</i> .....	288
Tabla 100:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Regulación Emocional; Covariable: PRE-Faceta Regulación Emocional)</i> .....	290
Tabla 101:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Relaciones)</i> .....	291
Tabla 102:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Relaciones; Covariable: PRE-Faceta Relaciones)</i> .....	293
Tabla 103:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Autocontrol)</i> .....	294
Tabla 104:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Autocontrol; Covariable: PRE-Factor Autocontrol)</i> .....	296
Tabla 105:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Bienestar)</i> .....	297
Tabla 106:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Bienestar; Covariable: PRE-Factor Bienestar)</i> .....	299
Tabla 107:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Emocionalidad)</i> .....	300

Tabla 108:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Emocionalidad; Covariable: PRE-Factor Emocionalidad)</i> .....	302
Tabla 109:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Sociabilidad)</i> .....	303
Tabla 110:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Sociabilidad; Covariable: PRE-Factor Sociabilidad)</i> .....	305
Tabla 111:	<i>Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Global de IE)</i> .....	306
Tabla 112:	<i>Estadísticos de Rango de PRE-Factor Global de IE y Rango de POS-Factor Global de IE</i> .....	308
Tabla 113:	<i>Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Factor Global de IE; Covariable: Rango de PRE-Factor Global de IE)</i> .....	309
Tabla 114:	<i>Prueba de Shapiro-Wilk para las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)</i> .....	310
Tabla 115:	<i>Contraste de las hipótesis referidas a las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)</i> .....	313
Tabla 116:	<i>Percepción de los conocimientos adquiridos en función de la estrategia lúdica empleada</i> .....	318
Tabla 117:	<i>Percepción de las relaciones en el grupo en función de la estrategia lúdica empleada</i> .....	324
Tabla 118:	<i>Percepción de las facetas de la inteligencia emocional en función de la estrategia lúdica empleada</i> .....	326



El aprendizaje de la programación representa un gran desafío para los estudiantes de las carreras donde se dicta esta materia y también para los profesores responsables de su enseñanza. Afortunadamente, existen prácticas docentes innovadoras que pueden allanar el camino. Ciertas *estrategias lúdicas*, como el aprendizaje basado en juegos digitales (en particular, los juegos serios o *serious games*), el desarrollo de videojuegos como aplicación del *construccionismo* y, más recientemente, la ludificación del aprendizaje (*gamification*), abren perspectivas prometedoras para aumentar la *motivación* de los estudiantes y facilitar así la adquisición de los conceptos de esta disciplina.

En este proyecto se llevó a cabo un análisis de la eficacia de las estrategias lúdicas mencionadas, las cuales fueron implementadas durante el dictado de la asignatura *Programación II* de la Tecnicatura Superior en Informática Aplicada, en el Instituto Nacional Superior del Profesorado Técnico (*Universidad Tecnológica Nacional*), una institución de Educación Superior localizada en Buenos Aires, República Argentina.

Los resultados de esta investigación pretenden ser un aporte sobre la aplicación de métodos de enseñanza innovadores para satisfacer las necesidades de educación actuales.

**Palabras clave:** aprendizaje de la programación, estrategias lúdicas, análisis de eficacia, *serious games*, construccionismo, ludificación.



# ABSTRACT

---

Learning to program represents a great challenge for students taking courses where this subject is required and also for their teachers. Fortunately, there are innovative teaching practices that can pave the way. Certain *playful strategies*, such as Digital Game-Based Learning (in particular, *serious games*), the development of video games as an application of *constructionism* and, more recently, the *gamification* of learning, open up promising perspectives to increase student *motivation* and facilitate the acquisition of this subject's concepts.

This thesis' main goal is to present an analysis of the effectiveness of the aforementioned playful strategies, which were implemented during the teaching of the subject *Programming II* from the program in Applied Computing at the National Higher Institute for Technical Teachers' Education (*National Technological University*), a Higher Education institution located in Buenos Aires, Argentina.

The results of this research are intended to be a contribution to knowledge on the application of innovative teaching methods to meet current educational needs.

**Keywords:** programming learning, playful strategies, efficiency analysis, serious games, constructionism, gamification.





Lo único permanente es el cambio. Hace más de dos mil años, los filósofos griegos ya lo sabían. Platón escribió sobre ello en *Cratilo*: “Dice en algún lugar Heráclito . . . que no podrías entrar dos veces en el mismo río” (Mondolfo, 1966, p. 11). En las últimas décadas, el continuo surgimiento de innovaciones tecnológicas provocó que el mundo entero entrara en un vertiginoso proceso de globalización y digitalización que ha impactado en todos los ámbitos, y las prácticas pedagógicas no son una excepción.

## **Contextualización y planteamiento del problema de investigación**

En el contexto de la República Argentina, las necesidades provenientes de esta nueva realidad llevaron a que las sucesivas autoridades nacionales tomaran diversas medidas, como el lanzamiento de *educ.ar* (2000), portal educativo creado a través de la Resolución 441/2000 del Ministerio de Educación de la Nación, que aporta contenidos de las diversas áreas del conocimiento, con el propósito de promover la enseñanza y el aprendizaje de calidad; *Conectar Igualdad* (2010), programa de incorporación de la nueva tecnología para el aprendizaje de alumnos y docentes creado a través del Decreto 459/10, que ha capacitado a miles de docentes y entregado millones de computadoras a alumnos de todo el país; *Program.AR* (2014), iniciativa llevada adelante por la *Fundación Sadosky* —institución público-privada presidida por el Ministro de Ciencia, Tecnología e Innovación Productiva de la Nación— que busca que el aprendizaje significativo de las Ciencias de la Computación llegue a las escuelas argentinas; la promulgación de la *Ley 25.922 de Promoción de la Industria del Software* (2014); el dictado de la *Resolución 263/15 del Consejo Federal de Educación* (2015), que declaró que la enseñanza y el aprendizaje de la programación durante la escolaridad obligatoria son de importancia estratégica para el sistema educativo argentino; así como también el lanzamiento del *PNIDE - Plan Nacional de Inclusión*

*Digital Educativa* (2015) y el *PLANIED - Plan Nacional Integral de Educación Digital* (2016), iniciativas del Ministerio de Educación y Deportes de la Nación para integrar la comunidad educativa en la cultura digital, favoreciendo la innovación pedagógica, la calidad educativa y la inclusión socioeducativa.

Estas iniciativas, sin duda, constituyen un gran avance de la sociedad ante la creciente necesidad de incentivar el aprendizaje de la programación y el desarrollo del *pensamiento computacional* (Wing, 2006). En la práctica, sin embargo, la enseñanza y el aprendizaje de la programación representan un enorme desafío. Existe la percepción de que se requiere tener cierta *aptitud* para esta disciplina, y que no poseerla conducirá irremediablemente al fracaso. No obstante, algunos autores consideran que es precisamente la *creencia* de que se necesita una aptitud inherente para la programación lo que inhibe una práctica frecuente y deliberada de la escritura de código por parte de los estudiantes (Scott y Ghinea, 2013). Asimismo, T. Jenkins (2002) cuestiona la existencia de tal aptitud y afirma que si un estudiante tiende a adoptar estilos de aprendizaje erróneos o su motivación es errónea, aprender a programar le resultará difícil (p. 55).

Para aumentar la *motivación* de los estudiantes y facilitar el aprendizaje, en la actualidad existen prácticas docentes innovadoras que abren perspectivas muy promisorias, principalmente ciertas *estrategias lúdicas*, como el aprendizaje basado en juegos digitales (en particular, los juegos serios o *serious games*), el desarrollo de videojuegos como aplicación del *construccionismo* y, más recientemente, la ludificación del aprendizaje o *gamification*.

El problema es que, por diversas razones, resulta difícil encontrar análisis comparativos de la eficacia de estas estrategias cuando son aplicadas al aprendizaje de la programación en la Educación Superior.

Por un lado, el aprendizaje basado en juegos digitales es una estrategia que ya cuenta con una larga tradición en todos los niveles educativos. En particular, los juegos serios (es decir, juegos cuyo propósito principal no es divertir) han captado mucha atención recientemente, y su utilización en la Educación Superior ha sido objeto de numerosos estudios. No obstante, en cuanto a su aplicación para la enseñanza

de la programación, debe señalarse que, en la mayoría de los casos, el objetivo de esta estrategia es que los estudiantes adquieran apenas los conceptos *introdutorios* de la disciplina.

A su vez, el desarrollo de videojuegos como un caso particular de implementación de la teoría del *construccionismo* es algo frecuente tanto en la Educación Primaria como en la Secundaria, y los estudios al respecto son bastante numerosos. Sin embargo, el objetivo principal de este enfoque no es que los niños y adolescentes aprendan a programar en un lenguaje de propósito general (como C o Java), sino que utilicen los contenidos curriculares para crear productos concretos (videojuegos) y desarrollen, durante el proceso, el pensamiento computacional. En la Educación Superior, en cambio, el desarrollo de videojuegos obviamente es obligatorio en las carreras relacionadas de manera directa con la formación de profesionales para la industria de estos juegos, pero debido a que es marginal su presencia en otras carreras donde se enseña programación, los estudios sobre su eficacia como estrategia pedagógica para aprender esta disciplina no son tan abundantes.

Finalmente, cabe mencionar que, si bien la *ludificación* (es decir, el uso de elementos del diseño de juegos en situaciones no lúdicas) es un concepto bastante nuevo (su popularización recién se dio a partir de 2010), ya existen muchas publicaciones sobre su utilización, tanto en los sistemas educativos formales como en los ámbitos empresariales. Así y todo, sobre ella existen grandes controversias, pues esta estrategia no siempre es correctamente comprendida y con frecuencia se la implementa mal (Bogost, 2015).

### **Justificación y objetivo del estudio**

En esta investigación se llevó a cabo un análisis de la *eficacia* de las tres estrategias lúdicas recién mencionadas, al ser empleadas en la materia Programación II de la Tecnicatura Superior en Informática Aplicada, carrera que se estudia en el Instituto Nacional Superior del Profesorado Técnico (*Universidad Tecnológica Nacional*), una institución de Educación Superior localizada en la Ciudad Autónoma de Buenos Aires, República Argentina.

Para ello, se planteó el objetivo de este estudio, que fue evaluar el impacto que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tiene sobre las siguientes dimensiones:

- la adquisición de los conocimientos específicos de la disciplina;
- las relaciones interpersonales en el aula;
- la inteligencia emocional de los estudiantes.

Cabe mencionar que la medición del impacto que la implementación de estrategias lúdicas tiene sobre las dos últimas dimensiones anteriores, por lo general, no está incluida en los sistemas de evaluación tradicionales, que privilegian la medición de la adquisición de los conocimientos específicos de las disciplinas.

Con esta investigación se pretendió hacer un aporte sobre la aplicación de métodos de enseñanza innovadores para satisfacer las necesidades de educación actuales, ya que se investigó empíricamente el potencial de las estrategias lúdicas como un recurso para la mejora de la enseñanza de la programación en la Educación Superior.

## **Estructura de la tesis**

La tesis está estructurada en dos grandes partes: el marco teórico y el desarrollo empírico. A continuación se describe brevemente el contenido de cada una de estas partes.

El marco teórico tiene como principales propósitos explicar el origen y el alcance del problema de investigación, conocer los enfoques aplicados previamente y proporcionar un encuadre de conocimiento para situar los objetivos, los resultados y las conclusiones. Está dividido en siete capítulos. Los tres primeros abarcan los conceptos que se han medido en el trabajo, en el cuarto se trata el tema que constituye la base para las estrategias lúdicas y, finalmente, estas últimas se presentan en los tres capítulos posteriores.

En el capítulo 1, titulado *El Aprendizaje de la Programación*, se comenta la historia general de esta disciplina y se mencionan sus características principales, así como también los beneficios que ofrece y las dificultades que presentan su aprendizaje y su evaluación.

En el capítulo 2, titulado *Las relaciones interpersonales en las aulas*, se aborda la importancia de este tema y se explica el test sociométrico, incluyendo el cálculo de los distintos índices sociométricos y la discusión sobre la validez de este instrumento.

En el capítulo 3, titulado *La inteligencia emocional*, se define este concepto proveniente de la psicología, se mencionan diferentes modelos surgidos para explicarlo y se trata su evaluación.

En el capítulo 4, titulado *El Juego*, se detallan los conceptos más importantes de esta temática, como la naturaleza del juego según diversas teorías, el estado de *flujo*, los videojuegos, los distintos tipos de jugadores y la disciplina que se conoce como *ludología*.

En el capítulo 5, titulado *El Aprendizaje mediante la Utilización de Videojuegos*, se describe esta estrategia y se comentan los estudios precedentes referidos tanto al aprendizaje basado en juegos como a los *serious games*, en general, y a su aplicación para el aprendizaje de la programación, en particular.

En el capítulo 6, titulado *El Aprendizaje mediante el Desarrollo de Videojuegos*, se explica esta estrategia lúdica basada en la teoría del construccionismo y se mencionan los antecedentes de su implementación para el aprendizaje de la programación.

En el capítulo 7, titulado *La Ludificación del Aprendizaje (Gamification)*, se presenta detalladamente esta estrategia y se señalan las investigaciones previas sobre su empleo en el aprendizaje de la programación.

El desarrollo empírico abarca todo el proceso metodológico de la investigación, desde el planteamiento de la metodología adoptada hasta la presentación y discusión de los resultados. Está dividido en tres capítulos.

En el capítulo 8, titulado *Metodología y Diseño de la Investigación*, se explica la metodología empleada, se define el diseño de la investigación, se identifican las variables, se enuncian las hipótesis del trabajo, y se establecen las muestras y los instrumentos utilizados para la recolección de los datos.

En el capítulo 9, titulado *Resultados*, se compendia la información producida mediante los análisis descriptivos e inferenciales de los datos recabados.

En el capítulo 10, titulado *Discusión y Conclusiones*, se discuten los resultados de la investigación a la luz de los estudios previos mencionados en el marco teórico, se presentan las conclusiones de la investigación y sus principales aportaciones, sus limitaciones y una prospectiva (futuras líneas de investigación).

Por último, se incluyen la bibliografía, los anexos (donde se recopilan —entre los elementos utilizados en esta investigación— las transcripciones de las entrevistas realizadas con los estudiantes) y el índice.

**PRIMERA PARTE**  
**MARCO TEÓRICO**

---





# EL APRENDIZAJE DE LA PROGRAMACIÓN

---

Después de estar, durante varios años, restringida a los planes de estudios de la Educación Superior y prácticamente ausente en la Educación Primaria y Secundaria —donde, con la popularización de la Internet, las computadoras habían dejado de tener otro uso más que el de publicar o conseguir información (Kafai y Burke, 2014, p. xi)—, en la segunda década del siglo XXI la programación pasó a ser considerada *el nuevo latín* (Cellan-Jones, 2011). Una evidencia de ello es, por ejemplo, la publicación del informe *Next Gen* (Livingstone y Hope, 2011) por Nesta (*National Endowment for Science, Technology and the Arts*), donde se indicaba que, aprendiendo a *usar* el *software* pero no a *crearlo*, los estudiantes no estaban desarrollando las competencias necesarias para mantener al Reino Unido como un líder en el mercado de los videojuegos y los efectos especiales.

Desde entonces, numerosos países han reconocido la importancia de comenzar el aprendizaje de la programación a partir de la Educación Primaria y han empezado a implementar currículos en los que las ciencias de la computación, en general, y la programación, en particular, ocupan un lugar destacado. Ha quedado claro que para triunfar en el mercado laboral del siglo XXI no alcanza con dominar los paquetes de ofimática: se requieren, además, nuevas competencias básicas como, entre otras, saber crear contenido y publicarlo en una página web.

En España, por ejemplo, la Comunidad de Madrid empezó a incorporar en la ESO (*Educación Secundaria Obligatoria*) a partir del curso 2014/2015 una nueva asignatura obligatoria: *Programación y Comunicaciones* (Serrano, 2014).

En la República Argentina, el gobierno nacional declaró, a través de la *Resolución 263/15 del Consejo Federal de Educación* (2015), que la enseñanza y el aprendizaje de la programación durante la escolaridad obligatoria son de importancia estratégica para el sistema educativo del país.

En los Estados Unidos, la ACM (*Association for Computing Machinery*) publicó los informes *Running On Empty: The Failure to Teach K–12 Computer Science in the Digital Age* (Wilson, Sudol, Stephenson y Stehlik, 2010) y *Rebooting the Pathway to Success: Preparing Students for Computing Workforce Needs in the United States* (Kaczmarczyk y Dopplick, 2014) advirtiendo sobre la urgente necesidad de reformar la Educación Primaria y la Educación Secundaria para que los estudiantes adquieran las competencias y el pensamiento computacional requeridos para poder desempeñarse con éxito en el nuevo contexto laboral desencadenado por la globalización y la digitalización. Esto llevó a que el congreso estadounidense promulgara, en 2015, la *STEM Education Act* a través de la cual se incorporaron las ciencias de la computación al conjunto STEM (*Science, Technology, Engineering, and Mathematics*), áreas que reciben diversos incentivos del gobierno federal (Guzdial y Morrison, 2016, p. 31) y a que el poder ejecutivo anunciara, en 2016, la iniciativa *Computer Science For All* (M. Smith, 2016), destinando fondos para posibilitar la inclusión de esta disciplina en todos los niveles del sistema educativo y convocando al país entero a comprometerse con el cambio.

### **1.1. Historia general de la programación y su aprendizaje**

Es indudable que el aprendizaje de la programación —incluso a partir de la Educación Primaria— ha pasado a tener, durante la última década, una enorme importancia en todo el mundo. Sin embargo, ya hace más de medio siglo que esta disciplina empezó a enseñarse en los sistemas educativos. A continuación se recopilan los principales antecedentes de esta temática, agrupados en seis periodos bien diferenciados.

#### **1.1.1. Décadas de 1950 y 1960: La programación llega a las universidades**

Mientras que, en la primera mitad de la década de 1940, para programar una computadora —generalmente electromecánica— era necesario conmutar físicamente sus circuitos, en los últimos años de esa misma década las computadoras electrónicas pasaron a almacenar los programas de la misma forma en que almacenaban los datos. Es por ello que, a partir de ese momento, el concepto de *programación* cambió: dejó de ser un concepto físico de ingeniería para volverse una representación simbólica. Por *programación* pasó a entenderse la manipulación de *código* (Vee, 2013, p. 48).

Los primeros cursos de programación dictados en ambientes académicos tuvieron lugar en la década de 1950, cuando las computadoras empezaron a estar disponibles comercialmente —aunque eran tan escasas y caras que solo grandes organizaciones podían adquirirlas—. Se trataba principalmente de cursos destinados a estudiantes universitarios de carreras afines a las ciencias exactas y las tecnologías. En los Estados Unidos, por ejemplo, el primer curso de programación ofrecido por el MIT (*Massachusetts Institute of Technology*) se dictó en 1951 y estuvo a cargo del matemático Gordon Welchman (Hafner y Lyon, 1998, p. 88), quien durante la Segunda Guerra Mundial había dirigido los esfuerzos británicos para romper el código Enigma utilizado por los alemanes en los ejércitos y la fuerza aérea.

Una década más tarde, en 1961, en la República Argentina se puso en marcha *Clementina*, la primera computadora instalada en el país para apoyar la investigación científica (Babini y Vilensky, 2003, p. 26), a la cual se pensaba darle “aplicaciones que fueran de utilidad para el país en muy diversas áreas: . . . para orientar y ordenar la administración pública; para impulsar la investigación operativa, . . . para diseñar planes y estrategias” (Sadosky, 2011, p. 113). Adquirida en el Reino Unido por la *Universidad de Buenos Aires* para su Facultad de Ciencias Exactas y Naturales, *Clementina* era una computadora Ferranti Mercury II, cuya programación se realizaba con un lenguaje ensamblador conocido como *Pig-2* y con un lenguaje de alto nivel —el primero de la historia— denominado *Autocode* (Factorovich y Jacovkis, 2009, p. 89), con el que se iniciaron las primeras generaciones de programadores argentinos (Aguirre, 2009, p. 27). El primer curso público de programación fue dictado por la doctora Cicely Popplewell, de la *University of Manchester* (Carnota y Borches, 2015, p. 114). Luego, cuando Popplewell regresó a su país, toda la actividad docente de programación de computadoras recayó en el matemático español Ernesto García Camarero (Jacovkis, 2011, p. 3).

Hasta ese momento, en realidad, existían pocos lenguajes de programación de alto nivel que fueran independientes de determinado modelo de computadora. Básicamente, si alguien quería aprender a programar en un lenguaje que estuviera disponible para computadoras de diferentes fabricantes, sus opciones se reducían al FORTRAN

—destinado a aplicaciones científicas y de ingeniería, y cuyo primer compilador estuvo listo en 1957—, el ALGOL —diseñado para facilitar la comunicación de métodos numéricos entre personas y llevarlos a cabo en diversas máquinas, e implementado en 1958—, el LISP —destinado al procesamiento de expresiones simbólicas, y cuyo intérprete estuvo disponible en 1959— y el COBOL —orientado a los negocios e implementado en 1960— (Sebesta, 2012, pp. 42-63).

El hecho de que solamente construyera 19 unidades del modelo Mercury (Carnota y Pérez, 2009, p. 127) llevó a que, en poco tiempo, los programadores de *Clementina* y los estudiantes que aprendían con ella vivieran en “un ambiente de aislamiento . . . ya que los programadores del resto del mundo usaban FORTRAN o COBOL para el desarrollo de sus aplicaciones” (Durán, Zoltan, Lew, Cortés y García, 2009, p. 109).

Mientras tanto, en 1961, en el *MIT* tuvo lugar el simposio *Computers and the World of the Future*. Allí, Alan Perlis, quien formaba parte del equipo que había desarrollado el ALGOL y era uno de los oradores, explícitamente defendió que debía enseñarse programación a todo el mundo, en todas las facultades de todas las universidades, para que todos tuvieran una notación para especificar procesos y la potencia para automatizarlos (Guzdial, 2016, p. 37).

En 1962, Fernando Corbató y sus colaboradores del *MIT* presentaron en la *Spring Joint Computer Conference* el sistema operativo de tiempo compartido CTSS (*Compatible Time Sharing System*) (Corbató, Merwin-Daggett y Daley, 1962). Hasta entonces, “todas la computadoras eran personales, por lo menos en el sentido de que . . . solo una persona (el programador) podía usarlas en un momento dado” (Tanenbaum, 2009, p. 720). Con CTSS, una única computadora podía ser utilizada simultáneamente por varios usuarios, cada uno accediendo a ella desde su propia terminal.

Posteriormente, en el *Dartmouth College* se desarrollaron en 1964 el sistema operativo de tiempo compartido DTSS (*Dartmouth Time-Sharing System*) y el lenguaje de programación BASIC (*Beginner's All-purpose Symbolic Instruction Code*). John Kemeny y Thomas Kurtz, los líderes de este último proyecto, defendían el conocimiento sobre las computadoras y su programación como una parte esencial de la educación liberal:

Los estudiantes de ciencias e ingenierías obviamente lo necesitan para poder llevar a cabo su trabajo. Pero la exposición a las computadoras, su programación, su potencia y sus limitaciones debe extenderse a todos los demás estudiantes, ya que muchos futuramente tendrán cargos de decisión en la industria y en el gobierno (Kemeny y Kurtz, 1968, p. 223).

Según estos mismos autores, durante el periodo 1967-1968 el sistema DTSS tuvo un total de 8150 usuarios, de los cuales 4600 pertenecían a 23 escuelas secundarias de la región (p. 226). Afirmaban, además, que en lo que concierne al usuario, la máquina habla BASIC, ya que el código de máquina es tan irrelevante para el usuario promedio como lo son las peculiaridades del *hardware* (p. 227) y que para entonces su biblioteca contenía alrededor de 500 programas, de los cuales muchos eran juegos. Su entusiasmo con estos últimos queda explícito en el siguiente relato:

Hemos descubierto que una computadora es una fuente de entretenimiento eficiente y de bajo costo. Muchos visitantes distinguidos se han quedado atrapados durante varias horas dirigiendo al equipo de fútbol americano de Dartmouth en un juego simulado de manera altamente realista (p. 228).

En 1965, la ACM —que había sido fundada en 1947 y para entonces ya era la mayor asociación científica y educativa internacional dedicada a la computación— publicó un informe en el que proponía un currículo para la formación de grado en ciencias de la computación, cuyas actualizaciones han servido desde entonces para definir los contenidos de esta disciplina enseñados en todo el mundo. El hecho de que la mayoría de las carreras existentes en aquel momento se había originado en departamentos de matemáticas o en facultades de ingeniería tuvo una clara influencia en la orientación de este currículo, pues su meta era preparar a los estudiantes para estudios de posgrado en ciencias de la computación o en otros campos, para la *programación de sistemas* y para la *programación de aplicaciones*, y con este fin tenía un pequeño núcleo obligatorio y un conjunto de materias optativas. El núcleo obligatorio se complementaba con el requerimiento de cinco cursos de matemáticas (cálculo, análisis, álgebra lineal) —a los que era altamente recomendable agregarles más matemáticas— y con optativas en campos relacionados, como la electrónica (Hemmendinger, 2007, p. 47).

En 1966, tras el golpe de estado que dio inicio a una dictadura militar en la República Argentina, las universidades nacionales fueron intervenidas, lo que llevó a que

numerosos docentes renunciaran a sus cargos. En la *Universidad de Buenos Aires*, esto tuvo un fuerte impacto sobre la enseñanza de la programación:

El vacío . . . fue cubierto por funcionarios de IBM, que convirtieron . . . la carrera en una especie de filial del centro de formación de la empresa. El paradigma fue la materia *Programación* cuyo contenido fue, por varios años, el aprendizaje de la cartilla de instrucciones del *assembler* del equipo IBM 360 . . . y del lenguaje FORTRAN, . . . dictado siguiendo un esquema de manual . . . Sin mediar una reforma curricular, lo que se multiplicó fue la oferta de materias optativas de programación de distintos lenguajes, en un sentido de mera formación técnica, sin la enseñanza de los fundamentos (Carnota, Factorovich y Pérez, 2009, p. 154).

También en 1966, el matemático Seymour Papert —quien por aquel entonces trabajaba en el *MIT* tras haber sido discípulo de Jean Piaget en Ginebra— se unió al ETD (*Educational Technology Department*) fundado y liderado por Wally Feurzeig en la empresa de alta tecnología BBN (*Bolt, Beranek and Newman, Inc.*), donde junto con Daniel Bobrow idearon Logo, un lenguaje de programación específicamente diseñado para la enseñanza de las matemáticas (Feurzeig, Papert, Bloom, Grant y Solomon, 1969, p. 1). Según Papert (1980), era tan inaceptable que los niños entraran a la cultura informática aprendiendo lenguajes como el BASIC, como lo sería que su aprendizaje de la poesía inglesa ocurriera a través de traducciones del *pidgin* (p. 211).

Entre 1967 y 1970, Logo fue utilizado en varias escuelas de Massachusetts para probar métodos experimentales de enseñanza en niños a partir de 7 años. Feurzeig (2011) señala que a los niños les resultaba natural formalizar procedimientos mediante el Logo:

Los niños comenzaron su trabajo con Logo utilizando procedimientos con los cuales estaban familiarizados: . . . crear y romper códigos secretos . . ., juegos de palabras . . . y juegos de adivinanzas . . . Al principio, los niños piensan que entienden perfectamente estos problemas porque . . . pueden hacer descripciones verbales vagas de sus procedimientos. Pero les resulta imposible hacer estas descripciones de forma precisa y general, en parte por falta de hábitos formales de pensamiento, y en parte por falta de un lenguaje adecuadamente expresivo. Usando Logo, el proceso de transformar descripciones verbales vagas en otras formales y precisas se hace posible y, en este contexto, a los niños les parece natural y agradable (p. 291).

En 1970, Papert fundó el *Logo Laboratory* en el *MIT*, donde se llevó a cabo la mayoría del desarrollo posterior del lenguaje (Papert, 1980, p. 211). La primera tortuga *inalámbrica* conducida por la habitación mediante comandos en Logo fue presentada por

BBN en 1972 y, ese mismo año, también fueron creadas en el *MIT* las tortugas gráficas que podían moverse por la pantalla para dibujar (Feurzeig, 2011, p. 293).

### 1.1.2. Década de 1970: La programación llega a los hogares

Durante esta década ocurrieron grandes avances tanto en el *hardware* como en el *software* de las computadoras, lo que permitió que por primera vez la programación estuviera disponible para el público general. Fue también un periodo en el que se reconoció la importancia de enseñar la programación como una disciplina sistemática.

En 1968, Edsger Dijkstra había afirmado que la calidad de los programadores es una función decreciente de la densidad instrucciones *goto* en los programas que producen (Dijkstra, 1968, p. 147). Durante la primera Conferencia sobre Ingeniería del Software de la OTAN, que tuvo lugar ese mismo año en Alemania, se habló de una *crisis del software*. Dijkstra fue uno de los participantes y advirtió que la difusión del conocimiento es de valor obvio, pues la difusión masiva de *software* cargado de errores es aterradora (NATO Science Committee, 1969, p. 16). Más tarde, en 1975, llegó incluso a afirmar que es prácticamente imposible enseñar programación correctamente a estudiantes que hayan estado expuestos al lenguaje BASIC con anterioridad. Como potenciales programadores, tienen la mente mutilada sin esperanza alguna de regeneración (Dijkstra, 1982, p. 130).

Por ello, en 1970 Niklaus Wirth —basándose en ALGOL— presentó el lenguaje Pascal, diseñado e implementado en la ETH (*Eidgenössische Technische Hochschule*) de Zürich con dos objetivos principales: ofrecer un lenguaje adecuado para enseñar programación como una disciplina sistemática basada en ciertos conceptos fundamentales reflejados clara y naturalmente en el lenguaje y desarrollar implementaciones de este lenguaje que sean confiables y eficientes en las computadoras disponibles en aquel momento (Wirth, 1971, p. 35).

El éxito de Pascal se debió, en parte, a la disponibilidad de un manual del usuario que incluía material tutorial además de la definición del lenguaje, pronto acompañado por un número creciente de libros de texto introductorios de autores de muchos países (Wirth, 1996, p. 107). Su impacto en la enseñanza de la programación fue profundo y duradero, como narra Robert Sebasta (2012):

En 1970, la mayoría de los estudiantes de informática, ingeniería y ciencias aprendían programación con FORTRAN, aunque algunas universidades usaban PL/I y ALGOL-W. A mediados de la década de 1970, Pascal se había convertido en el lenguaje más utilizado para este propósito. Esto era bastante natural, dado que había sido diseñado específicamente para la enseñanza de la programación. No fue sino hasta finales de la década de 1990 que Pascal dejó de ser el lenguaje más utilizado para la enseñanza de la programación en la Educación Superior (p. 76).

Además, el desarrollo de *software* siguiendo el *modelo en cascada* —cuyos pasos son la definición y el análisis de los requerimientos, seguidos del diseño, la programación, la prueba y el mantenimiento del sistema— se volvió muy popular después de la publicación del artículo *Managing the Development of Large Software Systems* (Royce, 1970).

La ACM-ICPC (*International Collegiate Programming Contest*), que se destaca por ser la más antigua, grande y prestigiosa competición mundial de programación («The ACM-ICPC», 2017) es descendiente directa de un torneo de programación celebrado en 1970 en la *Texas A&M University*. Desde entonces, cada año miles de estudiantes universitarios de todo el mundo estudian y se entrenan en temas avanzados de programación para representar a sus instituciones y países en este prestigioso evento.

Hasta 1971, la enseñanza de la programación en la *Universidad de Buenos Aires* permaneció bajo la hegemonía de IBM. Ese año, una huelga total de los estudiantes de la materia *Programación* llevó a que se cambiara el contenido de la materia “haciéndolo más general, para que posibilitara no solo utilizar equipos de fabricación IBM sino de otras marcas” (Carnota et al., 2009, p. 158). Así, se pasó a “dictar los fundamentos teóricos de la programación y a realizar las prácticas con un lenguaje abstracto respecto a cualquier máquina particular” (p. 163).

Ese mismo año, se inició en la *Université de Aix-Marseille* el desarrollo del lenguaje Prolog, que nació de un proyecto dirigido no a producir un lenguaje de programación sino a procesar lenguajes naturales; en este caso, el francés. El proyecto dio lugar a una versión preliminar de Prolog a fines de 1971 y una versión más definitiva a fines de 1972 (Colmerauer y Roussel, 1996, p. 331). El lenguaje Prolog permite escribir programas dentro del paradigma de la *programación lógica* o declarativa.



En 1971, Alan Kay dio inicio al LRG (*Learning Research Group*) en el *PARC* (*Palo Alto Research Center*), el centro de investigación que Xerox había fundado el año anterior. Kay estaba convencido de que algunos aspectos del proceso de aprendizaje se pueden aumentar mediante medios tecnológicos (Kay, 1972) y conocía muy bien las limitaciones de los sistemas de tiempo compartido para educar niños:

Las mejores imágenes que el tiempo compartido puede proporcionar son crudos dibujos de líneas de color verde y tonos musicales de onda cuadrada. Los niños, sin embargo, están acostumbrados a pintar con los dedos, a la televisión en color y a las grabaciones estereofónicas, y las cosas que pueden lograrse con un sistema de tiempo compartido de baja capacidad usualmente les parecen insuficientemente estimulantes para mantener su interés (Kay, 1977).

Por ello, su objetivo principal era construir el *Dynabook*, un dispositivo definido por él años atrás en su tesis de doctorado y para el cual dio más tarde la siguiente descripción:

El Dynabook será un *libro* mejor, uno activo (como los niños) en vez de pasivo. Puede ser algo con el mismo poder de captar la atención que tiene la TV, pero controlable por el niño en lugar de las emisoras. Puede ser como un piano (un producto de la tecnología, sí), pero que también sea una herramienta, un juguete, un medio de expresión y una fuente inagotable de placer (Kay, 1972).

No obstante, para Xerox la prioridad era la creación de la oficina del futuro (Hiltzik, 1999, p. 29) y no un sistema dirigido al juego infantil (p. 165). Debido a ello, Kay debió dejar de lado el sueño del *Dynabook* y participar de los esfuerzos que en aquel momento, en el *PARC*, estaban volcados hacia el desarrollo de la *computación personal*.

La computación personal había surgido en el *MIT* en 1961, cuando Wesley Clark y Charles Molnar diseñaron la LINC (*Laboratory INSTRument Computer*), una computadora relativamente pequeña, pensada para ser utilizada por un único usuario a la vez —no era un sistema de tiempo compartido— y orientada al trabajo con experimentos de laboratorio para biomedicina. La LINC fue luego distribuida y modificada por DEC (*Digital Equipment Corporation*), gracias a lo cual fueron comercializadas —entre originales y derivadas— unas 1200 unidades en todo el mundo (Clark, 1986, p. 133).

El impacto de los trabajos realizados en el *PARC* durante la década de 1970 llega hasta los días de hoy. Los científicos del *PARC* crearon más que una computadora

personal. Diseñaron, construyeron y utilizaron un sistema completo de *hardware* y *software* que revolucionó la computación. Además de la computadora Alto, los inventores del PARC hicieron el primer monitor orientado a gráficos, el primer *mouse* suficientemente simple para un niño, el primer programa de procesamiento de textos para usuarios no expertos, la primera red de comunicaciones de área local, el primer lenguaje de programación orientada a objetos y la primera impresora láser (D. K. Smith y Alexander, 1988, p. 14).

La computadora Alto a que se refieren los autores anteriores tuvo una importancia fundamental para el trabajo de Kay, ya que funcionó como un *Dynabook interino* (Kay, 1996, p. 533). Sebasta (2012) recuerda que una serie de proyectos de investigación se llevaron a cabo con este sistema, incluyendo varios experimentos para enseñar la programación a los niños (p. 86) y destaca que Smalltalk, el lenguaje desarrollado por el equipo de Kay para la programación de la computadora Alto, fue el primer lenguaje de programación que apoyó completamente la *programación orientada a objetos* —un paradigma restringido desde hacía años a los ambientes académicos— (p. 85).

A pesar de sus extraordinarias características, la computadora Alto nunca fue considerada por Xerox más que un proyecto interno y, en consecuencia, no se comercializó masivamente. Solo se construyeron alrededor de 2000 unidades (Kay, 1996, p. 535).

Sin embargo, otras empresas sí vieron el enorme negocio que significaría la difusión masiva de la computación personal, disponible a un costo accesible, algo que ocurrió poco después de que Intel lanzara al mercado su primer microprocesador en 1971.

Al contrario de la opinión de Ken Olsen, cofundador y presidente de DEC, quien en 1977 en la convención de la *World Future Society* dijo que no había ninguna razón para que un individuo tuviera una computadora en su casa (Wriston, 2007, p. 75), el mercado ya estaba listo para la llegada de las microcomputadoras hogareñas, y las ventas millonarias de la *Trinidad de 1977* —que es como la revista *Byte* apodó a las computadoras Apple II, Commodore PET y Tandy TRS-80— son una prueba de ello (Dillon, 2011, p. 47).

Las computadoras hogareñas de esta época tienen el mérito de haber permitido que el público general, además de usarlas como un juguete sofisticado, adquiriera algunos

conocimientos de programación —ya que casi todas proveían un intérprete de BASIC—, pero eran muy diferentes del *Dynabook* soñado por Kay, como testimonia Papert (1980):

Kay y yo hemos compartido una visión en la que la computadora se utilizaría de forma casual y personal para una diversidad de propósitos. Pero ni la terminal escolar de 1970 ni la computadora hogareña de RadioShack de 1980 tienen el poder y la flexibilidad para proporcionar siquiera una aproximación de esta visión. Para ello, una computadora debe ofrecer gráficos mucho mejores y un lenguaje mucho más flexible . . . , a un precio que las escuelas y los individuos puedan pagar (p. 210).

No obstante, gracias a su relativo bajo costo, fueron estas microcomputadoras rudimentarias las que primero llegaron a los hogares, oficinas y escuelas de todo el mundo.

### 1.1.3. Década de 1980: La programación llega a las escuelas

Durante este periodo, las microcomputadoras irrumpieron masivamente en las instituciones educativas, dando inicio a una verdadera revolución. Andrea diSessa la definió como la *segunda revolución de las computadoras en la educación*:

Incluso medida apenas en términos de *hardware* —con cientos de miles de unidades incorporadas cada año— esta oleada tecnológica merece el título de *revolución*. Sin embargo, se trata en realidad de la segunda revolución. La primera se inició con la percepción de que las computadoras podrían revolucionar la manera en que pensamos sobre el pensamiento y la forma como entendemos el aprendizaje . . . La actual es principalmente (1) una revolución económica —las computadoras acaban de pasar el umbral de precios que les permitirá ingresar en las escuelas . . .— y (2) una revolución de *relaciones públicas* —el público en general ahora es consciente de las posibilidades y promesas y, por supuesto, espera alguna acción. Pero la verdadera revolución, la tercera, aún no ha ocurrido (diSessa, 1987, p. 364).

A pesar de utilizar criterios diferentes para clasificar esta oleada tecnológica, William Paisley y Milton Chen también afirman que se trató de una segunda revolución:

La *primera revolución electrónica* —la ocurrida con la masificación de la radio y la televisión— prescindía de la alfabetización como base para el aprendizaje, al proveer (o, a menudo, dar la impresión de proveer) conocimientos a niños aún no alfabetizados o a adultos analfabetos, para quienes estos conocimientos no eran alcanzables cuando solo estaban disponibles en libros. Paradójicamente, la *segunda revolución electrónica* —la que se dio con la masificación de los medios interactivos (microcomputadoras, videotexto, etc.)— depende más de la alfabetización que los medios impresos tradicionales (Paisley y Chen, 1984, p. 106).

En efecto, sin la necesaria alfabetización, poco sería el provecho que se podría sacar de los medios interactivos de aquella época, incluidas las primeras microcomputadoras.

Debido a ello, en esta década surgieron numerosos proyectos para aproximar las computadoras y su programación al público general. Por ejemplo, en 1981 el *Capital Children's Museum* (actualmente denominado *National Children's Museum*) estableció en Washington DC el *Future Center*, un ambiente equipado con 20 computadoras Atari 800 (Gamarekian, 1982) en el cual un plantel de adolescentes ofrecía clases de programación a alumnos desde los 3 años de edad hasta adultos, quienes podían aprender usando juegos educativos o incluso escribir sus propios juegos (Paisley y Chen, 1984, p. 108).

Ese mismo año, la revista *Byte* publicó una edición especial dedicada al lenguaje Smalltalk-80. Por primera vez, la *programación orientada a objetos* era presentada en detalle al público general. En uno de los artículos, Adele Goldberg y Joan Ross explicaron que, a pesar de que Smalltalk-80 no estaba destinado a ser utilizado por niños, se podían escribir programas de aplicación que les permitirían ser creativos y, al mismo tiempo, aprender sobre la programación de las computadoras (Goldberg y Ross, 1981, p. 348).

También en 1981 ocurrió un hito importante en la historia de la computación: IBM lanzó su primera PC (*Personal Computer*) de 16 bits con el sistema operativo MS-DOS y así cambió radicalmente el mundo de los fabricantes de microcomputadoras, de los desarrolladores de *software*, de los vendedores minoristas y del creciente mercado de usuarios de microcomputadoras (Freiberger y Swaine, 2000, p. 345). A pesar del éxito en los ámbitos corporativos, esta microcomputadora y sus clones compatibles demoraron algunos años en imponerse en las aulas de todo el mundo, pues las computadoras que predominaron en las instituciones educativas al comienzo de este periodo pertenecen a una generación anterior, basada en microprocesadores de 8 bits, que culminó tras la aparición de los modelos Commodore 128 (en 1985), Tandy TRS-80 Color Computer 3 (en 1986) y Apple IIc Plus (en 1988), desarrollados por las mismas empresas que años antes habían lanzado la *Trinidad de 1977* (Loguidice y Barton, 2014, p. 33). Una clara excepción —por usar un microprocesador de 16 bits— es la computadora TI-99/4A, de Texas Instruments, que a partir de 1981 logró tener cierta participación dentro de las aulas en los países donde se la comercializaba, por ejemplo en la República Argentina (Reggini, 1982, p. 317).

Quizá la característica más notoria de las computadoras instaladas en los centros educativos al inicio de esta década haya sido la falta de uniformidad, algo que se puede observar en el siguiente relato de Nick Evans:

En el Reino Unido, aunque el modelo BBC Micro representa un gran porcentaje del mercado, varias autoridades educativas locales han tomado como estándar las máquinas RML 380Z y 480Z, y ahora están adoptando el modelo RML Nimbus de 16 bits como el sucesor lógico. Otros utilizan una mezcla de máquinas, incluyendo Commodores de varios tipos y también los modelos Sinclair Spectrum y ZX81, para los cuales hay una sorprendente cantidad de *software* disponible. Hablar entonces de una política uniforme de provisión y soporte significa simplificar excesivamente los asuntos involucrados y engañar a la gente (Evans, 1986, p. 6).

Para solucionar esta situación, la empresa Microsoft propuso en 1983 el estándar MSX, un intento de establecer compatibilidad de *hardware* y *software* entre diferentes marcas de microcomputadoras hogareñas de 8 bits (Needle, 1984, p. 9). A pesar de ser bastante populares en gran parte de Asia, Europa y Latinoamérica —principalmente en los países de origen de las más de veinte empresas que adhirieron a esta norma—, las microcomputadoras MSX fueron un fracaso comercial en los Estados Unidos y en el Reino Unido, donde los principales fabricantes no mostraron interés en adoptar el estándar (Shapiro, 1984, p. 10) y lanzaron casi simultáneamente computadoras personales y hogareñas basadas en el microprocesador Motorola 68000 y sus derivados —que ya implementaban un conjunto de instrucciones de 32 bits—, como por ejemplo los modelos Sinclair QL (de 1984), Apple Macintosh (de 1984), Atari ST (de 1985) y Commodore Amiga (de 1985). Por ser demasiado avanzadas y costosas, las computadoras de esta nueva generación no tuvieron aceptación en la Educación Primaria, pero sí fueron usadas en los niveles superiores de los sistemas educativos de muchos países. Refiriéndose específicamente al caso de Atari Inc., Gigi Bisson dio detalles sobre este fenómeno:

Durante un corto periodo en 1981, Atari Inc. intentó agresivamente ingresar al mercado de la educación. El *Atari Institute for Educational Action Research* donó un millón de dólares en *hardware* y estipendios en efectivo para proyectos educativos . . . Cinco años más tarde, Atari parece haber salido del mercado de la Educación Primaria donde Apple está profundamente atrincherada con sus computadoras de 8 bits, pero está consiguiendo posicionarse firmemente en el segmento universitario con el modelo ST (Bisson, 1986, p. 39).

Horacio Reggini, uno de los pioneros de la enseñanza del lenguaje Logo en la República Argentina, publicó en 1981 el artículo *Las computadoras en la educación*, en el cual describía el uso que se les daba entonces a las computadoras en el aula (tabla 1).

**Tabla 1**

*Modalidades de uso de las computadoras en el aula*

Las computadoras como...	Propósito
Máquinas de calcular	Instruir a niños o adolescentes en lenguajes Basic, Fortran, Cobol o Pascal, para resolver ecuaciones o automatizar facturaciones contables. No interesa desde un punto de vista educativo profundo.
Máquinas de enseñar	Programar al estudiante en vez de emplearlas en sentido inverso. Es la denominada instrucción asistida por computadora (en inglés: CAI, <i>computer assisted instruction</i> ).
Instrumentos de otras disciplinas	Consiste en que los estudiantes escriban programas y/o utilicen programas ya preparados para ayudarse en la resolución de los problemas clásicos que se plantean en las diversas asignaturas.
Instrumento intelectual para el estudiante	Le permite explorar un microcosmos rico en experiencias lógicas. El estudiante desarrolla una serie de conceptos compatibles con su propio crecimiento y percepción.

*Nota:* Basado en “Las computadoras en la educación” por H. Reggini, 1981, *Computadoras: ¿creatividad o automatismo?* por H. Reggini, 1988, pp. 14-16. Copyright 1988 por Ediciones Galápagos

La cuarta modalidad mencionada por Reggini (1988) se refiere principalmente a la utilización dada en aquellos años a las computadoras y al lenguaje Logo en la Educación Primaria, donde el objetivo no es que los estudiantes aprendan computación y lenguajes de programación como fines en sí mismos —para luego utilizarlos profesionalmente— sino que desarrollen el pensamiento computacional.

Con el objetivo de facilitar el aprendizaje del lenguaje Pascal tanto en la Educación Secundaria como en la Superior, en 1981 Richard Pattis publicó *Karel The Robot: A Gentle Introduction to the Art of Programming*, un libro en el cual —empleando la estructura del Pascal— se enseñaba a escribir programas para mover un robot por una cuadrícula, utilizando para ello comandos pertenecientes a un conjunto de tan solo cinco instrucciones básicas, a pesar de lo cual —según este autor— Karel tenía una potencia computacional equivalente a la que posee una máquina de Turing (Pattis, 1981, p. viii), es decir, permitía implementar los mismos algoritmos que una computadora. Douglas y Denise Green consideraron excelente el concepto de aprender un lenguaje de programación a través de la simulación y elogiaron el diseño del simulador real, reconociendo la inspiración de Seymour Papert y el Logo del MIT en este proyecto (Green y Green, 1982, p. 29).

En 1983, el Pascal ganó impulso en la industria del *software* cuando Borland presentó Turbo Pascal, un entorno de desarrollo integrado (*Integrated Development Environment* o IDE) que permitía editar, compilar y enlazar código, todo desde el mismo producto, lo que facilitaba enormemente el desarrollo de programas. Freiburger y Swaine (2000) describen cómo el fundador de Borland, Philippe Kahn, lo presentó en la empresa:

Mientras los otros directores de Borland observaban, Kahn hizo una demostración de Turbo Pascal. Quedaron aturdidos. Era asombrosamente rápido y tan compacto que funcionaba fácilmente en la memoria limitada de la PC. Este programa era mejor que cualquier cosa que recordaran haber visto en un mainframe o minicomputador —un producto pulido, atractivo y brillantemente codificado. Incluso los programadores aficionados podrían utilizarlo, y hasta se podría aprender a programar con él . . . Además, Kahn hizo otro anuncio bombástico: venderían Turbo Pascal por correo a 49,95 dólares. En aquel momento, Microsoft ofrecía un compilador de Pascal por aproximadamente diez veces ese precio (pp. 201-202).

En 1984, la organización *College Board* —que a través de sus miles de instituciones asociadas en todo el mundo promueve el acceso a la Educación Superior ofreciendo a estudiantes avanzados de la Educación Secundaria cursos y exámenes reconocidos como créditos por universidades de los Estados Unidos— ofreció por primera vez el examen *Advanced Placement Computer Science*. Para la evaluación de los conocimientos de programación, este examen utilizaba el lenguaje Pascal («The College Board», 2017).

También ese año, el MIT publicó *Structure and Interpretation of Computer Programs* (Abelson, Sussman y Sussman, 1984), un libro revolucionario porque en lugar de dedicar un curso completo a aprender las características de determinado lenguaje —algo bastante común, aun hoy en día— permitía concentrarse en analizar problemas y construir herramientas para resolverlos, gracias a que utilizaba Scheme (un dialecto del LISP), que es un lenguaje con una estructura simple y fácil de describir, como explican los autores:

Nunca enseñamos formalmente el lenguaje . . . Solo lo usamos, y los estudiantes lo asimilan en pocos días. Esta es una gran ventaja de los lenguajes como el LISP: tienen muy pocas maneras de formar expresiones compuestas, y casi ninguna estructura sintáctica. Todas las propiedades formales se pueden cubrir en una hora, como las reglas del ajedrez. Después de un corto tiempo nos olvidamos de los detalles sintácticos del lenguaje . . . y seguimos con los problemas reales: averiguar qué queremos calcular, cómo vamos a descomponer los problemas en partes manejables y cómo trabajaremos en las partes (p. XVIII).

En 1985, John Anderson y Brian Reiser., de la CMU (*Carnegie-Mellon University*), presentaron GREATERP (*Goal-Restricted Environment for Tutoring and Educational Research on Programming*), un programa de computadora que, básicamente, funcionaba como un tutor de programación. GREATERP seguía al estudiante mientras este escribía su código, símbolo por símbolo. Si la entrada era correcta, se quedaba en silencio y esperaba más datos. Si, en cambio, la entrada era errónea, lo interrumpía y le daba un consejo (J. R. Anderson y Reiser, 1985, p. 159).

También en 1985, la UNESCO (*Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura*) publicó un informe sobre el desarrollo de la informática en los sistemas educativos de un grupo de países de América Latina y el Caribe. En este informe se describía así la situación en la República Argentina:

La Comisión Nacional de Informática ha elaborado un documento completo que señala las pautas generales sobre las políticas educativas a seguir respecto a la enseñanza de la informática y la computación para los niveles pre-básico, básico, medio y superior. En los programas oficiales de la educación básica se han incorporado en forma regular diversos contenidos sobre el tema. En especial, aquellos referidos a los conceptos básicos de informática y lo que se ha llamado *alfabetización computacional*, su importancia en el mundo actual, la descripción general de los equipos, cómo operar y utilizar los computadores. También se han incorporado las bases de la programación y la aplicación de computadores a la solución de problemas . . . En el Ministerio de Educación y Justicia no se dispone de antecedentes precisos sobre experiencias en que se utilice el computador con fines educativos dentro de la educación formal. Sin embargo, se sabe que a veces se usan para apoyar a la docencia en establecimientos privados de nivel básico y medio. Con menos frecuencia se usan para los mismos fines en establecimientos públicos de nivel medio y superior (UNESCO, 1985, pp. 9-10).

Como puede observarse, los programas oficiales de este periodo ya establecían que debían enseñarse las bases de la programación —incluso desde la Educación Primaria—, aunque desde el propio Ministerio de Educación y Justicia se reconocía la carencia de un conocimiento preciso sobre la utilización que las computadoras tenían en los establecimientos educativos —tanto públicos como privados— del país.

Aprovechando las lecciones aprendidas en otros países que habían empezado antes a incorporar las computadoras en las aulas, en 1985 se inició en España el *Proyecto Atenea*, llevado a cabo por el Ministerio de Educación y Ciencia. Si bien su implantación



solamente tuvo lugar en las 11 comunidades autónomas cuyos sistemas educativos estaban aún bajo la órbita de este ministerio, se trabajó en forma coordinada con las otras seis comunidades que ya controlaban para entonces sus propios sistemas educativos —Cataluña, Galicia, Andalucía, Canarias, el País Vasco y la Comunidad Valenciana—, por lo cual se llevaron a cabo en forma simultánea siete proyectos paralelos, con nombres y planes diferentes, aunque con objetivos similares (Osorio y Nieves, 2014, pp. 122-123).

La OCDE (*Organización para la Cooperación y el Desarrollo Económicos*) evaluó el Proyecto Atenea en 1990 y, en el informe publicado, destacó que uno de sus principales aciertos fue “la adopción, en un momento de bastante incertidumbre, de un estándar de *hardware* que goza de plena vigencia” (OCDE, 1991, p. 78). En efecto, —y a diferencia de lo que ocurría en otras partes— las computadoras destinadas a las instituciones educativas del Proyecto Atenea eran todas compatibles entre sí, pues “se establecieron criterios de calidad para el equipamiento: PC compatible, sistema operativo MS-DOS, color, capacidades gráficas, ratón e impresora matricial” (p. 21). También recordó que los objetivos del Proyecto Atenea eran los siguientes:

- 1) Reflexionar sobre los currículos de las distintas asignaturas y analizarlos desde la perspectiva de las nuevas tecnologías de la información para definir cómo pueden integrarse dichas tecnologías en el aula.
- 2) Desarrollar materiales didácticos para experimentar las posibilidades que ofrecen las nuevas tecnologías de la información en los procesos de enseñanza y aprendizaje.
- 3) Usar las nuevas tecnologías de la información como recursos para mejorar la calidad de la enseñanza en las distintas áreas del conocimiento y sus aspectos interdisciplinares.
- 4) Potenciar el uso de los ordenadores para la creación de nuevos contextos autónomos de aprendizaje individual y grupal y para el desarrollo de la creatividad, la autoestima y los procesos de pensamiento en los alumnos.
- 5) Explorar la utilidad de las nuevas tecnologías de la información para la educación de alumnos con necesidades especiales (p. 16).

Como se puede observar, la enseñanza de la programación no estaba contemplada entre los objetivos del Proyecto Atenea, a pesar del interés de algunos profesores:

En las etapas iniciales, la formación tecnológica fue muy importante . . . Se enseñó al profesorado a utilizar principalmente el *software* de propósito general. Algunos profesores solicitaban a los monitores el aprendizaje de un lenguaje de programación, que no formaba parte de los contenidos de formación (p. 33).

Sin embargo, aunque la enseñanza de la programación estaba incluida en el proyecto, esto no impidió que en muchas instituciones se la impartiera, como quedó registrado en el informe de la OCDE:

En el Bachillerato (14 a 17 años), existe la posibilidad de programar asignaturas optativas de Enseñanzas y Actividades Técnico-Profesionales. Esta es la vía que utilizan muchos centros para enseñar programación a los alumnos (p. 59) . . . Ciertos sectores del profesorado . . . se hacen eco de una demanda social emergente y exploran con sus alumnos los caminos de la programación de ordenadores (p. 75).

Es de suponer que estos casos no eran excepcionales, ya que la lista del *software* adquirido en el marco del proyecto incluye dos versiones de Logo (Logo SB y Acti-Logo) —destinados a la Educación Primaria y Secundaria— así como también Turbo Pascal —dirigido a la Educación Secundaria— (p. 85).

Asimismo, el informe de la OCDE vincula el Proyecto Atenea con la enseñanza de la robótica y, en particular, con la programación de robots mediante el uso del lenguaje Logo:

La utilización de los ordenadores en las aulas ha llevado a la promoción de la robótica como una actividad curricular completa, no solo en centros de enseñanzas medias donde existen relaciones con asignaturas técnicas, sino también en aquellos donde no las hay; asimismo, en centros de enseñanza primaria, especialmente en zonas rurales, se encontraron muestras de aplicaciones basadas en Logo (p. 43).

La robótica educativa también ganó un gran impulso en 1985, cuando se inauguró oficialmente el *MIT Media Lab* (Laboratorio de Medios del *MIT*) (Moss, 2011, p. xi). Entre las corporaciones que desde el primer momento subsidiaron el laboratorio a cambio de compartir sus frutos intelectuales se encontraba la danesa Lego A/S (p. 16). Por ello, cuando en el *MIT Media Lab* se desarrolló LEGO/Logo, esta empresa pronto estuvo en condiciones de lanzar una versión comercial del sistema. LEGO/Logo es un entorno que tiene como objetivo llevar actividades de diseño e invención al aula. Con LEGO/Logo, un estudiante puede construir una máquina con piezas de LEGO, conectar la máquina a una computadora y escribir (usando una versión modificada del lenguaje de programación Logo) un programa para controlar la máquina (Resnick, Ocko y Papert, 1988, p. 14).

También en 1985, la manera de manejar las PC de IBM y sus compatibles —y también la forma de programar aplicaciones para ellas— cambió radicalmente, a partir del lanzamiento del sistema operativo Windows, con el que Microsoft seguía la tendencia que otros fabricantes estaban imponiendo: la incorporación de la interfaz gráfica de usuario (*Graphical User Interface* o GUI) en las computadoras personales. Apple había lanzado el modelo Lisa en 1983 y el Macintosh en 1984. VisiCorp tenía VisiOn), que ponía todos sus productos, incluyendo VisiCalc, en ventanas fáciles de usar. Digital Research había clonado la GUI del Mac en su producto GEM e IBM ofrecía TopView (Freiberger y Swaine, 2000, pp. 392-393).

Ese mismo año, Bjarne Stroustrup escribió *The C++ Programming Language*, un libro que —hasta la publicación del primer estándar oficial en 1998— funcionaría como una referencia del lenguaje C++ —un desarrollo suyo que incorpora el soporte de la *programación orientada a objetos* al lenguaje C— (Stroustrup, 1986). El lenguaje C había sido creado a principios de la década anterior en los Laboratorios Bell (*Bell Labs*), pero recién se popularizó una década más tarde, como recuerda su autor:

Durante la década de 1980, el uso del lenguaje C se extendió ampliamente, y los compiladores estuvieron disponibles en casi todas las arquitecturas y sistemas operativos; en particular, se hizo popular como herramienta de programación para computadoras personales, tanto para fabricantes de *software* comercial como para usuarios finales interesados en la programación (Ritchie, 1993, p. 681).

En 1986, Andrea diSessa y Harold Abelson —en aquel entonces investigadores de la *University of California, Berkeley* y el *MIT*, respectivamente— presentaron Boxer, un entorno derivado de Logo donde usuarios no expertos usarían la programación para controlar un *medio reconstruible*. A pesar de lucir muy similares a las ventanas de los sistemas con GUI, las cajas (*boxes*) de Boxer eran los objetos de cómputo del sistema, cuyo contenido podía ser gráfico o textual como, por ejemplo, el código de un procedimiento o el registro de una base de datos, modificables —en tiempo real— tanto por los programas como manualmente (diSessa y Abelson, 1986, pp. 862-863).

Un año más tarde, en 1987, Apple dio a conocer el *software* HyperCard, descrito por Bill Atkinson —su creador— como una herramienta que permitía que los programas

fueran escritos por personas que no eran programadores (Lewis, 1987). En efecto, con HyperCard, los usuarios podían convertirse en autores del *software* que utilizaban:

Los usuarios pueden escribir *software* para almacenar imágenes, sonidos, números de teléfono o virtualmente cualquier tipo de datos concebibles y recuperarlos en la forma que consideren más útil . . . Para la mayoría, HyperCard es una introducción indolora a los programas que establecen enlaces entre bloques de información . . . Debajo de todo esto hay un nuevo y elegante lenguaje llamado HyperTalk, que está estructurado casi como el inglés convencional . . . Con HyperCard, Apple visualiza una nueva clase de usuarios de computadoras llamados *desarrolladores de información*, que pueden incorporar sus ideas al *software* sin tener que pasar meses o años aprendiendo lenguajes de programación exóticos (Lewis, 1987).

Simultáneamente, la publicación del libro *Object-Oriented Software Construction* (B. Meyer, 1988) ayudó a la definitiva popularización de la *programación orientada a objetos* y sus metodologías en los ámbitos académicos de todo el mundo.

Hacia el final de esta década, la programación ya se había afianzado en todos los niveles educativos. Tal es así, que en 1989 la UNESCO organizó y llevó a cabo la primera IOI (*International Olympiad in Informatics*), para estudiantes no mayores de 19 años, que tuvo lugar en Pravetz, Bulgaria. Los equipos participantes, formados por tres o menos estudiantes, debían resolver en 4 horas un problema de naturaleza algorítmica, utilizando cualquier lenguaje de programación —la organización sugería utilizar BASIC, Logo o Pascal— y una computadora personal —que podía ser una perteneciente al propio equipo o una ofrecida por la organización, que disponía de los modelos Apple II+ e IBM-PC/XT/AT compatibles— (Kenderov y Maneva, 1989, pp. 5-6). Con el pasar de los años, la IOI fue creciendo y contando con la participación de cada vez más países. Por ejemplo, la República Argentina envió equipos a partir de la segunda edición —que se realizó en 1990 en la ciudad de Minsk, Unión Soviética— (Kalinichenko, s. f.).

#### **1.1.4. Década de 1990: Disminuye el entusiasmo por la programación**

A comienzos de la década de 1990, varias innovaciones hacían suponer que las instituciones de todos los niveles educativos se mantendrían interesadas en enseñar la programación por mucho tiempo más.

En 1990, un comité internacional definió la primera versión de Haskell, un lenguaje que desde entonces ha sido ampliamente utilizado para enseñar la *programación funcional*. Este paradigma ofrece una visión de alto nivel de la programación, dándoles a sus usuarios elementos que les ayudan a construir bibliotecas de funciones elegantes, pero a la vez poderosas y generales (Thompson, 1999, p. xiii).

También en 1990, Mitchel Resnick desarrolló MultiLogo, una versión de Logo capaz de ejecutar diferentes programas al mismo tiempo, como explica este autor:

A menudo los niños construyen máquinas con LEGO y quieren que diferentes programas corran al mismo tiempo. Por ejemplo, después de construir un parque de diversiones con una rueda de la fortuna y un carrusel, un niño puede querer que las dos atracciones ejecuten diferentes programas a la vez. Muchos niños se sorprenden de que el Logo tradicional no pueda hacer una cosa tan simple (Resnick, 1997, p. 45).

Un año más tarde, Resnick presentaría StarLogo, que además de paralelismo incorporaba aspectos de la *programación orientada a objetos* (p. 41).

En 1991, tuvieron lugar tres importantes eventos relacionados con la programación y su enseñanza. El primero es la publicación del libro *Constructionism*, en el cual Idit Harel y Seymour Papert expusieron el trabajo llevado a cabo por el *Epistemology and Learning Group* del *MIT Media Lab*. Así resumen estos investigadores su propuesta:

Lo que hace único al Grupo de Epistemología y Aprendizaje no es su diversidad, sino la búsqueda de la unidad subyacente. La creación de una multitud de situaciones de aprendizaje (a veces llamadas entornos de aprendizaje o *micromundos*) es un gran activo, pero lo que le da al *construccionismo* su condición de proyecto teórico es su dimensión epistemológica ... El *construccionismo* ... comparte la connotación constructivista del aprendizaje como *construcción de estructuras de conocimiento*, independientemente de las circunstancias del aprendizaje. A continuación, añade la idea de que esto sucede de manera especialmente favorable en contextos donde el aprendiz está conscientemente comprometido en la construcción de una entidad pública, ya sea un castillo de arena en la playa o una teoría del universo (Papert y Harel, 1991, pp. 1-11).

El segundo evento ocurrido en 1991 y relacionado con la programación y su enseñanza fue el lanzamiento de GNU/Linux, un sistema operativo que es *software* libre —o sea que, además de ser gratuito, su código fuente está disponible para que los usuarios lo puedan estudiar, modificar y mejorar— y que incluye excelentes herramientas como la

GCC (*GNU Compiler Collection*) para programar en los lenguajes C, C++, Objective-C, Fortran, Ada y Go (Free Software Foundation, s. f.). Por tales motivos, ha sido adoptado en muchas instituciones educativas alrededor del mundo (Stallings, 2005, p. 95).

El tercer evento ocurrido en 1991 —y que posteriormente tendrá un gran impacto sobre la programación y su enseñanza— es el surgimiento del lenguaje Python. Según Lambert y Osborne (2012), Python tiene los siguientes beneficios pedagógicos:

1) Python tiene una sintaxis simple y convencional. Las sentencias de Python son muy cercanas a las de los algoritmos en pseudocódigo . . . ; 2) Python tiene una semántica segura. Cualquier expresión o declaración cuyo significado viole la definición del lenguaje produce un mensaje de error; 3) Python es útil en distintas escalas. Es muy fácil para un principiante escribir programas sencillos, pero también hay características avanzadas . . . ; 4) Python es altamente interactivo. Las expresiones y declaraciones se pueden ingresar en un intérprete para permitir que el programador pruebe el código y reciba una retroalimentación inmediata . . . ; 5) Python es un lenguaje de propósito general con recursos para aplicaciones contemporáneas, incluyendo aplicaciones distribuidas; 6) Python es gratuito y tiene un uso generalizado en la industria . . . La experiencia con Python tiene un gran valor en el *curriculum vitae* (p. xiv).

Sin embargo, a pesar de la existencia de una nueva teoría del aprendizaje para sustentarla —el construccionismo—, de la disponibilidad de un sistema operativo libre con herramientas para la programación —GNU/Linux y la GCC— y del surgimiento de un lenguaje con claros beneficios pedagógicos —Python—, el entusiasmo por la enseñanza de la programación en la Educación Primaria y en la Educación Secundaria —tan importante en la década anterior— no duró demasiado. En poco tiempo, la mayoría de los educadores pasó a ver la programación como una actividad muy específica y técnica que solamente era apropiada para un pequeño segmento de la población (Kafai y Burke, 2014, pp. xi-xii). Los sistemas educativos en España no fueron una excepción a esta tendencia, como confirman Osorio y Nieves (2014):

El entusiasmo inicial y la estimación de que las computadoras iban a producir un cambio radical en el proceso de enseñanza-aprendizaje dio paso a un sentimiento de decepción porque muy poco había cambiado realmente. Esta percepción generó cierto escepticismo sobre la utilidad de las computadoras en las escuelas primarias y secundarias, de modo que las acciones gubernamentales que apoyaban su incorporación en las escuelas disminuyeron significativamente a principios de los años noventa (p. 129).

Además, a principios de los noventa, se popularizaron los constructores de GUI. En 1991, Microsoft lanzó Visual Basic, un sistema que se destacaba por su constructor de GUI integrado. Del mismo modo, en 1995 Borland lanzó Delphi, un IDE para Pascal. Michael Kölling recuerda cómo el desarrollo de GUI dificultaba que los estudiantes aprendieran los conceptos introductorios de la programación:

Estos IDE se usaron a veces en la enseñanza. No obstante, por ser herramientas profesionales enfocadas en la construcción de GUI (normalmente con generación automática de código), no facilitaban el aprendizaje de los principios fundamentales de la programación, y los intentos de enseñar esta disciplina empezando con la construcción de GUI hicieron que se generalizara una visión contraria al uso de los IDE en los cursos introductorios (Kölling, 2015, p. 7).

No obstante, aunque en las instituciones educativas y en los hogares el entusiasmo por la programación continuaba en declive, el interés por el uso de las computadoras pronto volvería a aumentar —llegando incluso a niveles nunca antes vistos— después de que en 1993 el CERN (*Conseil Européen pour la Recherche Nucléaire*) pusiera en el dominio público el protocolo de la Web —que había sido creada 4 años antes— permitiendo de esta manera su adopción masiva fuera de los ámbitos científicos. Tim Berners-Lee, el inventor de la Web, narra así este momento histórico y sus consecuencias inmediatas:

El 30 de abril Robert [Cailliau] y yo recibimos una declaración . . . diciendo que el CERN estaba de acuerdo en permitir que cualquiera usara el protocolo de la Web y su código gratuitamente, para crear un servidor o un navegador . . . sin pagar regalías y sin ninguna otra restricción (Berners-Lee y Fischetti, 2000, p. 74) . . . Merit Inc. —la empresa responsable del enlace troncal de Internet de la *National Science Foundation*— midió el uso de los diferentes protocolos en Internet. En marzo de 1993, las conexiones a la Web habían representado el 0,1% del tráfico de Internet. En setiembre ya eran el 1% y en diciembre el 2,5%. Nunca antes un protocolo de Internet había tenido un crecimiento tan espectacular (p. 80).

La popularización del acceso a la Web reavivó el interés de los gobiernos en incorporar computadoras en los sistemas educativos. Según Osorio y Nieves (2014),

los programas de incorporación de las tecnologías de la información proliferaron nuevamente, pero esta vez con la incorporación de la comunicación, y se popularizó el término TIC (*Tecnologías de la Información y la Comunicación*). Este auge se debió al éxito de la Web y al creciente peso de las políticas gubernamentales orientadas al desarrollo de la sociedad de la información (p. 128).

Asimismo, la necesidad de instalar *software* en las instituciones disminuyó bastante, ya que la Web permitió la posibilidad del trabajo *en línea*. El sitio web MaMaMedia.com —dirigido a niños de hasta 12 años— fue uno de los primeros en aprovechar las nuevas posibilidades que se abrieron con la popularización de la Web:

En 1995, en los albores de la Web, Idit Harel reunió a un pequeño equipo de educadores, empresarios, informáticos y diseñadores de medios para crear un lugar en línea donde todos los niños pudieran aprender y divertirse . . . Basado en 15 años de proyectos de investigación en el MIT y en Harvard, la idea central de MaMaMedia.com fue incentivar a los niños a usar la Web, desde una edad temprana, para un aprendizaje lúdico, expresivo y activo, no solo para buscar y encontrar información . . . Este ambiente innovador anima a los niños a ser diseñadores de medios y no meros consumidores («MaMaMedia.com», s. f.).

Casi al mismo tiempo, un nuevo integrante aparece en el universo de la programación: el lenguaje Java, que fue desarrollado por Sun Microsystems a partir de 1991 y presentado al público en 1995. Muchas empresas empezaron a utilizar Java debido a que funciona *independientemente de la plataforma*. Las tiendas en línea, por ejemplo, pueden usar Java para crear interfaces de usuario para que los compradores adquieran sus productos. También pueden usar Java del lado del servidor para mantener sus bases de datos actualizadas a medida que sus productos van siendo comprados (Russell, 2001, p. xx).

Además de conseguir una amplia penetración en los ámbitos empresariales, el lenguaje Java también fue adoptado rápidamente por muchas instituciones de la Educación Superior en todo el mundo. Bowen (1997) describió así este fenómeno:

El mercado editorial ha recibido una avalancha de libros de Java. El primer año ya se habían lanzado 150 nuevos títulos, con muchos más en espera. Eso supera el número de libros de C++ . . . Establecerse en el mundo académico le dará a Sun Microsystems una oportunidad de infiltrarse en el mundo corporativo. El mundo académico ya había sido el vehículo propulsor del UNIX, la programación en C y los paradigmas orientados a objetos. Entrar a la Educación Superior no solo le permitirá conseguir adherentes en los altos cargos de las corporaciones a medida que los graduados vayan siendo promovidos, sino que también será un recurso fundamental para formar ingenieros capacitados en las más recientes herramientas.

A mediados de los años noventa, la difusión de la *programación orientada a objetos* llevó a la popularización de nuevas técnicas de programación. Para resolver problemas que se presentan una y otra vez, es común aplicar soluciones que ya han sido utilizadas



anteriormente y cuya efectividad es ampliamente conocida: los *patrones de diseño*. Estas técnicas se hicieron muy conocidas a partir de la publicación del libro *Design Patterns: Elements of Reusable Object-Oriented Software* (Gamma, Helm, Johnson y Vlissides, 1994).

Asimismo, específicamente para la enseñanza de este paradigma surgieron lenguajes y entornos de programación específicos. En 1995, Michael Kölling y John Rosenberg —de la *University of Sydney*— presentaron el lenguaje Blue, que defendían así:

A menudo se ha argumentado que las universidades y los colegios no pueden permitirse el lujo de enseñar *lenguajes de juguete*, ya que tienen la obligación de proporcionar a sus estudiantes habilidades del mundo real para empleos del mundo real. Esto ha llevado a muchas instituciones a adoptar el C++ como su lenguaje de enseñanza. Estamos de acuerdo en que un graduado debe ser competente como programador en C++ o en un lenguaje similar. Es nuestra firme convicción, sin embargo, que la experiencia con un año de un buen lenguaje de enseñanza y un año de C++ produce mejores programadores de C++ que dos años de C++ (Kölling y Rosenberg, 1996, p. 190).

También en 1995, en la *Universidad de Valladolid* el profesor Miguel A. Revilla creó el *UVa Online Judge*, un sitio web que contiene cientos de problemas similares a los utilizados durante los concursos de programación. Los usuarios registrados —el sistema está abierto al público desde 1997— pueden enviar sus programas escritos en una variedad de lenguajes (Pascal, ANSI C, C++, C++11, Java o Python 3), tratando de resolver cualquiera de los problemas disponibles en la base de datos, y este juez automático los corregirá inmediatamente («UVa Online Judge», s. f.).

En 1996, se incorporó al sistema AgentSheets —cuyo desarrollo se había iniciado en 1989 en la *University of Colorado* (Repenning, 1991, p. 191)— el entorno de programación Visual AgenTalk. La forma de programar propuesta por este entorno —conocida como *programación táctil*— era muy novedosa y hacía que los programas fueran fáciles de comprender y componer, ya que los comandos tenían una sintaxis visual que combinaba componentes textuales y pictóricos, y la manipulación mediante *drag and drop* (arrastrar y soltar) les permitía a los usuarios mover comandos y reglas y comprender su funcionalidad al aplicárselos a los agentes. Cualquier comando o regla se podía ejecutar en cualquier momento y la retroalimentación era inmediata. También permitía publicar los programas en la *World Wide Web* (Repenning y Ambach, 1996, p. 102).

Ese mismo año, la empresa Chrysler inició el proyecto C3 (*Chrysler Comprehensive Compensation*) para desarrollar un nuevo sistema de nómina. Basándose en el trabajo de Jim Coplien y Ward Cunningham sobre el proceso de desarrollo de *software*, Kent Beck —con Ron Jeffries y el equipo C3 de Chrysler— inventó y le dio el nombre definitivo a la *Extreme Programming* (Fraser et al., 2003, p. 459), que se ha vuelto desde entonces una metodología muy destacada entre los *procesos ágiles* de desarrollo de *software*.

También en 1996, un equipo de investigadores de Apple Computer, Inc. desarrolló el entorno Squeak, cuya máquina virtual está escrita completamente en Smalltalk, lo que la hace fácil de depurar, analizar y cambiar. Se trata de un entorno de desarrollo en el que es fácil construir *software* educativo que puede ser utilizado —e incluso programado— por personas no técnicas y por niños (Ingalls, Kaehler, Maloney, Wallace y Kay, 1997, p. 318).

Después de salir de Apple y mudarse a Walt Disney Imagineering Research & Development, Inc., el equipo dirigido por Alan Kay creó Etoys, un sistema de autoría multimedia programado en Squeak. En 1997, Etoys fue probado por primera vez con el objetivo de ayudar a que los niños aprendan ideas poderosas, construyéndolas (Kay, 2007, p. 4). Según las propias palabras de Kay:

Etoys es como Logo, pero con objetos universales que pueden disfrazarse; . . . Etoys es como StarLogo, pues permite la ejecución simultánea de un enorme número de objetos; . . . Etoys es como HyperCard, pues permite la autoría de presentaciones usando objetos multimedia programables, pero de manera colaborativa, en tiempo real y con la posibilidad de publicar las creaciones en la Web para ser vistas desde un navegador; . . . Etoys es como Smalltalk, ya que todo en este entorno es dinámico —incluso el idioma de la interfaz puede cambiarse en cualquier momento— y, además, . . . Etoys es *ecuménico*: corre de manera igual en más de 20 plataformas . . . y en la computadora portátil OLPC (*One Laptop Per Child*) XO (pp. 5-9).

También en 1997, un equipo dirigido por James Rumbaugh, Grady Booch e Ivar Jacobson diseñó y publicó el UML (*Unified Modeling Language*), un lenguaje gráfico de modelado de sistemas de *software* ampliamente utilizado para especificar, construir y documentar sistemas (Rumbaugh, Booch y Jacobson, 2000).

En 1998, Lego A/S lanzó al mercado el primer *kit* de su serie LEGO Mindstorms. Fue un éxito inmediato, aunque también un poco sorprendente —debido al perfil de sus principales aficionados—, como describen David Robertson y Bill Breen (2013):

Con un precio minorista de 199 dólares, el *kit* original consistía en una aplicación de *software* que los clientes podían usar para programar un ladrillo basado en microcontroladores, denominado RCX (*Robotic Command Explorer*), tres motores, tres sensores y una colección de aproximadamente setecientos ladrillos, vigas, engranajes, ejes y ruedas para construir una amplia gama de robots. En esos primeros meses cruciales posteriores al lanzamiento de Mindstorms, los directivos del grupo Lego quedaron desconcertados cuando supieron por medio de encuestas que el 70 por ciento de los aficionados a Mindstorms —un *kit* inicialmente orientado al público infantil— eran adultos, no niños (pp. 345-347).

Hacia el final de esta década, sin embargo, la importancia de los lenguajes de programación diseñados con fines educativos (Pascal, Blue, etc.) había disminuido considerablemente como consecuencia de la popularización de la orientación de objetos en la enseñanza introductoria de la programación. Estos lenguajes educativos comenzaron a ser reemplazados en la mayoría de las instituciones de enseñanza por los lenguajes más ampliamente utilizados en la industria. C++, Visual Basic y Java eran los lenguajes de elección en los cursos introductorios en este periodo —todos ellos fueron diseñados para ser utilizados por programadores profesionales (Kölling, 2015, p. 7).

Como consecuencia de esta tendencia, el equipo que 4 años antes había desarrollado Blue presentó, en 1999, el entorno BlueJ que, según su autor, no era más que una reimplementación del entorno Blue para el lenguaje Java:

BlueJ no solo admitía Java como el lenguaje de usuario, sino que también estaba implementado en Java (mientras que Blue había sido implementado en C++). La promesa del desarrollo multiplataforma, con la arquitectura de máquina virtual y compilación *just-in-time* de Java, resultó ser vital para mantener BlueJ funcionando en diferentes sistemas operativos(p. 12).

También en 1999, Randy Pausch y su equipo de la CMU presentaron una nueva versión de Alice, su herramienta programable de modelado 3D diseñada para hacer que la creación de mundos gráficos interactivos tridimensionales fuera accesible para estudiantes universitarios no científicos (Kelleher y Pausch, 2005, p. 114). Esta nueva versión, conocida como Alice99, mantuvo el lenguaje Python para programar la manipulación de los objetos en la pantalla, pero redujo sensiblemente la cantidad de escritura de código que se necesitaba para generar las animaciones, ya que

—al igual que en otros entornos más o menos contemporáneos como AgentSheets o Etoys— los objetos se podían arrastrar y soltar en una ventana donde las acciones aparecían agrupadas en diversos menús (p. 118).

Asimismo, ese año la organización *College Board* dejó de usar en lenguaje Pascal en el examen *Advanced Placement Computer Science* y pasó a evaluar los conocimientos de programación usando el lenguaje C++ («The College Board», 2017).

También ese año, los líderes del equipo que había desarrollado el UML publicaron *The Unified Software Development Process* (Jacobson, Booch y Rumbaugh, 2000), el primer libro donde se describía el nuevo *Proceso Unificado de Desarrollo Software* —un marco de desarrollo dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental—.

### 1.1.5. Década de 2000: Se priorizan otros usos de las computadoras

El nuevo milenio se inició con fuertes cuestionamientos referidos, en algunos casos, a las deficiencias en la enseñanza de la programación y, en otros, a la ausencia de la enseñanza de la programación en muchos cursos de informática restringidos a enseñar apenas el manejo de las computadoras.

En 2001, un grupo internacional de profesores que contaba con representantes de Estados Unidos, Reino Unido, Australia, Polonia e Israel y estaba liderado por Michael McCracken del *Georgia Institute of Technology* llevó a cabo un estudio para determinar si los estudiantes de los cursos introductorios de informática sabían cómo programar en el nivel de habilidad esperado. Los resultados —presentados ese año en la conferencia anual ITiCSE (*Innovation and Technology in Computer Science Education*)— fueron alarmantes:

Varias universidades participaron en nuestra evaluación de prueba y los resultados decepcionantes sugieren que muchos estudiantes no saben cómo programar al finalizar sus cursos introductorios. Para una muestra combinada de 216 estudiantes de cuatro universidades, el puntaje promedio fue de 22,89 puntos sobre 110 puntos posibles en los criterios de evaluación desarrollados para este estudio (McCracken et al., 2001, p. 125).

También en 2001, Andrea diSessa publicó el libro *Changing Minds: Computers, Learning, and Literacy*. Allí, este autor separa dos conceptos que tienen nombres similares

pero significados muy distintos: *computer literacy* (alfabetización en el uso de las computadoras) y *computational literacy* (alfabetización computacional):

El primer concepto es tan antiguo como las propias computadoras, y significa algo así como poder encender una computadora, insertar un CD y tener suficientes habilidades de teclado y ratón para hacer que pasen algunas cosas interesantes en algunas aplicaciones estándar. La *alfabetización computacional* es diferente ... Si existe una verdadera alfabetización computacional, será infraestructural de la misma manera que la alfabetización textual lo es en las escuelas actuales. Los estudiantes la aprenderán y usarán constantemente a través de sus carreras escolares y más allá en diversas actividades científicas, humanísticas y expresivas. Fuera de las escuelas, una alfabetización computacional permitirá a la civilización pensar y hacer cosas que serán nuevas para nosotros, así como las innovaciones de la sociedad alfabetizada moderna habrían sido casi incomprensibles para las anteriores sociedades no alfabetizadas (diSessa, 2001, pp. 4-5).

Ese mismo año, Mathew Nelson —un ingeniero de *software* del *Advanced Technology Internet Group* en IBM— desarrolló Robocode, un simulador de batallas entre robots con forma de tanques de guerra programables en Java que rápidamente se volvió muy popular entre los estudiantes de programación de todo el mundo al ser publicado en AlphaWorks —una comunidad web que existía en aquel entonces para que los desarrolladores obtuvieran una vista previa de las tecnologías emergentes de los laboratorios de investigación de IBM— (S. Li, 2002, p. 1).

Basándose en que la *alfabetización computacional* debería pasar a ser un componente crítico de la educación y conociendo los problemas que se presentan en los cursos introductorios de programación —por haber participado, dos años antes, del ya mencionado estudio liderado por McCracken—, el *Georgia Institute of Technology* utilizó en 2003 por primera vez el MediaComp (*Media Computation*), un nuevo enfoque para la enseñanza de la programación que luego sería adoptado por muchas otras instituciones educativas alrededor del mundo. Mark Guzdial lo describe así:

Creemos que la programación y la computación formarán parte de una educación general y liberal, pero los cursos de informática tendrán que cambiar para que esto suceda. Estamos explorando un enfoque de computación con medios —los estudiantes aprenden a programar escribiendo código en Python para manipular sonidos, imágenes y películas—, el cual atraerá al estudiante de artes liberales, sin perder el foco en la programación (Guzdial, 2003, p. 5).

En realidad, esas ideas ya habían sido llevadas a la práctica dos años antes en el *MIT Media Lab* por Casey Reas y Benjamin Fry, quienes consideran que la computadora, que surgió originalmente como una herramienta para cálculos rápidos, se ha convertido en un medio para la expresión (Reas y Fry, 2007, p. 3). En 2001, ellos crearon Processing, un lenguaje de programación textual diseñado específicamente para generar y modificar imágenes. Processing no es solamente una herramienta para dibujar, ya que uno de sus principales objetivos es que sus usuarios aprendan programación (p. 1).

En 2003, después de haber usado —durante apenas 4 años— el lenguaje C++ en el examen *Advanced Placement Computer Science*, la organización *College Board* lo dejó de lado y empezó a evaluar los conocimientos de programación utilizando el lenguaje Java («The College Board», 2017).

También en 2003, Hernando Barragán inició en el *Interaction Design Institute Ivrea* —una escuela de posgrado que funcionó en Italia desde 2001 hasta 2005— el desarrollo de la plataforma Wiring, fuertemente inspirada en Processing y basada en principios de código abierto. Su pequeña tarjeta de Entrada/Salida es una computadora autónoma y barata con muchas capacidades de conexión, que se puede utilizar para controlar todo tipo de sensores y actuadores: los sensores permiten que la placa adquiera información del entorno circundante, mientras que los actuadores permiten que la placa cree cambios en el mundo físico (Cvijikj y Michahelles, 2011, p. 77).

Dos años después, Massimo Banzi —uno de los directores de la tesis de maestría de Barragán—, junto con David Cuartielles, David Mellis, Tom Igoe y Gianluca Martino crearon el proyecto Arduino como una bifurcación de Wiring. La disponibilidad de Arduino —basado en código y *hardware* abiertos— permitió que en muchas instituciones educativas, además del desarrollo del *software*, se pasara a enseñar también el desarrollo del *hardware*, pues Arduino hace que las habilidades iniciales sean fáciles de adquirir, a la vez que mantiene las complejidades de las aplicaciones del mundo real (R. Anderson y Cervo, 2012, p. xxiii).

En 2004, la *informática física* ganó visibilidad con el libro *Physical computing: sensing and controlling the physical world with computers*, en el cual sus autores proclaman que la

forma en que las personas se comunican con las computadoras —usando los dedos— es demasiado limitada frente a lo que podría ser realmente:

Steve Jobs . . . se propuso construir *computadoras para el resto de nosotros*. La idea era permitir a las personas que no eran expertos en informática —como artistas, educadores y niños— aprovechar el poder de la informática. Ahora necesitamos hacer *computadoras para el resto de ti*. Necesitamos computadoras que respondan al resto de tu cuerpo . . . La tecnología GUI te permite *arrastrar y soltar*, pero no se dará cuenta si *giras y gritas*. Ha hecho que sea más fácil abrir una carpeta e iniciar un programa, pero nos gustaría que una computadora pueda abrir una puerta o arrancar un automóvil (O’Sullivan y Igoe, 2004, p. xvii).

También en 2004, Randy Pausch, Wanda Dann y Stephen Cooper publicaron el libro *Learning to Program with Alice*. Además de presentar el uso pedagógico de Alice como plataforma educativa para la enseñanza de la programación, el libro estaba acompañado de una versión completamente renovada del sistema, a tal punto que incluso el lenguaje de programación utilizado era otro:

En lugar de usar Python como lenguaje de programación, el equipo detrás de Alice 2.0 usó Java. Python era, y sigue siendo, un gran lenguaje para los principiantes, pero Java fue elegido en gran parte porque puede ejecutarse en más sistemas operativos que Python. Así que los creadores optaron por hacer que el entorno esté disponible para más usuarios en lugar de seguir con un lenguaje que tenía una curva de aprendizaje menos pronunciada (Nagle, 2015, p. 37).

Ese mismo año, Valentina Dagiene —de la *University of Vilnius* (Lituania)— lanzó el Desafío Bebras, que empezó en las escuelas primarias y secundarias como un concurso de informática nacional y, después de más de una década de crecimiento, se ha convertido en un gran evento internacional que tiene como objetivo involucrar a los estudiantes en la informática y promover el aprendizaje mediante la resolución de tareas cortas basadas en conceptos de computación. Las tareas son un punto crucial del desafío: deben incluir al menos un concepto de informática, envuelto en una historia interesante, y ser atractivas, breves y dinámicas (Dagiene, Stupuriene, Vinikiene y Zakauskas, 2017, p. 232).

Entre 2000 y 2004, el número de los ingresantes a las instituciones superiores de América del Norte que indicaban que se especializarían en ciencias de la computación había disminuido en más del 60 por ciento, llegando incluso a ser un 70 por ciento más bajo que su pico a principios de la década de 1980 (Vegso, 2005).

En este contexto, Pausch publicó en 2005 —esta vez, junto con Caitlin Kelleher— una completa taxonomía de entornos y lenguajes para programadores novatos. En esta, los sistemas están organizados de acuerdo con su objetivo principal, ya sea enseñar programación o utilizar la programación para empoderar a sus usuarios y, a continuación, según el enfoque que cada autor del sistema emplea para hacer que el aprendizaje sea más fácil para los programadores novatos (Kelleher y Pausch, 2005, p. 83).

Al mismo tiempo, Michael Kölling —quien después de desarrollar BlueJ en Australia se había mudado al Reino Unido y trabajaba en la *University of Kent*— estaba diseñando el sistema Greenfoot, que sería publicado un año más tarde y estaba orientado a estudiantes de la Educación Secundaria, como relata aquí:

BlueJ en ese momento tenía una base de usuarios significativa en las escuelas, principalmente en los últimos 2 o 3 años de la educación escolar: aproximadamente una cuarta parte de nuestros usuarios estaban en las escuelas, el resto principalmente en las universidades. Sin embargo, . . . sentíamos que este no era el sistema correcto para enseñar programación en las escuelas (Kölling, 2015, p. 17). La diferencia más importante entre los dos grupos —la escuela secundaria y el primer año de la universidad— es que este último es un grupo selectivo. Todos los estudiantes han elegido estar allí, por lo que se puede asumir un cierto nivel de motivación . . . Esto no es cierto en la escuela secundaria. La programación allí se presenta a toda la población, y muchos . . . no están interesados . . . Esto significa que el primer y más fundamental problema de una herramienta de programación dedicada a este grupo de edad no es enseñarles algo, sino generarles interés en la materia (Kölling y Henriksen, 2005, p. 1).

También en 2005, Arvid Halma lanzó RoboMind, un entorno para aprender conceptos básicos de programación. Al igual que en Karel —que había sido publicado casi un cuarto de siglo antes—, los estudiantes deben programar un robot para que se desplace por una grilla 2D dibujando y superando obstáculos pero, en este caso, en lugar de Pascal tienen que utilizar Robo, un lenguaje específico de este entorno (Halma, 2007, p. v).

En 2006, Idit Harel —en aquel entonces directora de la *World Wide Workshop Foundation*— lanzó Globaloria, una plataforma en línea orientada a que estudiantes de primaria y secundaria aprendan a crear juegos y simulaciones educativas para la Web («Globaloria», s. f.).



Ese mismo año, Jeannette Wing publicó en *Communications of the ACM* su artículo *Computational Thinking* (Pensamiento Computacional), en el que manifiesta que todas las personas se beneficiarían de aprender a pensar como un científico de la computación (Wing, 2006, p. 33). El impacto que logró fue tan grande que la computación —y en especial la programación— de a poco volvieron a encontrar un lugar en las escuelas primarias y secundarias de todo el mundo.

Basándose en las ideas de Papert y tras 4 años de trabajos, en 2007 el grupo *Lifelong Kindergarten* del *MIT Media Lab* —dirigido por Mitchel Resnick— lanzó Scratch, un entorno que proporciona una interfaz intuitiva que hace que aprender a programar sea relativamente fácil, como explica Badger (2009):

Con Scratch, programar es tan fácil como arrastrar y soltar bloques de una parte de la pantalla a otra . . . Usando Scratch, aprendemos cómo diseñar, pensar, colaborar, comunicar, analizar y programar en un lenguaje informático. Muchas de estas son habilidades de aprendizaje del siglo XXI (pp. 1-6).

Scratch ha servido de inspiración y base para muchos otros entornos posteriores de programación gráfica *drag and drop* (de arrastrar y soltar bloques). En el sitio web oficial de Scratch, los usuarios pueden cargar sus proyectos, así como también ver los proyectos de otros usuarios y comunicarse con ellos a través de comentarios y publicaciones en los foros. Hacia fines de 2017, la cantidad de usuarios registrados superaba los 25 millones —con una edad promedio que no alcanzaba los 14 años—, se habían compartido más de 29 millones de proyectos, y el número de comentarios publicados superaba los 145 millones («Scratch - Imagine, Program, Share», s. f.).

En 2008, Danny Yaroslavski publicó LightBot, otro de los varios entornos que existen para aprender programación a través del manejo de un robot:

Los usuarios deben navegar un robot en pantalla alrededor de un laberinto de cubos y encender luces . . . LightBot comienza presentando a los usuarios los principios básicos de programación para que el robot se mueva, gire, salte y encienda luces. A medida que el usuario progresa a través de los niveles o desafíos, se incluyen conceptos más complejos como procedimientos y bucles para niños mayores o aquellos que están listos . . . La programación es completamente visual y el usuario puede adaptar y probar algoritmos a lo largo del camino (D. Morris, Uppal y Wells, 2017, p. 53).

En 2009, los laboratorios FUSE(laboratorios de Microsoft Research) (*Future Social Experiences*) de Microsoft Research publicaron Kodu. Según consta en la página web del proyecto, este IDE

les permite a los niños crear juegos en la PC y la Xbox a través de un lenguaje de programación visual simple. Kodu se puede utilizar para enseñar creatividad, resolución de problemas, narración de cuentos y programación. Cualquiera puede usar Kodu para hacer un juego, incluso niños pequeños y adultos sin habilidades de diseño o programación («Kodu | About», s. f.).

Hacia el final de la década, la educación en informática en las escuelas secundarias estadounidenses casi no recibía atención. Como resultado, la programación había desaparecido casi por completo de los planes de estudios. En las escuelas a menudo se enseñaba poco más que el manejo del teclado, la navegación web y el uso de aplicaciones (Repenning, Webb y Ioannidou, 2010, p. 265).

Ante tal panorama, estos autores decidieron dar inicio a un proyecto para aumentar el interés por la programación, siguiendo un enfoque basado en módulos de complejidad creciente y que denominan *Scalable Game Design*:

Nuestro proyecto iDREAMS (*Integrative Design-based Reform-oriented Educational Approach for Motivating Students*) explora una serie de dimensiones pragmáticas relacionadas con el pensamiento computacional y sobre cómo llevar la educación en informática a las escuelas públicas. El proyecto comenzó en 2009 con el objetivo de proporcionar experiencias de diseño y programación de juegos a más de 2000 estudiantes durante 3 años (p. 266).

Finalmente, a partir de 2010 la situación comenzó a revertirse. Sin duda, los avances tecnológicos ocurridos tuvieron su influencia en ello, ya que, para entonces, otros dispositivos capaces de conectarse a la Web y ejecutar *software* coexistían en el mercado con las computadoras personales —el iPhone había sido lanzado por Apple en 2007, los primeros teléfonos con Android salieron a la venta en 2008, el mismo año en que Samsung lanzó su primer Smart TV, y la tableta iPad de Apple estuvo disponible en 2010— y además muchos entornos ya podían correr totalmente en línea —en cualquier navegador web— por lo que su implementación en las escuelas sería mucho más sencilla.

### 1.1.6. Década de 2010: La programación retorna a las aulas

En 2010, un grupo formado por organizaciones sin fines de lucro y por empresas tecnológicas se unió en una causa común (lograr que la informática tuviera un lugar destacado como una materia central en todos los planes de estudio de la educación obligatoria) y creó la coalición *Computing in the Core*. La importancia de esta alianza queda en evidencia al ver la lista de sus integrantes, mencionados por Aspray (2016):

Los miembros originales fueron las asociaciones profesionales informáticas ACM y *Computing Research Association*, la organización docente CSTA (*Computer Science Teachers Association*), las organizaciones de mujeres en informática NCWIT (*National Center for Women & Information Technology*) y el *Anita Borg Institute*, y las compañías informáticas Google, Microsoft y SAS . . . Otras organizaciones se unieron más tarde, incluyendo el *National Council of Teachers of Mathematics* (Consejo Nacional de Maestros de Matemáticas) y la *National Science Teachers Association* (Asociación Nacional de Maestros de Ciencias). En 2011, después de una reunión entre los directores ejecutivos y los presidentes de la ACM y la *IEEE Computer Society*, esta última también se unió a la coalición (p. 131).

De 2010 a 2012, la coalición *Computing in the Core* estuvo a cargo de la organización de la CSEdWeek (*Computer Science Education Week*), un evento anual que tiene como objetivo despertar en los estudiantes el interés por las ciencias de la computación («About CSEdWeek», s. f.).

Después de 4 años de experiencia enseñando programación con Greenfoot, en 2010 Michael Kölling publicó el libro *Introduction to programming with Greenfoot: object-oriented programming in Java with games and simulations*, el cual se diferenciaba —por su enfoque lúdico— de otros libros introductorios clásicos:

Los capítulos y ejercicios están estructurados en tareas de desarrollo reales y prácticas. Primero, hay un problema que tenemos que resolver, luego observamos los elementos del lenguaje y las estrategias que nos ayudan a resolver el problema. Esto es bastante diferente de muchos libros de programación introductoria que a menudo están estructurados alrededor de los elementos de los lenguajes . . . No hay ninguna razón por la cual la programación de computadoras deba ser seca, formal o aburrida. Divertirse en el camino está bien. Creemos que podemos lograr que la experiencia sea interesante y pedagógicamente sólida al mismo tiempo. Este es un enfoque que hemos llamado *diversión seria*: hacemos algo interesante y aprendemos algo útil en el camino (Kölling, 2010, p. 1).

Ese mismo año, Google lanzó App Inventor, un nuevo entorno de desarrollo visual para dispositivos móviles con Android, dirigido a usuarios con poca o ninguna experiencia en programación. El líder técnico del proyecto, Mark Friedman, describió así el impacto que tuvo esta herramienta al ser probada por primera vez:

Cuando Hal Abelson me contó sobre la idea que después se convertiría en App Inventor, hablamos sobre la fuerza motivadora única que los teléfonos celulares podrían tener en la educación . . . Cuando construimos App Inventor y lo probamos en las clases de Dave Wolber, empezamos a darnos cuenta de que algo aún más poderoso estaba sucediendo: App Inventor estaba convirtiendo a los estudiantes de consumidores en creadores. ¡A los estudiantes les parecía divertido y emocionante crear aplicaciones para sus teléfonos celulares! (Wolber, Abelson, Spertus y Looney, 2011, pp. xiii-xiv).

En 2011, Alexander Repenning presentó un nuevo modelo de interacción entre los programadores y los entornos de programación: *Conversational Programming* (programación conversacional), que fue incorporado en AgentSheets y se puede conceptualizar como una forma simple de programación por pares que reemplaza a uno de los programadores humanos por un agente computacional. La programación conversacional no se limita a la retroalimentación sintáctica, sino que también proporciona una rica información semántica sobre los programas, ya que constantemente los ejecuta y los marca con anotaciones (Repenning, 2011, pp. 191-194).

También en 2011, Zach Sims y Ryan Bubinski lanzaron *Codecademy*, una plataforma interactiva en línea para aprender lenguajes como JavaScript, PHP, Python, Ruby y HTML/CSS, ya sea de manera gratuita o pagando un arancel para hacerlo con mayor intensidad. A comienzos de 2018, los usuarios registrados habían superado los 45 millones («Codecademy - learn to code, interactively, for free», s. f.). A pesar de su enorme popularidad, algunos autores como Steven Foote (2015) advierten que, para saber programar, es necesario aprender mucho más que la mera escritura de código:

Un curso de Codecademy proporciona un editor de texto dentro de tu navegador donde puedes escribir tu código y leer los resultados. Sin embargo, . . . como no tienes instalado el entorno de ejecución, solo puedes usar las habilidades aprendidas mientras te encuentres en el sitio web de Codecademy. En otras palabras, Codecademy te enseña a escribir código pero no te enseña a programar (p. 251).

Sin embargo, desde 2011 no ha parado de crecer la cantidad de plataformas interactivas en línea que incluyen cursos para aprender lenguajes de programación en su oferta. Mientras que en algunas de estas plataformas solamente se ofrecen cursos de programación —como en *CodeHS*, *Treehouse*, *Bloc*, *Code Avengers*, *freeCodeCamp*, *Code School*, *Tynker* y muchas más—, en otras los lenguajes de programación son apenas una parte de la oferta total —como en *Khan Academy*, *Coursera*, *edX*, *Udacity*, *Udemy*, *Lynda.com* y *MIT OpenCourseware*, entre otras.

Además, para reducir la brecha entre el número de cargos de ingeniería de *software* disponibles y la insuficiente cantidad de candidatos calificados para llenarlas, en 2011 surgieron en los Estados Unidos los primeros campos de entrenamiento de codificación (*Coding Bootcamps*): *Code Academy* (después rebautizado como *The Starter League*) en Chicago, *Dev Bootcamp* y *Hack Reactor* en San Francisco, *General Assembly* en Nueva York y muchos más. En otros países surgieron iniciativas similares, como por ejemplo: la *Barcelona Code School* en España y la *Elium Academy* en Bélgica.

En estas instituciones es posible capacitarse en programación en un corto periodo, lo que ha llevado a algunos autores a considerar que la programación no es más que un oficio, al contrario de la informática, que es un campo académico:

Las carreras ofrecidas en estos campos de entrenamiento rechazan la sabiduría convencional de que una profesión tan cerebral como la programación requiere de una base de estudio abstracto. Aunque la programación siempre ha tenido una fuerte cuota de autodidacticismo, esta tendencia amplía y formaliza la ética de los *hackers* de valorar la capacidad y los logros por sobre las certificaciones y los títulos. Quizás en cierto sentido refuerce la distinción entre la programación como un oficio y la informática como un campo académico (Moon, 2012).

El futuro de los campos de entrenamiento de codificación es incierto. Tras el cierre de *Dev Bootcamp* —uno de los pioneros— en 2017, *The New York Times* publicó un artículo en el que explicaba que la demanda de los empleadores estaba cambiando y las instituciones educativas deberían adaptarse. Muchos campos de entrenamiento no han evolucionado para ofrecer más que cursos de desarrollo web básico, pero lo que las empresas a menudo están buscando son habilidades de codificación más avanzadas (Lohr, 2017).

En 2012, fue lanzada en el Reino Unido la primera computadora Raspberry Pi. Membrey y Hows (2015) explican la situación que motivó la creación de esta revolucionaria computadora de placa única:

Algún tiempo atrás, la gente sabía cómo sacar el mejor provecho de la máquina porque realmente sabía cómo funcionaba . . . Hoy las empresas se encuentran con un problema. La gente que realmente conoce las computadoras por dentro es cada vez más difícil de encontrar . . . ¿Cómo se llegó a esto? Bueno, las . . . computadoras pasaron de ser una curiosidad a ser una parte básica de la vida de todos . . . La gente tiene una idea aproximada sobre cómo encenderlas, instalar *software* y demás, pero la forma en que realmente funcionan no se considera relevante . . . Eben Upton y la *Raspberry Pi Foundation* están tratando de . . . hacer que las computadoras sean interesantes nuevamente. Para ello, han creado una computadora que —incluso para los estándares de hoy en día— no se queda atrás. ¿Es tan poderosa como tu computadora portátil? Bueno, no, pero ¿puedes comprar una *laptop* por 35 dólares? (p. xxiii).

La *informática física* alcanzó —gracias a la disponibilidad de microcontroladores como el Arduino y de computadoras de placa única como la Raspberry Pi— una enorme difusión dentro y fuera de los sistemas educativos, provocando que el aprendizaje de la programación ganara un nuevo atractivo, como se sugirió en *Digital Diner*:

Si la fiebre inventora toca a tu puerta, te recomendamos que la dejes entrar. Puede ser increíblemente gratificante. Si tienes un conocimiento mínimo de la programación, existen sitios web, plataformas y herramientas que te ayudarán a desarrollar tus habilidades para que puedas crear un dispositivo de *hardware* con botones, perillas y motores . . . El *software* es divertido, pero cuando puedes hacer que tu proyecto sea físico, es aún mejor (Meike, 2012).

En 2013, la *BeagleBoard.org Foundation* lanzó su computadora BeagleBone Black, un sistema de *hardware* abierto que combina en una sola placa las capacidades de una Raspberry Pi y de un Arduino, al sorprendente precio de 45 dólares (Verry, 2013).

También en 2013, Shuchi Grover y Roy Pea —de la *Stanford University*— publicaron su artículo *Computational Thinking in K–12: A Review of the State of the Field*, donde exploran las causas de la resonancia que tuvo a partir de 2006 el célebre artículo de Jeannette Wing acerca del *pensamiento computacional*, estudian cómo ha sido interpretada su definición y qué avances se han realizado desde su publicación, identifican lagunas en la investigación y articulan las prioridades para futuras investigaciones (Grover y Pea, 2013).

Ese mismo año, la coalición *Computing in the Core* delegó en Code.org —una organización sin fines de lucro estadounidense— la coordinación de la CSEdWeek. Desde entonces, esta se lleva a cabo en torno a una nueva idea: la *Hora del Código*, un desafío en línea orientado a enseñar programación mediante tutoriales cortos y divertidos. En el informe anual de Code.org correspondiente al año 2017, se indica que

en solo cuatro años, 25 países, 40 de los 50 estados que integran los Estados Unidos y casi 200 ciudades estadounidenses han anunciado planes para expandir el acceso a la informática en la educación primaria y secundaria; . . . alrededor de 750000 maestros han empezado a utilizar Code.org para dar clases de informática a más de 25 millones de estudiantes en todo el mundo; . . . y la *Hora del Código* ha sido usado más de 500 millones de veces, alcanzando a uno de cada 10 estudiantes en el planeta («Code.org 2017 Annual Report», s. f.).

Casi al mismo tiempo, fue lanzada —promovida por la Comisión Europea— la *Semana Europea de la Programación* (*EU Code Week*),

una iniciativa popular organizada por voluntarios que celebra la creación con código. La idea es hacer que la programación sea más visible, mostrarles a los jóvenes, adultos y ancianos cómo se hacen realidad las ideas con código, desmitificar estas habilidades y reunir personas motivadas para aprender. La convocatoria saltó de 10000 participantes en la primera edición —en 2013— a 968000 en la cuarta edición —en 2016— («About EU Code Week», s. f.).

Otro hecho de relevancia ocurrido en 2013 fue el lanzamiento del sitio web Pencil Code, un desarrollo liderado por David Bau —un programador de Google—, inspirado en el lenguaje Logo —del cual tomó los gráficos de tortuga— y utilizado desde entonces por Google en sus programas de divulgación de la informática. Pencil Code es un sitio de programación colaborativa para dibujar arte, reproducir música y crear juegos. También es un lugar para experimentar con funciones matemáticas, geometría, gráficos, páginas web, simulaciones y algoritmos. Los programas están abiertos para que todos los vean y copien, ya que Pencil Code es un proyecto de código abierto. El lenguaje principal es Coffeescript («Pencil Code», s. f.).

En 2014, otro derivado del Logo llegó a la Web: el StarLogo Nova. Desarrollado en el marco del *Scheller Teacher Education Program (STEP)* del MIT, el StarLogo Nova combina la programación visual mediante *drag and drop* con el modelado 3D, y corre en cualquier navegador web sin necesidad de una instalación previa («StarLogo Nova», s. f.).

Ese mismo año, el *MIT Media Lab*, la *Tufts University* y la *Playful Invention Company* lanzaron ScratchJr, una versión de Scratch adaptada para que los niños que recién están aprendiendo a leer también puedan aprender a programar, como se proclama en el sitio oficial de esta herramienta:

¡La programación es la nueva alfabetización! Con ScratchJr, niños (entre 5 y 7 años) pueden programar sus propias historias y juegos interactivos. En el proceso, aprenden a resolver problemas, diseñan proyectos y se expresan de forma creativa en un ordenador («ScratchJr», s. f.).

En 2015, con fondos del programa *Horizon 2020* de la Unión Europea, se dio inicio —con pruebas piloto en Austria, España y el Reino Unido— al proyecto *No One Left Behind* (Nadie se queda atrás), para que los niños desarrollen juegos digitales en dispositivos móviles, con el objetivo de mejorar sus capacidades en todas las materias académicas, así como sus habilidades lógicas, creativas y sociales. Se ha desarrollado para ello la aplicación *Create@School*, derivada de *Pocket Code*, adaptada para entornos educativos. Con esta aplicación, los estudiantes y los profesores pueden programar juegos y crear animaciones. Los elementos de diseño del juego son fáciles de seleccionar y de comprender («No One Left Behind», s. f.).

Lanzado dos años antes, *Pocket Code* es un entorno de programación desarrollado por el equipo de Wolfgang Slany —de la *Technische Universität Graz* (Austria)— y disponible para dispositivos móviles. El lenguaje usado en *Pocket Code* es *Catrobat*, que es un lenguaje de programación visual y un conjunto de herramientas de creatividad para teléfonos celulares inteligentes, tabletas y navegadores móviles. A través de *Catrobat*, es posible promover las habilidades del pensamiento computacional y la filosofía del *software* libre y de código abierto, de una manera sencilla, divertida y atractiva, a escala mundial. *Catrobat* y el *software* desarrollado por el equipo de *Catrobat* están fuertemente inspirados en el sistema de programación Scratch desarrollado por el *Lifelong Kindergarten Group* en el *MIT Media Lab*. *Catrobat* es un proyecto independiente de *software* libre y de código abierto («*Catrobat*», s. f.).



En 2016, se anunció en la República Argentina el Plan 111 Mil, puesto en marcha a principios del año siguiente y que, según consta en su sitio web oficial,

es un plan nacional que busca formar en los próximos 4 años a 100000 programadores, 10000 profesionales y 1000 emprendedores. El objetivo es cubrir la demanda laboral de las Industrias Basadas en el Conocimiento, uno de los sectores que más crece y exporta en Argentina . . . El curso dura dos cuatrimestres y . . . se dicta en escuelas técnicas, centros de formación profesional y universidades . . . El plan te capacita y certifica para que puedas trabajar en las mejores empresas del sector. Te da herramientas de desarrollo de *software* para diferentes usos y destinatarios. Hay más de 45000 empresas que te están esperando . . . El certificado que brinda 111 Mil tiene validez nacional y está avalado por el Ministerio de Educación y el Ministerio de Producción («Plan 111mil», 2016).

Al ser implementado en la Ciudad Autónoma de Buenos Aires, el Plan 111 Mil fue denominado Programa *Codo a Codo*. Se estableció que la capacitación sería gratuita, estaría abierta para cualquier persona mayor de 18 años que tuviera título secundario, tendría una duración de 9 meses y se podría cursar de manera presencial o en línea. Sobre los contenidos que se dictarían, se publicó la siguiente descripción:

El trayecto contempla módulos de aprendizaje elementales para poder comprender cualquier lenguaje de programación, siendo Java el que será utilizado para exponer los conceptos y ejercitar los trabajos prácticos. También se verán aspectos del ámbito laboral para una incorporación exitosa en la industria («Programa Codo a Codo», s. f.).

También en 2016 fue el lanzamiento de Mumuki, una plataforma argentina libre y gratuita para aprender y enseñar a programar que “encuentra automáticamente errores a los ejercicios . . . y lleva a cabo estadísticas” («Mumuki», s. f.).

## 1.2. La programación orientada a objetos

En la POO (*Programación orientada a objetos*), el paradigma que más vigencia tiene actualmente, en vez de un procesador de bits consumiendo estructuras de datos, tenemos un *universo de objetos* bien comportados, cada uno de ellos pidiéndole a otro que cortésmente le realice sus variados deseos (Ingalls, 1981, p. 290). Como un ejemplo concreto de esto, Fontela (2003) plantea utilizar el UML (*Unified Modeling Language*) para realizar el *diagrama de secuencia* del proceso de anotación de un estudiante en las materias de un posgrado, estableciendo para ello la siguiente consigna:

Se supondrá que cada alumno podrá anotarse en varios cursos, para los cuales habrá que verificar que tenga aprobadas las correlativas. En una segunda etapa, el profesor del curso y el coordinador del posgrado evaluarán la situación del alumno y aprobarán o no la solicitud, dado que pueden considerar que la orientación general que el aspirante le está dando a sus estudios no debería orientarlo en esa dirección (p. 109).

El diagrama de secuencia que muestra cómo los diferentes objetos interactúan en este sistema se muestra en la figura 1.

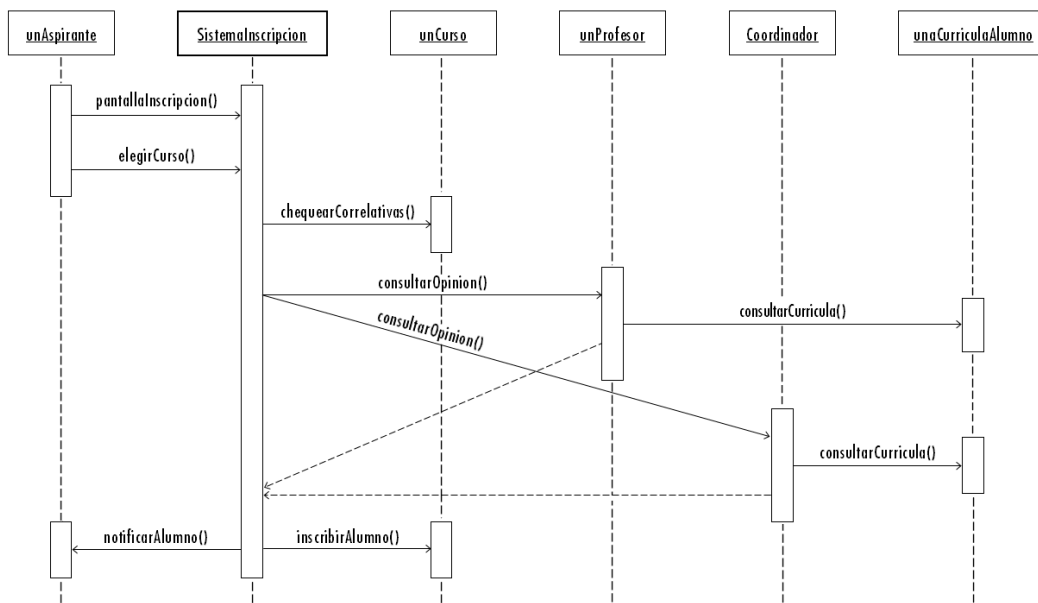


Figura 1. Diagrama de secuencia para un sistema de inscripción a un posgrado. Adaptado de *Programación Orientada a Objetos. Técnicas Avanzadas de Programación* por C. Fontela, 2003, p. 111. Buenos Aires: Nueva Librería. Copyright 2003 por Nueva Librería SRL

Los cuatro pilares de la POO son la abstracción, el encapsulamiento, la herencia y el polimorfismo. Es por ello que, para programar un sistema en este paradigma, primero deberá emplearse el primer pilar (la *abstracción*) para definir qué objetos serán necesarios y establecer cómo se irán comunicando entre sí y con los usuarios cuando el sistema esté funcionando. La abstracción es una simplificación que incluye solo aquellos detalles relevantes para determinado propósito y descarta los demás. Así, cada objeto se define a través de una terna de *identidad* (el nombre de la variable con la que se hace referencia al objeto), *estado* (el valor de los atributos del objeto) y *comportamiento* (las funciones o métodos que el objeto ejecuta cuando recibe solicitudes).

Definir cuáles partes del estado y del comportamiento de un objeto podrán ser accedidos desde otros objetos es lo que se determina empleando el segundo pilar de la POO (el *encapsulamiento*). Por lo general, lo deseable es que el estado de un objeto no pueda manipularse directamente desde otros objetos y que su comportamiento sí pueda verse y, por lo tanto, solicitarse. En Java, por ejemplo, existen cuatro formas de indicar la accesibilidad en una declaración: *public*, *private*, *protected* y sin especificar.

En la mayoría de los lenguajes orientados a objetos, pueden definirse *clases* (declaraciones de atributos y métodos comunes) a partir de las cuales después es posible crear los objetos. Por ejemplo, si en un sistema se declara la clase *Boton*, posteriormente será posible crear (o *instanciar*) los objetos *btnAbrir*, *btnGuardar*, *btnSalir*, etc., como se muestra en la figura 2.

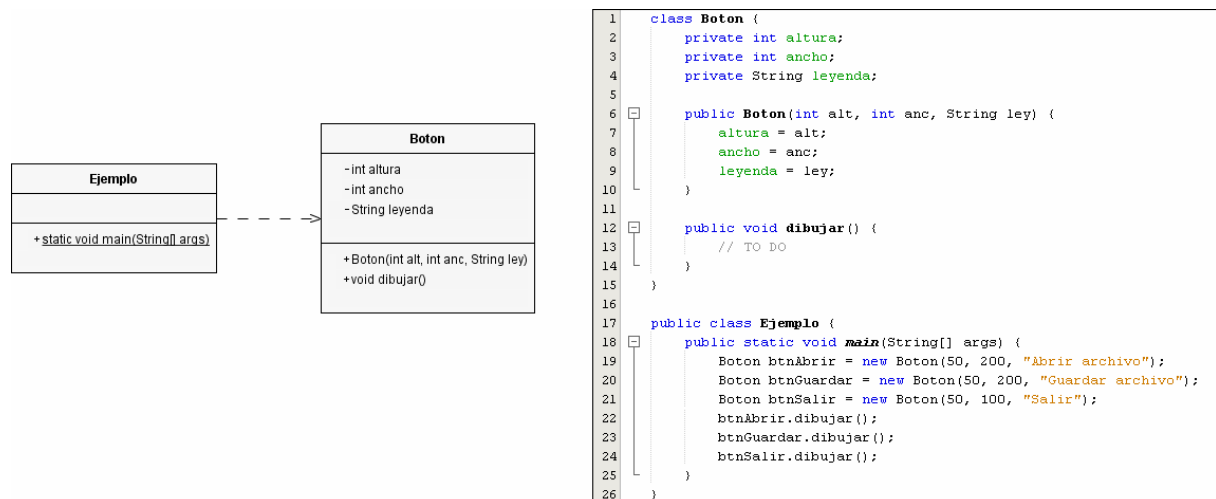


Figura 2. La clase *Boton* en UML y su utilización para crear los objetos *btnAbrir*, *btnGuardar* y *btnSalir* en Java

En el ejemplo anterior, la identidad el primer objeto que aparece es *btnAbrir* (creado en la línea 19 del programa), su estado está dado por los valores de los atributos privados *altura*, *ancho* y *leyenda* (50, 200 y "Abrir Archivo", respectivamente) y su comportamiento es el método público *dibujar()* cuya ejecución se le solicita en la línea 22.

Cuando al definir una clase se hace mención a otra clase, se establece una relación entre ambas. Las seis principales relaciones son la *generalización*, la *realización*, la *asociación*, la *agregación*, la *composición* y la *dependencia* (figura 3), cada una de las cuales corresponde a un tipo distinto de flecha en el *diagrama de clases* en UML, mediante el cual se suelen representar las clases y sus relaciones (Rumbaugh et al., 2000, pp. 41-52).

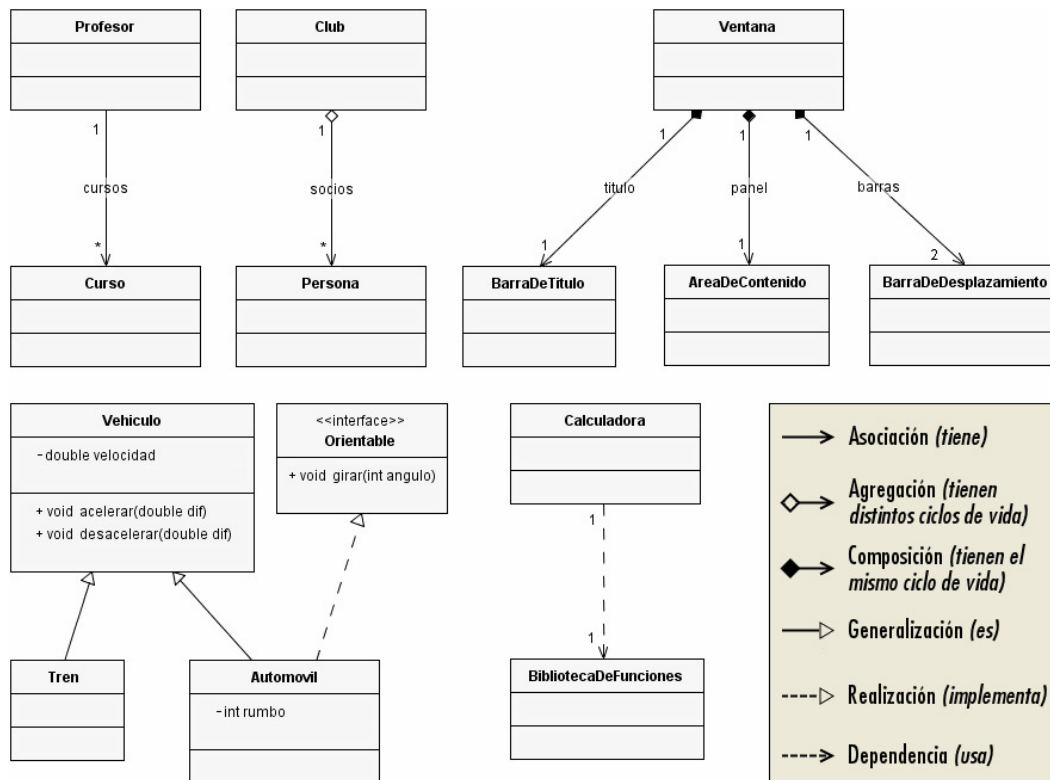


Figura 3. Principales relaciones entre clases

El tercer pilar de la POO (la *herencia*) no es más que otro nombre para la primera de las relaciones anteriores: la generalización. Significa que si una clase tiene este tipo de relación con otra, los objetos de la clase más específica heredarán ciertos atributos y métodos de la clase más general. Uno de los principales propósitos de la herencia es aprovechar parte del código existente. En Java, se utiliza la palabra reservada *extends* para indicar que una clase deriva de otra.

El cuarto y último pilar de la POO (el *polimorfismo*) permite que una variable de una clase más general pueda contener una referencia a un objeto de cualquier clase más específica, derivada de la anterior. De esta manera, puede usarse una única variable para acceder sucesivamente a objetos de *muchas formas*, lo que reduce la cantidad de código necesario para desarrollar un sistema. Para que el polimorfismo funcione correctamente, cada clase derivada debe *sobrescribir* los métodos heredados de la clase más general. Por ejemplo, si la clase general *FiguraGeométrica* posee un método *calcularArea*, las clases derivadas *Triángulo* y *Cuadrado* deberán sobrescribir ese método para que se realice el cálculo correcto correspondiente a la figura en cuestión.

En Java, por una cuestión de eficiencia, existen *tipos primitivos* para trabajar con datos que no requieran ser modelados como objetos (figura 4). Según la cantidad de espacio que ocupen en la memoria, estos tipos primitivos permiten utilizar rangos de datos de distinta extensión. Por ejemplo, un dato del tipo *int* ocupa más espacio en la memoria que otro de tipo *short*, por lo cual su valor puede pertenecer a un rango más extenso. Cuando se requiere construir objetos a partir de datos primitivos, es posible usar clases de envoltorio (*wrapper classes*). Por ejemplo, a partir de un dato del tipo *int* se puede construir un objeto de la clase *Integer*. Así, el dato primitivo puede ser tratado como un objeto.

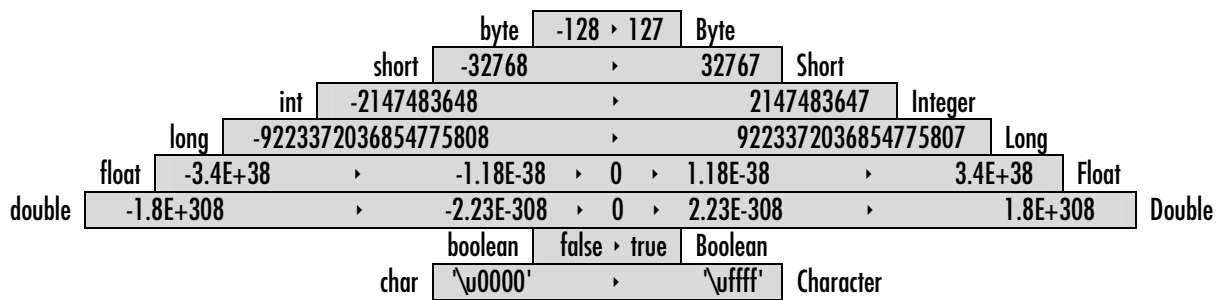


Figura 4. Tipos primitivos (izquierda), sus rangos y las clases de envoltorio correspondientes en Java (derecha)

Por último, cabe mencionar una característica muy útil presente en varios lenguajes: la *sobrecarga*. En Java, esta permite otorgarle diferentes funcionalidades a un mismo método, según de qué tipo sean los argumentos con los que se lo utilice. Por ejemplo, el método *partir* podría sobrecargarse para que, si recibe un número, lo divida por 2 y, en caso de recibir una cadena de caracteres, la separe en dos mitades.

### 1.3. Razones para la importancia de aprender programación

Como ya se mencionó al inicio de este capítulo, la importancia del aprendizaje de la programación es cada vez más evidente en todo el mundo, y una prueba de ello es el hecho de que actualmente numerosos países la contemplan en su legislación. Guzdial (2016, pp. 3-10) señala para este fenómeno las seis razones que se explican a continuación.

#### 1.3.1. El mercado laboral demanda programadores

La digitalización del mundo contemporáneo sería imposible sin la actuación de un ejército de programadores encargados de desarrollar y mantener el *software*. Actualmente,

todos los países tienen una gran dificultad para cubrir esta demanda. La República Argentina no es una excepción. Según una encuesta del *Observatorio Permanente Software y Servicios Informáticos* (OPSSI) de la *Cámara de la Industria Argentina del Software* (CESSI), de las 7800 plazas disponibles para cubrir puestos claves como ingenieros de *software* y analistas de sistema en 2016, el 65% quedaron vacantes (Bellucci, 2017).

En los Estados Unidos —que es la sede de las mayores empresas de tecnología informática del mundo— los números son mucho más alarmantes:

Una proyección de la *Oficina de Estadísticas Laborales de Estados Unidos* prevé que para 2020 habrá 1,4 millones de empleos en la industria de las ciencias computacionales, un sector cuyo crecimiento pronosticado es del doble del promedio nacional. Sin embargo, ese año solo se espera que se gradúen unos 400000 estudiantes de ciencias computacionales, creando una brecha de talento de aproximadamente un millón de puestos (Bellucci, 2017).

Según Guzdial (2016), la enseñanza de la programación en los colegios puede ayudar a desarrollar la motivación para seguir una carrera en computación (p. 3).

### **1.3.2. La informática forma parte de nuestras vidas**

Aprender a programar también es importante porque la informática forma parte de nuestras vidas. Guzdial (2016) lo justifica así:

Les pedimos a los estudiantes que estudien química porque viven en un mundo donde hay interacciones químicas. Les pedimos que estudien biología porque viven en un mundo vivo. También viven en un mundo computacional, y la informática probablemente impactará en sus vidas más que recordar la estructura de un anillo de benceno o las etapas de la mitosis (p. 4).

### **1.3.3. Al programar se ejercita el pensamiento computacional**

En la última década, el desarrollo del pensamiento computacional ha pasado a ser uno de los objetivos educativos más buscados. Wing (2006) fue enfática al afirmar que el pensamiento computacional y la programación no son lo mismo (p. 35), y dio varios ejemplos de conceptos computacionales presentes en la vida diaria:

Cuando tu hija va a la escuela por la mañana, pone en la mochila las cosas que necesita para el día; eso es *prefetching* y almacenamiento en caché. Cuando tu hijo pierde sus guantes, le sugieres que vuelva sobre sus pasos; eso es *backtracking* ... ¿En qué fila te pones en el supermercado? Eso es modelado de rendimiento para sistemas de servidores múltiples (p. 34).

Sin embargo, aunque esas actividades cotidianas son ejemplos interesantes, donde el pensamiento computacional se ejercita de la forma más directa es —sin lugar a dudas— al programar. Pea y Kurland (1984) mencionan siete cambios en el pensamiento que se producen al aprender a programar:

- 1) el pensamiento riguroso, la expresión precisa, la necesidad de hacer suposiciones explícitas . . .;
- 2) la comprensión de conceptos generales tales como las variables, los procedimientos formales y las funciones . . .;
- 3) una mayor facilidad con el arte de la *heurística*, para resolver problemas en cualquier dominio . . .;
- 4) la idea general de que la depuración de errores es una actividad constructiva y planificable, aplicable a cualquier tipo de resolución de problemas . . .;
- 5) la idea general de que uno puede inventar pequeños procedimientos como bloques de construcción para la construcción gradual de soluciones a problemas grandes . . .;
- 6) una mejor autoconciencia y alfabetización sobre el proceso de resolución de problemas. . .;
- 7) un mayor reconocimiento de que rara vez hay una única *mejor* manera de hacer algo, sino diferentes maneras que tienen ciertos costos y beneficios con respecto a objetivos específicos (p. 143).

#### **1.3.4. Aprender programación forma parte de la alfabetización computacional**

Guzdial (2016) compara los efectos de la masificación de las computadoras con los de la invención de la imprenta, ya que alcanzar la alfabetización computacional en la sociedad implica que las personas puedan leer y escribir a través del uso de la computación, lo cual incluye la capacidad de leer y escribir programas de computadora (p. 5). Además, plantea el objetivo de una sociedad computacionalmente alfabetizada:

El objetivo de una sociedad computacionalmente alfabetizada es poder utilizar la informática como una forma de expresión y una manera de pensar sobre dominios distintos de la informática. En muchos campos, la informática permite a las personas hacer cosas que no podrían hacerse sin ella. Tradicionalmente, la ciencia ha sido teórica o empírica, pero ahora también hay una tercera forma de hacer ciencia: la computacional. Hoy en día, los científicos usan el modelado y la simulación para llegar a comprender el mundo de una manera diferente (p. 6).

#### **1.3.5. Saber programación puede aumentar la productividad**

No es necesario tener los conocimientos de programación de un desarrollador de *software* profesional para sacarle provecho a la programación, ya que —incluso creando programas muy básicos para automatizar ciertas tareas repetitivas— es posible aumentar la

productividad, al reducir los tiempos y comprender mejor los procesos involucrados. Según Guzdial (2016):

La gente usa la informática no solo porque les ahorra tiempo, sino porque les permite pensar de nuevas maneras sobre sus dominios . . . Así como el proceso de escritura cambia la forma en que los autores entienden su conocimiento, el proceso de producción en el medio informático (la programación) también puede llevar al programador a transformar su comprensión (p. 7).

### 1.3.6. Aprender programación puede ayudar a reducir las brechas digitales

Surgido en la década de 1990 para referirse a la línea divisoria percibida entre quienes tienen acceso a las últimas tecnologías de la información y aquellos que no lo tienen (Compaine, 2001, p. xi), el término *brecha digital* en realidad abarca varias brechas, como claramente fue proclamado por Kofi Annan —ex Secretario General de las Naciones Unidas— en la Cumbre Mundial sobre la Sociedad de la Información:

La llamada *brecha digital* son en realidad varias brechas en una. Existe una *brecha tecnológica*: grandes diferencias en la infraestructura. Hay una *brecha de contenido*. Una gran cantidad de información disponible en la web simplemente no es relevante para las necesidades reales de las personas. Y casi el 70 por ciento de los sitios web del mundo están en inglés, a veces desplazando las voces y opiniones locales. Existe una *brecha de género*, donde las mujeres y las niñas disfrutan de menos acceso a la tecnología de la información que los hombres y los niños . . . Hay una *brecha comercial*. El comercio electrónico está uniendo cada vez más a algunos países y compañías. Pero otros corren el riesgo de una mayor marginación . . . Y existen obvias *disparidades y obstáculos sociales, económicos y de otra índole* que afectan la capacidad de un país de aprovechar las oportunidades digitales («Secretary-General's address to the World Summit on the Information Society», 2003).

Castaño Collado (2008) también sostiene que hay más de una brecha digital. Refiriéndose a la situación en la Unión Europea, ella afirma que las mujeres tienen menos oportunidades que los hombres en el campo de la informática:

El aumento sostenido del número de usuarios de ordenadores y de las conexiones a Internet parece indicar que la primera brecha digital puede resolverse en el futuro. La segunda brecha digital, relacionada con las habilidades necesarias para obtener todos los beneficios del acceso, afecta más a las mujeres que a los hombres . . . Los ingenieros y consultores de alto nivel en el campo de la informática, los creadores de *software*, son mayoritariamente hombres. Los trabajadores manuales y los que prestan los servicios masivos son mujeres (pp. 24-33).



Además de la cuestión del género, al describir la situación en los Estados Unidos, Guzdial (2016) también menciona el aspecto étnico:

La mayoría de las personas que se dedican a la informática en los EE. UU. son hombres descendientes de europeos (blancos/caucásicos) o de asiáticos . . . Menos del 20% de los estudiantes en programas de informática de pregrado en los EE. UU. son mujeres o miembros de otros grupos étnicos/raciales. Menos del 20% de los empleados técnicos en la industria informática son mujeres o miembros de otros grupos étnicos/raciales (p. 8).

Por ello, este autor afirma que necesitamos proporcionar a todos el acceso a la educación informática —con énfasis en la programación—, para que todos tengan la oportunidad de acceder a los trabajos mejor remunerados (p. 10).

## 1.4. Modelos didácticos para la programación

Disponer de un modelo didáctico claramente definido que establezca cómo llevar a cabo la enseñanza de la programación puede parecer —a priori— una buena idea. Sin embargo, en la práctica muchas veces no es posible imponer un único modelo didáctico, por lo que es importante conocer las diferentes posibilidades que existen. En la literatura, Kaasbøll (1998, p. 196) encontró tres modelos didácticos para la enseñanza de la programación: la escalera semiótica, la taxonomía de objetivos cognitivos y la resolución de problemas, los cuales se describen a continuación.

### 1.4.1. Escalera semiótica

En su obra *Fundamentos de la teoría de los signos*, publicada originalmente en 1938, C. W. Morris (1985) definió la *semiosis* como “el proceso en el que algo funciona como signo” (p. 27). Asimismo, señaló los elementos o factores que la componen: el *vehículo signico* (lo que actúa como signo), el *designatum* (aquello a que el signo alude), el *interpretante* (el efecto que produce en determinado intérprete en virtud del cual la cosa en cuestión es un signo para él) y el *intérprete* (p. 27). También estableció que las relaciones entre estos factores constituyen dimensiones que podrían ser estudiadas por disciplinas separadas. Así, la *semántica* es el estudio de las relaciones de los signos con los objetos a los que son aplicables, la *pragmática* es el estudio de la relación de los signos con los intérpretes, y la *sintaxis* es el estudio de la relación formal de los signos entre sí (p. 31-32).

Como consecuencia de los avances tecnológicos ocurridos en la segunda mitad del siglo XX, Stamper (1996) propuso un marco semiótico más completo —la *escalera semiótica* entre el mundo físico y el mundo social— al considerar tres dimensiones más: la dimensión *física* de los signos —que abarca los medios en los que se representan los signos y el *hardware* utilizado para transmitirlos y procesarlos—, la dimensión *empírica* de los signos —que trata de las propiedades estadísticas de los conjuntos de signos cuando se usan medios físicos y dispositivos diferentes— y la dimensión *social* en la que estos encuentran sus propósitos (p. 350-351).

Como un modelo didáctico para la enseñanza de la programación, la *escalera semiótica* ha sido implementada de diferentes maneras. Por ejemplo, Kaasbøll (1998) no menciona las dimensiones agregadas por Stamper:

Este enfoque de enseñanza se basa en las características —similares a un lenguaje— que tienen las herramientas informáticas . . . La secuencia de enseñanza y aprendizaje parte de la sintaxis, y pasa a la semántica y la pragmática de las herramientas . . . Su razón de ser es que el conocimiento sintáctico es necesario para expresar cualquier cosa, y por lo tanto debe preceder al aprendizaje del significado de los elementos del lenguaje. Una vez que conocen el significado de estos elementos, los estudiantes pueden aprender a usarlos para fines específicos, lo que constituye la pragmática (p. 196).

En cambio, May y Dhillon (2009) sí proponen utilizar todas las dimensiones —desde la física hasta la social— para analizar los lenguajes de programación (p. 433). Según estos autores, analizar los elementos de un lenguaje de programación basándose en la *escalera semiótica* podría facilitar el aprendizaje de los estudiantes novatos e incluso mejorar las habilidades de los programadores más experimentados:

Un instructor podría realizar un ejercicio en el aula donde el producto final sería una tabla organizada que les permita a los estudiantes realmente tener una comprensión enriquecida de algún elemento del lenguaje. Después se les podría pedir a los estudiantes que realicen análisis similares de otros elementos que podrían utilizarse como parte de una tarea de programación real. Argumentamos que forzar a los estudiantes a llevar a cabo este tipo de ejercicio facilitará una comprensión más profunda de los elementos del lenguaje de programación . . . Esta comprensión más rica podría mejorar las curvas de aprendizaje de las personas que se enfrentan por primera vez a un nuevo lenguaje y sin duda podría también aumentar la capacidad de un programador experimentado para crear programas más elaborados y eficientes (p. 436-437).

En la figura 5 pueden observarse las dimensiones de la *escalera semiótica* que según May y Dhillon (2009) se deberían considerar al analizar cualquier elemento de un lenguaje de programación.



Figura 5. Escalera semiótica para analizar lenguajes de programación. Adaptado de “Interpreting beyond syntactics: A semiotic learning model for computer programming languages” por J. May y G. Dhillon, 2009, *Journal of Information Systems Education*, 20(4), p. 434. Copyright 2009 por la Information Systems & Computing Academic Professionals, Inc.

### 1.4.2. Taxonomía de objetivos cognitivos

En la década de 1950, Benjamin S. Bloom —de la *University of Chicago*— editó el primer tomo (denominado *El dominio cognitivo*) de un manual dedicado a presentar una novedosa taxonomía de objetivos educativos desarrollada por un comité de examinadores de universidades estadounidenses. Según Bloom, Engelhart, Furst, Hill y Krathwohl (1956), el propósito de la taxonomía es ayudar a los profesionales que se ocupan de los problemas curriculares y de evaluación, ya que los objetivos educativos clasificados por la taxonomía corresponden a las conductas esperadas de los estudiantes (p. 12).

Casi medio siglo más tarde, Lorin W. Anderson editó, junto con David R. Krathwohl —uno de los autores de la taxonomía anterior— una nueva versión revisada (L. W. Anderson et al., 2001). De acuerdo con Krathwohl (2002),

al igual que la Taxonomía original, la revisión es una jerarquía en el sentido de que las seis categorías principales de la dimensión *Proceso Cognitivo* difieren en su complejidad, siendo *recordar* menos compleja que *comprender*, que es menos compleja que *aplicar*, y así sucesivamente (p. 215).

Las diferencias entre la taxonomía original de 1956 y la taxonomía revisada de 2001 se pueden observar en la tabla 2.

**Tabla 2**

*Estructura del Dominio Cognitivo (taxonomía de 1956) y de la dimensión Proceso Cognitivo (taxonomía de 2001)*

Taxonomía de 1956		Taxonomía de 2001	
Categoría	Subcategoría	Categoría	Proceso Cognitivo
1. Conocimiento	1.1. Conocimientos específicos	1. Recordar - Traer de la memoria de largo plazo conocimientos relevantes	1.1. Reconocer
	1.1.1. Conocimiento de la terminología		1.2. Recuperar
	1.1.2. Conocimiento de hechos específicos		
	1.2. Conocimiento de los modos y medios para trabajar con los anteriores		
	1.2.1. Conocimiento de convenciones		
	1.2.2. Conocimiento de tendencias y secuencias		
	1.2.3. Conocimiento de clasificaciones y categorías		
	1.2.4. Conocimiento de criterios		
	1.2.5. Conocimiento de la metodología		
	1.3. Conocimiento de los conceptos universales y abstracciones en un campo dado o materia		
	1.3.1. Conocimiento de principios y generalizaciones		
	1.3.2. Conocimiento de teorías y estructuras		
	2. Comprensión		2.1. Transferencia
2.2. Interpretación		2.2. Ejemplificar	
2.3. Extrapolación		2.3. Clasificar	
		2.4. Resumir	
			2.5. Inferir
			2.6. Comparar
			2.7. Explicar
3. Aplicación		3. Aplicar - Llevar a cabo o usar un procedimiento en una situación dada	3.1. Ejecutar
			3.2. Implementar
4. Análisis	4.1. Análisis de elementos	4. Analizar - Descomponer el material en sus partes constitutivas y detectar cómo estas se relacionan entre sí y con una estructura global	4.1. Diferenciar
	4.2. Análisis de las relaciones		4.2. Organizar
	4.3. Análisis de los principios de organización		4.3. Atribuir
5. Síntesis	5.1. Producción de una comunicación personal y única	5. Evaluar - Juzgar basándose en criterios y estándares	5.1. Comprobar
	5.2. Elaboración de un plan de acción		5.2. Criticar
	5.3. Derivación de un conjunto de relaciones abstractas		
6. Evaluación	6.1. Evaluación a partir de evidencia interna	6. Crear - Juntar elementos para formar un todo nuevo y coherente o hacer un producto original	6.1. Generar
	6.2. Evaluación a partir de criterios externos		6.2. Planear
			6.3. Producir

*Nota:* Basado en “A revision of Bloom’s taxonomy: An overview” por D. R. Krathwohl, 2002, *Theory into practice*, 41(4), pp. 212–218. Copyright 2002 por el College of Education, The Ohio State University

Al explorar la bibliografía en busca de modelos didácticos para la enseñanza de la programación, Kaasbøll (1998) encontró una estrategia similar a la taxonomía anterior:

Otros . . . han utilizado una estrategia de enseñanza que se asemeja a la taxonomía de objetivos cognitivos de Bloom. La secuencia de instrucción comprendía *usar un programa de aplicación, leer el programa y cambiar el programa*. También se podría agregar *crear un programa* (p. 196).

### 1.4.3. Resolución de problemas

La resolución de problemas es quizás el modelo más comúnmente empleado para aprender programación. Sin embargo, según Kaasbøll (1998) no corresponde considerarlo estrictamente un modelo didáctico:

El proceso de resolución de problemas . . . enfatiza el resultado del proceso de aprendizaje en términos de conocimiento y experiencia personal. Por ello, es más un modelo de aprendizaje que una estrategia de enseñanza. Una estrategia de enseñanza para la resolución de problemas podría ser el paso *cambiar el programa* en el modelo de la taxonomía de objetivos cognitivos, o todo el curso podría organizarse como un conjunto de problemas para resolver (p. 196).

## 1.5. Evaluación del aprendizaje de la programación

Todo aprendizaje debe ser evaluado siempre mediante instrumentos que sean adecuados para establecer en qué medida se han cumplido —o no— los objetivos educativos predeterminados. En el caso de la programación, McCracken et al. (2001) señalan que existen dos categorías de evaluación: la *objetiva* y la *basada en el desempeño* (p. 127). El instrumento utilizado para la evaluación objetiva son las pruebas objetivas, y para la evaluación basada en el desempeño son frecuentes los trabajos para hacer en casa, los exámenes de respuestas cortas y los miniproyectos de laboratorio.

### 1.5.1. Pruebas objetivas

Las pruebas objetivas (es decir, aquellas en las que las respuestas se puntúan de manera objetiva) son uno de los dos tipos principales de pruebas escritas que existen (el otro tipo lo constituyen las pruebas de ensayo, basadas en preguntas de respuesta libre). Las pruebas objetivas pueden incluir desde preguntas de bajo nivel —de *verdadero o falso*—, pasando por preguntas de opción múltiple, hasta preguntas de alto nivel —que exigen descubrir relaciones y evaluar supuestos a partir de la interpretación—

(McCormick y James, 1995, p. 234). Estos autores destacan la utilidad de las pruebas objetivas para evaluar a grupos numerosos, aunque también resaltan su bajo potencial para detectar las concepciones erróneas de los estudiantes:

La facilidad de calificación las hace muy útiles para valorar a grupos grandes de estudiantes . . . Sin embargo, su capacidad de previsión de errores es limitada, mientras que las pruebas de ensayo . . . pueden poner de manifiesto las concepciones erróneas . . . que tienen los estudiantes (p. 234).

Sobre su utilización como instrumentos para medir el aprendizaje de la programación, McCracken et al. (2001) señalan que las pruebas objetivas pueden servir para evaluar algunos conocimientos específicos, pero no si los estudiantes adquirieron la capacidad de programar:

Las formas objetivas de evaluación, como las preguntas de opción múltiple, pueden proporcionar un medio de bajo costo para determinar el conocimiento del estudiante sobre áreas tales como la sintaxis del lenguaje o el comportamiento de los programas. Las pruebas objetivas pueden proporcionar retroalimentación instantánea y se pueden utilizar para la evaluación formativa y sumativa. Por otro lado, las preguntas de opción múltiple no pueden evaluar directamente la capacidad de los estudiantes para crear programas de computadora que funcionen (p. 127).

En efecto, para medir la capacidad de crear programas se requiere implementar la evaluación basada en el desempeño, algunos de cuyos instrumentos se describen a continuación.

### 1.5.2. Trabajos para hacer en casa

En su obra *Evaluating Educational Programmes: The Need and the Response*, Robert E. Stake (1976) hizo la siguiente analogía culinaria: Cuando el cocinero prueba la sopa está haciendo una evaluación *formativa*, cuando la prueba el invitado, está haciendo una evaluación *sumativa* (p. 21). Doménech Betoret (2011) le adjudica la autoría del término *evaluación formativa* a Scriven y lo explica así:

La *evaluación formativa* es un término acuñado por Scriven en 1967 como oposición a la *evaluación sumativa* . . . El objetivo de la evaluación formativa no es sancionar o certificar los resultados alcanzados, sino ayudar tanto al estudiante como al profesor a orientar sus esfuerzos a fin de conseguir el dominio pleno de los objetivos educativos (p. 198).

De acuerdo con este último autor, los trabajos para hacer en casa pueden y deben tener la finalidad de una evaluación formativa:

Los exámenes convencionales parciales o finales, lo mismo que los trabajos o tareas para hacer en casa, prácticas, etc., que son calificados y tienen su peso determinado para calcular la nota final, pueden y deben contribuir a corregir errores y a mejorar los hábitos de estudio (p. 198).

Acerca de su uso para evaluar el aprendizaje de la programación, McCracken et al. (2001) afirman que los trabajos para hacer en casa son adecuados porque el entorno en que se los lleva a cabo es cercano a la realidad —al contrario de una prueba objetiva, por ejemplo—, aunque reconocen que insumen mucho tiempo de ejecución:

Por lo general, durante un curso se asignan varios de estos trabajos, que tienden a ser de escala relativamente grande, con un tiempo máximo generoso (de varias semanas) para completarlos . . . Suelen tener una gran complejidad cognitiva. Son justos, generalizables y significativos en el sentido de que los estudiantes están operando en un entorno cercano a la realidad; sin embargo, los estudiantes son penalizados si no pueden dedicarse a completar el trabajo durante suficiente tiempo. Este tipo de evaluación es bastante vulnerable al plagio (p. 127).

### **1.5.3. Exámenes de respuestas cortas**

En este tipo de evaluación —al igual que en las prueba de ensayo— las respuestas no están disponibles para ser seleccionadas. No obstante, en este caso las respuestas son mucho menos libres y más acotadas. Sobre su uso como instrumentos para medir el aprendizaje de la programación, McCracken et al. (2001) señalan que se los puede configurar para solicitar a los estudiantes la generación de fragmentos de código cortos:

Estos exámenes consisten típicamente en pedirles a los estudiantes que generen fragmentos de código . . . Es difícil (pero no imposible) hacer que los exámenes de respuestas cortas sean significativos o generalizables debido al tiempo limitado disponible para que los estudiantes los completen, pero pueden proporcionar complejidad cognitiva a bajo costo (p. 127).

### **1.5.4. Miniproyectos de laboratorio**

Usados con bastante frecuencia para evaluar el aprendizaje de la programación, estos miniproyectos son trabajos cortos, generalmente llevados a cabo durante una sesión de laboratorio de duración fija. Según McCracken et al. (2001), con ellos se disminuyen los casos de plagio, pero la presión temporal puede perjudicar a los estudiantes ansiosos:

La naturaleza cerrada de estas sesiones reduce las oportunidades de plagio. Los miniproyectos permiten evaluar un gran número de objetivos, aunque de una manera más superficial y menos cognitivamente compleja que con trabajos más grandes para llevar a casa. La experiencia de completar un miniproyecto puede no ser tan significativa o generalizable como los trabajos mayores. Los miniproyectos pueden ser injustos con aquellos estudiantes a quienes las pruebas les generan ansiedad o que tienen problemas con la presión del tiempo (p. 127).

## 1.6. Requisitos cognitivos para el aprendizaje de la programación

Pea y Kurland (1984, p. 155-157) asumen que al menos seis factores cognitivos son requisitos para poder aprender a programar: la capacidad matemática, la capacidad de procesamiento, la capacidad de aplicar el razonamiento analógico, la capacidad de aplicar el razonamiento condicional, la capacidad de aplicar el pensamiento procedimental y la capacidad de aplicar el razonamiento temporal. Estos factores se describen a continuación.

### 1.6.1. Capacidad matemática

El primer factor cognitivo que es un requisito para el aprendizaje de la programación es la *capacidad matemática*, para la cual Karsenty (2014) da dos definiciones:

La *capacidad matemática* es un constructo humano que puede definirse de forma cognitiva o pragmática, según el propósito de la definición. Las definiciones cognitivas se usan para referirse a este constructo desde una perspectiva teórica; la capacidad matemática se puede definir como . . . la capacidad de aprender y dominar nuevas ideas y habilidades matemáticas . . . Las definiciones pragmáticas se usan generalmente cuando se observa este constructo desde una perspectiva de evaluación (por ejemplo, si la meta es evaluar el aprendizaje o identificar el potencial de los estudiantes). Desde esta perspectiva, se puede definir como la capacidad de realizar tareas matemáticas y resolver efectivamente problemas matemáticos (p. 372).

De acuerdo con Pea y Kurland (1984), aunque muchos usos de la computadora hoy en día no son matemáticos (por ejemplo, la administración de bases de datos, el procesamiento de textos), la asociación entre la capacidad matemática y la habilidad de programación se puede buscar en el propósito original de las computadoras:

Las computadoras se desarrollaron primero para ayudar a resolver problemas matemáticos difíciles . . . Si bien dudamos de que las capacidades matemáticas y de programación se relacionen si se excluye la *inteligencia general*, la capacidad matemática no se puede descartar como un requisito para el dominio de ciertas habilidades de programación (pp. 155-156).



### 1.6.2. Capacidad de procesamiento

La *capacidad de procesamiento* es el segundo factor cognitivo necesario para el aprendizaje de la programación, la cual es una disciplina que a menudo requiere mucha memoria y concentración. Según Pea y Kurland (1984), las diferencias individuales en la capacidad de procesamiento son, por lo tanto, un posible predictor sobre quién se puede convertir en un buen programador:

Ciertas tareas permiten . . . evaluar cuánta información una persona puede coordinar en un momento dado y parecen indicar procesos que son básicos para el aprendizaje. Los desempeños en tales tareas se han correlacionado de forma fiable con la inteligencia general, el nivel de desarrollo piagetiano y la capacidad de aprender y usar estrategias de resolución de problemas (p. 156).

### 1.6.3. Capacidad de aplicar el razonamiento analógico

El tercer factor cognitivo que es un requisito para el aprendizaje de la programación es la *capacidad de aplicar el razonamiento analógico*, un tipo de pensamiento para el cual Bartha (2016) da la siguiente definición:

El *razonamiento analógico* es cualquier tipo de pensamiento que se basa en una analogía, es decir, en una comparación entre dos objetos, o sistemas de objetos, que resalta los aspectos en los que se cree que son similares. Un *argumento analógico* es una representación explícita de una forma de razonamiento analógico que cita similitudes aceptadas entre dos sistemas para respaldar la conclusión de que existe alguna similitud adicional.

Pea y Kurland (1984) afirman que la *capacidad de aplicar el razonamiento analógico* es un factor cognitivo necesario para el aprendizaje de la programación, ya que un estudiante que tiene conocimientos y habilidades relevantes para programar podría, sin embargo, no conectarlos con el dominio de la programación, ni transferir el conocimiento adquirido al programar hacia otros dominios, si le faltara esta capacidad de aplicar el razonamiento analógico:

Las transferencias de conocimiento y las estrategias, tanto *hacia* como *desde* el aprendizaje de la programación, pueden depender de la capacidad de aplicar el razonamiento analógico . . . Por ejemplo, al aprender a programar, muchos estudiantes comparan el flujo de control propio de los dispositivos computacionales con el de los modelos físico-mecánicos que ya poseen (p. 156).

#### 1.6.4. Capacidad de aplicar el razonamiento condicional

El cuarto factor cognitivo que se requiere para poder aprender a programar es la *capacidad de aplicar el razonamiento condicional*, un tipo de pensamiento definido por Nickerson (2015) de la siguiente manera:

El *razonamiento condicional* es el razonamiento sobre eventos o circunstancias que dependen de otros eventos o circunstancias. Es un tipo de razonamiento que todos utilizamos constantemente. Sin la capacidad de hacer uso del mismo, los seres humanos serían criaturas muy diferentes, y muy empobrecidas cognitivamente. La imaginación sería imposible . . . No se podrían tomar decisiones que tengan en cuenta posibles contingencias . . . Gran parte de la educación . . . implica el aprendizaje de relaciones condicionales. Aprendemos cómo funciona el mundo y cómo encontrar nuestro camino en él de forma segura, al aprender cómo las condiciones específicas conducen a consecuencias específicas (p. 1).

Dado que trabajar con sentencias condicionales es una parte importante de la programación, pues estas guían la operación de bucles, la validación de las entradas y otras funciones, Pea y Kurland (1984) sostienen que la *capacidad de aplicar el razonamiento condicional* es fundamental para aprender a programar:

Es razonable predecir que un estudiante que tiene una comprensión suficiente de la lógica condicional, de las diversas estructuras de control *si...entonces* y de los conectores lógicos de negación, conjunción y disyunción será un programador más exitoso que un estudiante que tiene problemas para monitorear el flujo de control mediante sentencias condicionales (p. 156).

#### 1.6.5. Capacidad de aplicar el pensamiento procedimental

La *capacidad de aplicar el pensamiento procedimental* es el quinto factor cognitivo requerido para el aprendizaje de la programación. En su libro *Mindstorms: children, computers, and powerful ideas*, Seymour Papert da ejemplos cotidianos de este tipo de pensamiento y afirma que es una poderosa herramienta intelectual:

Todo el mundo trabaja a diario con procedimientos. Jugar a un juego o darle instrucciones a un conductor perdido son ejercicios de pensamiento procedimental. En la vida cotidiana vivimos y usamos los procedimientos, pero no necesariamente reflexionamos sobre ellos . . . El pensamiento procedimental es una poderosa herramienta intelectual, y hacer una analogía de uno mismo con una computadora puede ser una buena estrategia para desarrollarlo (Papert, 1980, pp. 154-155).

Pea y Kurland (1984) reconocen el carácter *procedimental* de las instrucciones de los programas de computadora, y consideran que los ejemplos de la vida cotidiana tienen características diferentes, por lo que califican a estos de *cuasiprocedimentales*:

Dar y seguir instrucciones de tareas, órdenes y recetas es *cuasiprocedimental* en lugar de *procedimental* pues, a diferencia de las instrucciones procedimentales en un programa de computadora, a menudo hay *ambigüedad* en los ejemplos cotidianos, de modo que las instrucciones, las órdenes y las recetas no siempre tienen un significado inequívoco, ni están limitados por una *secuencialidad* estricta. Uno puede, a menudo, optar por omitir los pasos en una receta o en un conjunto de instrucciones, o reordenar los pasos. Ninguna de estas opciones está disponible en el estricto carácter procedimental de las instrucciones de un programa de computadora. Sin embargo, las similitudes entre los casos cotidianos y las instrucciones de los programas son lo suficientemente convincentes como para justificar la designación de los primeros como *cuasiprocedimentales* (p. 156).

De todas formas, estos autores admiten que la *capacidad de aplicar el pensamiento cuasiprocedimental* —adquirida a partir de experiencias de la vida cotidiana— suele tener un impacto positivo en el aprendizaje de la programación:

Es de suponer que los estudiantes más familiarizados con estos procedimientos lineales, análogos al *flujo de control* de las operaciones expresadas como instrucciones en un programa de computadora, se familiarizarán más fácilmente con el *pensamiento procedimental* promocionado como una faceta central de la experiencia en programación (p. 157).

### 1.6.6. Capacidad de aplicar el razonamiento temporal

Ejecutar una subrutina o procedimiento *antes* que otro, asegurarse de que un contador no exceda cierto valor *hasta* que otra operación haya finalizado, imaginar los posibles resultados generados como consecuencia de las opciones de diseño de un programa: todas estas operaciones fundamentales requieren una comprensión temporal. Es por ello que Pea y Kurland (1984) señalan la *capacidad de aplicar el razonamiento temporal* como el sexto y último factor cognitivo que se requiere para aprender a programar:

Crear y entender programas requiere una comprensión de la lógica temporal de las instrucciones secuenciales: es el corazón intelectual de aprender a programar ... En la enseñanza de la programación, el concepto teórico central que guía todo el esfuerzo es que las formas clásicas del pensamiento espacial-geométrico-pictórico deben ser aumentadas, y ocasionalmente reemplazadas, por la lógica temporal-imaginativa-memorial (p. 157).

## 1.7. Obstáculos para el aprendizaje de la programación

En la actualidad, los sistemas educativos de todo el mundo están siendo sometidos a profundos cambios. Jacobs, Renandya y Power (2016) consideran que el más profundo de estos cambios es la transición hacia un modelo de aprendizaje centrado en el estudiante:

A menudo . . . escuchamos acerca de muchos cambios en la educación, desde un mayor uso de la tecnología a una mayor flexibilidad en la organización de las escuelas. Pero quizás el cambio más profundo lo constituyen los intentos de avanzar hacia un modelo de aprendizaje centrado en el estudiante . . . *Centrado en el estudiante* simplemente significa que los estudiantes se vuelven más activos, no solo realizando actividades de aprendizaje —desde hablar, calcular, experimentar y escribir, hasta crear videos y materiales basados en la web—, sino también pensando y moldeando su propio aprendizaje (p. xiii).

Según Robins, Rountree y Rountree (2003), el aprendizaje de la programación no escapa a esta última tendencia:

La mayoría de los principiantes aprenden a programar a través de la instrucción formal, como, por ejemplo, en un curso universitario introductorio. La enseñanza y el aprendizaje de principiantes en este contexto son el tema de una extensa literatura educativa. La teoría actual sugiere que hay que poner el foco no en la *enseñanza llevada a cabo por el instructor*, sino en el *aprendizaje del estudiante* y en la comunicación efectiva entre el docente y el estudiante. El objetivo es fomentar el aprendizaje profundo de los principios y habilidades, y crear aprendices reflexivos, independientes y dispuestos a continuar aprendiendo de por vida (p. 156).

Por ello, es fundamental reconocer algunos obstáculos que los estudiantes pueden tener que enfrentar al aprender a programar: su motivación inadecuada, su estilo de aprendizaje inapropiado, la creencia en su falta de aptitud, el ritmo del curso y la dificultad de la disciplina.

### 1.7.1. Motivación inadecuada del estudiante

En general, se dice que una actividad está intrínsecamente motivada si no hay una recompensa externa obvia asociada con ella. Por el contrario, se dice que una actividad tiene una motivación extrínseca si participar en la actividad conduce a algunas recompensas externas como comida, dinero o refuerzo social. Si bien Malone (1980) considera que a menudo es difícil distinguir si una motivación es *intrínseca* o *extrínseca*, defiende la existencia de esta dicotomía debido a su atractivo conceptual y a su utilidad:

Otra forma de definir el tipo de motivación es permitir que la persona haga la distinción . . . si percibe que sus acciones y consecuencias están en gran medida bajo su propio control o si están determinadas principalmente por fuerzas externas. A pesar de estas ambigüedades, la distinción entre motivación intrínseca y extrínseca es conceptualmente atractiva y heurísticamente útil (p. 3).

El estudiante que se acerca a una carrera de computación siempre tiene alguna motivación, ya sea *intrínseca* (porque tiene un interés genuino en la disciplina), *extrínseca* (porque considera su futuro diploma apenas como un medio hacia una carrera lucrativa) o *social* (porque quiere complacer a sus padres o a su familia). Según T. Jenkins (2002), el tipo de motivación puede ser un factor determinante del desempeño del estudiante que aprende programación:

Se ha demostrado —y esto tal vez no sea sorprendente— que los estudiantes con peor desempeño en programación tienen más probabilidades de tener una motivación principalmente extrínseca que sus colegas que sobresalen en esta disciplina. Por otra parte, también se ha demostrado que los estudiantes de programación usualmente mantienen algún tipo de motivación durante los cursos, incluso si esta proviene de un factor negativo como es el miedo al fracaso. De una forma o de otra, todos ellos están motivados para aprender. Ninguno falla a propósito (p. 54).

Por lo tanto, aunque podría pensarse que la falta de motivación constituye uno de los obstáculos para el estudiante que quiere aprender programación, el verdadero problema es, en realidad, la inadecuación de su motivación.

### 1.7.2. Estilo de aprendizaje inapropiado

Adoptar un estilo de aprendizaje inadecuado puede ser un enorme obstáculo para cualquier estudiante. Es por ello que, según T. Jenkins (2002), el docente debe asegurarse de que los estudiantes no adopten los enfoques que prefieran, sino los más apropiados:

Los estudiantes prefieren aprender de diferentes maneras. Algunos pueden considerar el aprendizaje como un proceso solitario y aprenden mejor de esa manera. Otros pueden preferir un entorno de aprendizaje más dinámico y aprenden mejor mediante discusiones con sus compañeros. Algunas asignaturas pueden exigir un enfoque de aprendizaje particular pero, sin orientación, los estudiantes tenderán a adoptar el estilo que prefieran o que les haya servido mejor en el pasado. Es una responsabilidad crucial del docente asegurarse de que los estudiantes adopten el enfoque más apropiado para la materia en cuestión (p. 54).

Este autor señala que la clasificación más conocida de estilos de aprendizaje divide el aprendizaje en enfoques *profundos* y enfoques *superficiales*, y explica que los aprendices que adoptan enfoques profundos se concentran en obtener una comprensión de un tema, mientras que los que adoptan enfoques superficiales no se concentran más que en memorizar. Lo complejo de la programación es que, en principio, cualquiera de estos enfoques parecería ser perfectamente aplicable para su aprendizaje:

A primera vista, puede parecer que lo vital para la programación es el aprendizaje profundo, que proporciona una comprensión aplicable a nuevas áreas problemáticas. Sin embargo, también podría argumentarse que la programación puede aprenderse esencialmente como un proceso que equivale a una simple búsqueda de *coincidencia de patrones*, donde se detectan problemas comunes y se aplican soluciones de trabajo conocidas. Este enfoque suena mucho más como una forma de aprendizaje superficial (p. 54).

Sin embargo, la adopción de estos estilos por separado es insuficiente para aprender a programar, ya que esta competencia solo puede desarrollarse exitosamente aplicando ambos enfoques simultáneamente, lo que es algo nuevo para la mayoría de los estudiantes:

El aprendizaje superficial puede ser útil para recordar los detalles de la sintaxis o cuestiones como la precedencia operadores, pero se requieren elementos de aprendizaje profundo (y, por lo tanto, comprensión) para desarrollar una verdadera competencia. Esta es la clave: los dos estilos de aprendizaje deben aplicarse al mismo tiempo . . . Esto pone la programación más allá de la experiencia educativa de la mayor parte de los estudiantes; requiere una combinación de estilos de aprendizaje que ellos, en su mayoría, no han tenido que aplicar antes (p. 54).

### 1.7.3. Creencia del estudiante en su falta de aptitud

En la década de 1950 se hizo evidente la escasez de mano de obra capacitada para programar los cada vez más numerosos sistemas de cómputo disponibles en el mundo. La industria reaccionó desarrollando instrumentos para facilitar el reclutamiento de personal:

Para identificar y reclutar posibles programadores, la SDC (*System Development Corporation*) empleó una estrategia que se convertiría en una práctica estándar en la industria . . . SDC desarrolló un conjunto de pruebas de aptitud y perfiles psicológicos . . . El objetivo de las pruebas de aptitud era filtrar según los rasgos que se consideraban esenciales en un buen programador, como la capacidad de pensar de forma lógica y de llevar a cabo un razonamiento abstracto. El propósito de los perfiles psicológicos era identificar individuos con la personalidad apropiada para la tarea de programar (Ensmenger, 2010, pp. 61-62).

Según Ensmenger (2010), ninguna prueba de aptitud para la programación fue tan popular como la IBM PAT, que fue ampliamente utilizada en todo el mundo durante varias décadas:

En 1955, IBM contrató a dos psicólogos, Walter McNamara y John Hughes, para que desarrollaran una prueba de aptitud que permitiera identificar el talento para la programación. Denominada IBM PAT (*Programmer Aptitude Test*) a partir de 1959, en el transcurso de las siguientes décadas esta prueba se convertiría en el instrumento estándar de la industria para evaluar la capacidad de programación (p. 64).

De acuerdo con este autor, la IBM PAT era una prueba que requería velocidad y precisión, ya que contenía muchas preguntas que debían ser respondidas en un tiempo relativamente corto, y su estructura no sufrió grandes cambios durante todos los años en que estuvo vigente:

Aunque posteriormente se publicaron algunas variaciones menores de la PAT, su estructura general permaneció sorprendentemente constante. La primera sección requería que los candidatos identificaran la regla subyacente que definía el patrón de una serie de números. La segunda sección era similar a la primera, pero involucraba formas geométricas en lugar de números. La tercera y última sección planteaba problemas que podían reducirse a formas algebraicas, como *¿Cuántas manzanas puedes comprar por 60 centavos si tres cuestan 10 centavos?* Los candidatos disponían de 50 minutos para responder aproximadamente 100 preguntas, por lo que se requerían velocidad y precisión (p. 65).

En aquella época, la utilización de pruebas de aptitud se basaba en la suposición ampliamente difundida de que la capacidad de programación era una capacidad innata más que una capacidad aprendida, algo que debía identificarse en lugar de ser inculcado (Ensmenger, 2010, pp. 67-68).

Sin embargo, según T. Jenkins (2002), hoy en día, la aceptación de tal suposición ya no es tan amplia:

Algunos estudios insinúan que hay cierta relación entre las habilidades de programación y la experiencia en matemáticas, mientras que otros concluyen que no hay ninguna conexión . . . Sin duda, tener algo de experiencia en programación antes de empezar un curso de programación ayuda, pero esto no es lo mismo que tener *aptitud*. Hay disponibles varias pruebas de aptitud para la programación, pero la evidencia de su efectividad no es concluyente . . . Es posible que la aptitud para la programación ni siquiera exista (p. 54).

De acuerdo con Scott y Ghinea (2013), el verdadero obstáculo que deben enfrentar los estudiantes de programación no es entonces su falta de aptitud, sino más bien la *creencia* en su falta de aptitud, la cual los lleva a evitar la ejercitación frecuente que es precisamente la manera correcta de desarrollar esta competencia:

Con frecuencia, los estudiantes empiezan a creer que se requiere una aptitud inherente para convertirse en un programador. Tal creencia inhibe la práctica frecuente y deliberada de la escritura de código por parte de los estudiantes. Por lo tanto, es importante que las pedagogías de la programación refuercen la noción de que . . . la aptitud es una cualidad maleable que se incrementa mediante el esfuerzo (p. 3).

#### 1.7.4. Ritmo del curso

En las universidades, la programación se enseña —y por lo tanto se aprende— a un ritmo establecido —durante uno o dos semestres, o incluso varios años— y este ritmo de dictado del curso de programación puede ser un enorme obstáculo para los estudiantes que deben seguirlo. T. Jenkins (2002) critica los cursos que se dictan a un ritmo definido por las necesidades de evaluación y que no consideran que la programación es una habilidad y, como tal, requiere que se respete del ritmo de aprendizaje propio de cada estudiante:

Dado que el ritmo de la instrucción no está bajo el control de los estudiantes (y es más que probable que diferentes estudiantes aprendan a diferentes ritmos), habrá quienes pierdan un concepto básico y no puedan seguir el contenido de las clases posteriores. No habrá vuelta atrás: el curso será para ellos como un tren de alta velocidad sin frenos. Tales estudiantes rápidamente llegarán a la conclusión de que ellos simplemente *no pueden cursar programación* . . . y lo atribuirán a la dificultad percibida hacia la disciplina. Por supuesto, el ritmo del curso a menudo depende principalmente de las necesidades de evaluación. Este esquema puede ser razonable para muchas materias, pero es bastante ridículo para aprender una habilidad como es la programación (p. 57).

#### 1.7.5. Dificultad de la disciplina

La programación no es una disciplina fácil. T. Jenkins (2002) está en contra de que su enseñanza en las universidades ocurra precisamente cuando los estudiantes se encuentran atravesando una etapa complicada de sus vidas —el fin de la adolescencia y la transición hacia la vida adulta—, lo que hace que su aprendizaje se vuelva más difícil:



La programación normalmente se enseña como una asignatura fundamental al inicio de la carrera. Este es un momento difícil para muchos estudiantes: un momento de transición en el que se adaptan a la vida universitaria y al estudio. Es posible que vivan lejos de su hogar por primera vez, que tengan dificultades para hacer nuevos amigos y encontrar un lugar en su nuevo entorno, y que les resulte difícil manejar sus propias finanzas y su propio tiempo de estudio y privado. En el medio de todo esto, en su curso de programación encontrarán . . . un material potencialmente desafiante que formará la base del resto de su aprendizaje. Se perderán si no lo entienden. Aunque este material ya es bastante difícil de dominar cuando un estudiante está bien establecido, los departamentos de computación insisten en enseñarlo durante un periodo que es de transición, lo cual aumenta su dificultad (p. 55).

Gran parte del *shock* de los primeros encuentros entre los estudiantes y los entornos de programación puede atribuirse al intento de abordar al mismo tiempo todos los tipos de dificultades diferentes que estos sistemas plantean. Du Boulay (1986) clasifica los problemas que contribuyen a la dificultad de la programación, separándolos en cinco áreas con cierto grado de superposición, pues ninguno de los cinco problemas es completamente separable de los demás:

Primero está el problema general de la *orientación*: descubrir para qué sirve la programación, qué tipos de problemas se pueden abordar . . . En segundo lugar, existen dificultades asociadas con la comprensión de las propiedades generales de la máquina que se está aprendiendo a controlar —la *máquina nocional*— y cómo el comportamiento de la máquina física se relaciona con esta . . . En tercer lugar, hay problemas asociados con la *notación* de los diversos lenguajes formales que deben aprenderse . . . En cuarto lugar, asociadas con la notación están las dificultades de adquirir *estructuras* estándar o patrones que se pueden utilizar para cumplir objetivos en pequeña escala . . . Finalmente, está el problema de dominar la *pragmática* de la programación, donde el estudiante necesita aprender cómo especificar, desarrollar, probar y depurar un programa utilizando las herramientas disponibles. (pp. 57-58).

Algunos de los procesos que forman parte de la programación son más difíciles que otros. Según T. Jenkins (2002), una vez establecido el algoritmo, los demás procesos son más fáciles:

La parte más difícil es traducir la especificación al algoritmo. Esto también es lo más importante, ya que es crucial que se utilice un algoritmo correcto y eficiente como base de cualquier codificación. Dado un algoritmo correcto, los otros procesos son esencialmente mecánicos (p. 55).

De acuerdo con este autor, además de la dificultad propia de la disciplina, los estudiantes también deben enfrentar la *reputación* negativa que tienen, a menudo, los cursos de programación y los programadores:

Los cursos de programación tienen la reputación de ser difíciles. Los nuevos estudiantes reciben esta visión a través de sus predecesores, quienes a menudo exageran en la narración . . . Al mismo tiempo, existe la imagen pública del programador como un *nerd* socialmente inadaptado, alimentado a base de pizza y café, que pasa sus horas produciendo un código ininteligible . . . Si los estudiantes se acercan a un curso con la expectativa de que será difícil y con una imagen negativa de aquellos que sobresalen en la disciplina, es difícil imaginar que estén muy motivados. Y si los estudiantes no están motivados, no tendrán éxito (p. 56).

## LAS RELACIONES INTERPERSONALES EN LAS AULAS

Si bien Jacob Levy Moreno ya había utilizado el término *sociometría* en 1916 (Arruga i Valeri, 1974, p. 15), puede considerarse que la piedra fundamental del movimiento sociométrico fue puesta recién en 1934, con la publicación de su libro *Who shall survive? A new approach to the problem of human relations* (p. 16).

Definida como “la ciencia de las relaciones interpersonales” (Portuondo, 1971, p. 13), la sociometría proporciona técnicas para evaluar y medir tales relaciones dentro de los más diversos grupos humanos.

En su libro *The development of social network analysis: a study in the sociology of science* (Freeman, 2004) dedicado a la historia y el desarrollo del ARS (*Análisis de Redes Sociales*) —un enfoque cuya importancia dentro de las ciencias sociales ha crecido de forma continua en los años recientes, algo que se evidencia en el surgimiento de instituciones académicas como la *International Network for Social Network Analysis* («INSNA», s. f.); publicaciones especializadas como la *Revista Hispana para el Análisis de Redes Sociales* («REDES», s. f.) y programas de computación específicos como Pajek («Pajek: analysis and visualization of large networks», s. f.)—, Linton C. Freeman —un sociólogo estructuralista de la *University of California, Irvine* — señala que, sin los aportes colosales de Jacob. L. Moreno, el ARS directamente no existiría (p. v). Además, el tercer capítulo completo —titulado *The Birth of Social Network Analysis I: Sociometry*— Freeman lo dedica a tratar la sociometría como antecedente del ARS.

### 2.1. La importancia de estudiar las relaciones humanas en las aulas

En particular, la importancia de estudiar las relaciones humanas dentro de las aulas fue resaltada en la década de 1950 por un grupo de investigadores del *Horace Mann-Lincoln Institute of School Experimentation, Teachers College, Columbia University*:

Sabemos que, en estos tiempos, no es necesario recordarles a los profesores la importancia del estudio de las relaciones humanas: la forma en que las personas se llevan unas con otras, la forma en que operan cuando están en grupos. El psicólogo, el sociólogo, el antropólogo, el estadista nos señalan la urgencia del problema. La necesidad de una mejor comprensión de las relaciones humanas se manifiesta en problemas que van desde el ajuste personal, la vida en el aula, las relaciones familiares y la eficiencia profesional hasta las tensiones intergrupales y las relaciones internacionales (Cunningham, Elzi, Hall, Farrell y Roberts, 1951, p. vii).

Estudiar las relaciones humanas en las aulas, según Arruga i Valeri (1974), es importante porque ayuda a averiguar muchos factores que impactan sobre el desarrollo de las clases, entre los cuales este autor destaca los siguientes:

Características de los líderes, los aislados, los excluidos, etc. según sus funciones, según las diversas clases sociales y según el ambiente familiar; la influencia de la coeducación, de las relaciones profesor-líderes, del distinto origen regional, de las diferencias lingüísticas, sociales, raciales, etc., en la integración social; la influencia de los factores psicosociales: inteligencia, creatividad, interés, actitudes, estatus, nivel socioeconómico, dimensiones familiares, etc., en el estatus sociométrico; la integración de los niños considerados estudiosos por el profesor; las diferencias estructurales según la edad, clase social, objetivos de los grupos, etc.; las tensiones existentes en grupos impuestos autoritariamente donde hay grupos espontáneos (p. 25).

## 2.2. El test sociométrico

El principal instrumento provisto por la sociometría para evaluar y medir las relaciones interpersonales en un grupo es el *test sociométrico*. Bastin (1966) lo define en base a su utilidad:

Un instrumental que puede suministrarle [tanto al educador como al psicólogo] indicaciones acerca de la vida íntima de los grupos que él educa o examina, así como acerca de la posición social y el papel de cada individuo dentro de dichos grupos (p. 4).

En cambio, Arruga i Valeri (1974) también menciona —aparte de su utilidad— el contenido del test, cuando propone esta definición:

Se llama así al método empleado para conocer la estructura básica interrelacional de un grupo a través de las respuestas de sus componentes sobre sus propias atracciones y rechazos. Es un instrumento que mide la organización de los grupos sociales. Su contenido es de lo más sencillo que pueda existir: pedir a los sujetos de un grupo que expresen los compañeros que prefieren y aquellos que les disgustan (p. 21).

Por último, Fernández Prados (2000) además lo clasifica como una técnica de investigación de orden cuantitativo:

El test sociométrico es una de las técnicas de investigación de orden cuantitativo de la metodología sociométrica que permite determinar el grado en que los individuos son aceptados o rechazados en un grupo (su estatus sociométrico), descubrir las relaciones entre los individuos y revelar la estructura del grupo mismo (p. 54).

La realización del test sociométrico consta de un trabajo preparatorio “que bajo ningún concepto puede omitirse” (Bastin, 1966, p. 4), la aplicación del cuestionario, el análisis de las respuestas y la síntesis de los resultados. A su vez, estas etapas se dividen en los pasos que se detallan a continuación.

### 2.2.1. Trabajo preparatorio I: Elaboración del protocolo o ficha técnica

Hay consenso entre diferentes autores sobre la necesidad de iniciar el trabajo preparatorio mediante la elaboración de un protocolo o ficha técnica. Bastin (1966) sostiene que “es necesario encontrarse familiarizado con el grupo al que se desea interrogar y conocer sus principales actividades” (p. 14). Arruga i Valeri (1974) señala que “conviene . . . al investigador sociométrico poseer un conjunto de datos sobre el grupo antes de la realización del test” (p. 31) y enumera los datos que se muestran en la tabla 3.

**Tabla 3**

*Un conjunto de datos sobre el grupo que se deberían incluir en el protocolo de un test sociométrico*

Dato sobre el grupo	Finalidad
Tamaño o magnitud	Permite delimitar en sus líneas generales la forma del test sociométrico que se realizará, por ejemplo: el número de elecciones que se permitirá como respuesta.
Objetivos	Permiten delimitar los criterios de elección adecuados para explorar las interrelaciones, por ejemplo: si el objetivo del grupo es resolver problemas escolares, las elecciones podrían basarse en criterios de afinidad para el trabajo, que buscan relaciones <i>horizontales</i> , como la fraternización y la reciprocidad.
Tiempo de existencia o formación	Si el grupo es de formación reciente, los sentimientos tenderán a ser superficiales.
Tiempo transcurrido desde la incorporación de los últimos miembros	Permite conocer la posibilidad de interrelaciones débiles, y conociendo los nombres de estos miembros, quiénes posiblemente estén menos integrados al grupo.

*Nota:* Basado en *Introducción al test sociométrico* por A. Arruga i Valeri, 1974, pp. 31-32. Copyright 1974 por Editorial Herder S. A.

Fernández Prados (2000) también coincide en que la elaboración del protocolo es lo primero que debe hacerse:

Antes de la construcción de un cuestionario es preciso una fase de reflexión que nos ayude a tener en cuenta algunas consideraciones del grupo y a clarificar los objetivos que pretendemos con el test sociométrico. Esta es la misión del protocolo o ficha técnica (p. 54).

Sin embargo, comparado con los anteriores, este autor es mucho más formal en cuanto al contenido del protocolo, ya que lista las cuatro partes de que este debe constar, detallando en cada una de ellas un gran número de ítems (tabla 4).

**Tabla 4**

*Partes del protocolo de un test sociométrico*

Parte del protocolo	Ítem
Características externas o circunstanciales del grupo	Nombre del colegio Localización o dirección Curso y grupo Profesor o tutor (sería conveniente listado de profesores)
Características internas o propias del grupo	Listado de alumnos (expresando sexo, edad u otra variable) Antigüedad del grupo Incorporaciones y bajas recientes en el seno del grupo Objetivos del grupo
Características del cuestionario del test	Objetivo o criterio escogido (Afectivo, trabajo o lúdico) Tipos de preguntas (Elección, rechazo y percepción) Posibilidad de elegir (Exogrupo o intragrupo) Número de elecciones (Limitadas o ilimitadas)
Características de la administración del test	Ausentes (¿A quién?) Día y hora (¿Cuándo?) Duración (¿Cuánto?) Administrador/a (¿Por quién?) Observaciones (¿Qué?)

*Nota:* Basado en *Sociología de los grupos escolares: sociometría y dinámica de grupos* por J. S. Fernández Prados, 2000, pp. 54-56. Copyright 2000 por Universidad de Almería. Servicio de Publicaciones

### 2.2.2. Trabajo preparatorio II: Elaboración del cuestionario

La elaboración del cuestionario del test sociométrico puede llevarse a cabo con distintos grados de rigurosidad. Según Bastin (1966), “este cuestionario, como

puede verse fácilmente, no requiere material alguno; alcanza una hoja de papel y un lápiz” (p. 4).

Por su parte, Portuondo (1971) solamente resalta la conveniencia de que las preguntas estén correctamente adaptadas a la edad cronológica y mental del grupo (p. 14).

En cambio, Fernández Prados (2000) vuelve a ser más detallado que los demás autores y recomienda elaborar un cuestionario en el cual no falten el *título* —que debería expresar el objetivo del test—, el *encabezamiento* —donde se recogerán los datos personales del participante—, las *instrucciones* —que explicarán a quiénes y a cuántos participantes se puede elegir y, eventualmente, en qué orden hay que escribirlos—, las *preguntas* —en las cuales se plasman los criterios de elección o de rechazo— y el *agradecimiento* (p. 56).

Además de sugerir la estructura ideal del cuestionario, este autor también da cinco recomendaciones para elaborarlo: mantenerlo breve —de no más de cuatro preguntas—; utilizar un lenguaje sencillo y claro; elegir un tamaño de letra adecuado para la edad de los participantes; solo formular preguntas realistas —huyendo de situaciones hipotéticas— y cuidar la presentación —agregando algunas imágenes para quitarle solemnidad— (p. 58).

El grupo liderado por Ruth Cunningham en el *Horace Mann-Lincoln Institute* diseñó y publicó su *Escala de la Distancia Social en el Aula (Classroom Social Distance Scale)*, la cual no utiliza preguntas. En su lugar, presenta una lista de frases que describen sentimientos hacia otra persona y, al hacer el test, el participante debe poner una marca en el espacio correspondiente a la frase que mejor describa lo que siente por cada uno de los demás participantes. El cuestionario estándar de estos autores se muestra en la figura 6.

### **2.2.3. Trabajo preparatorio III: Motivación del grupo**

Ante la aplicación de cualquier instrumento, las personas suelen sentir ansiedad o incomodidad, llegando algunas al punto de no querer responder o, lo que puede ser peor, responder al azar con tal de evadirse de la situación. Por ello, es fundamental motivar al grupo para que sus miembros respondan espontáneamente y con la verdad. Bastin (1966) sugiere que el investigador sociométrico solicite a los participantes que respondan con toda sinceridad “después de algunas palabras de introducción a propósito de las ocasiones que han tenido de apreciarse mutuamente” (p. 15).

HORACE MANN-LINCOLN INSTITUTE OF SCHOOL EXPERIMENTATION  
TEACHERS COLLEGE, COLUMBIA UNIVERSITY

**ESCALA DE LA DISTANCIA SOCIAL EN EL AULA**

Nombre: ..... Fecha: .....

Escuela: ..... Grado: .....

No nos gustan todos nuestros amigos de la misma manera. Algunos nos gustan más que otros. Puede haber algunas personas que no nos gusten en absoluto.

La lista de verificación que se encuentra más abajo te dará una forma de decir cuán próxima sería la relación que te gustaría tener con otros niños y niñas en tu sala. Debajo de cada nombre listado en la parte superior de la página, pon una marca en el espacio correspondiente a la frase que mejor describe tu sentimiento hacia esa persona. Por supuesto, debes sustituir *él* por *ella* al pensar antes de marcar el nombre de una niña.

*Cuando llegues a tu propio nombre, marca el espacio que mejor describe lo que piensas que la mayoría de los niños y niñas sienten por ti.*

*Nadie en tu sala verá este papel, solo tu maestra/o.*

<i>Nombre</i>	1. Me gustaría tenerlo como uno de mis mejores amigos.	2. Me gustaría tenerlo en mi grupo pero no como un amigo cercano.	3. Me gustaría estar con él de vez en cuando, pero no con frecuencia o por mucho tiempo.	4. No me molesta que esté en nuestra sala, pero no quiero tener nada que ver con él.	5. Desearía que él no estuviera en nuestra sala.

Figura 6. Escala de la Distancia Social en el Aula. Adaptado de *Understanding group behavior of boys and girls* por Cunningham et al., 1951, p. 405. Copyright 1951 por Columbia University



### 2.2.4. Aplicación del cuestionario

Esta es la parte más breve de todo el proceso. Según Bastin (1966), “su duración es de un cuarto de hora a lo sumo. Son escasos los tests o cuestionarios que pueden jactarse de suministrar a los psicólogos tanta información en tan breve tiempo” (p. 4).

Asimismo, el test debería ser presentado por alguien en quien los miembros del grupo confíen, ya que, en caso contrario, de acuerdo con Arruga i Valeri (1974), su funcionamiento se vería perjudicado:

Conviene sea este [el presentador] una persona que goce de la simpatía y confianza del grupo en cuestión. La actuación de una persona o de un grupo de sociómetras extraños al grupo provocan normalmente una actitud sino hostil, al menos de resistencia interna, que bloquean muchos datos que serían de gran interés (p. 55).

Fernández Prados (2000, p. 59) presenta una serie de recomendaciones para la aplicación del cuestionario, las cuales se resumen en la tabla 5.

**Tabla 5**

*Recomendaciones para la aplicación del cuestionario de un test sociométrico*

Recomendación	Descripción
Tener en cuenta las condiciones previas	El test sociométrico requiere que se cumplan tres condiciones: la primera es que exista grupo, la segunda que el grupo tenga un mínimo rodaje y conocimiento mutuo y la tercera es que exista una cierta predisposición a contestar sinceramente.
Garantizar el anonimato	Con relación a la publicidad y comunicación de los resultados caben plantearse tres situaciones: en la primera solo el profesor tendría la información, en el segundo caso el profesor comunicaría al interesado algunos resultados de modo privado y, por último, siempre y cuando exista un acuerdo previo, se podrá hacer público el informe final. Lo normal es que se asegure el anonimato.
Evitar la solemnidad	No se debe actuar como si el test fuera una actividad extraordinaria o muy diferente del resto, su planteamiento debe ser lo más natural posible, lejos de presentaciones excesivamente serias.
Evitar la intercomunicación	Es fundamental que el cuestionario sea respondido por cada uno de los miembros de la clase sin la influencia u opinión del resto de los compañeros. Incluso podría ser conveniente que se fuera contestando uno a uno, en caso de que se previera una posible comunicación entre los alumnos.
Explicar previamente todo	Es recomendable solucionar previamente todas las dudas y preguntas sobre el cuestionario, para evitar resolverlas durante su administración y así mantener un clima de concentración y silencio.
Dar tiempo suficiente.	El test sociométrico no hay que entenderlo en ningún caso como un ejercicio o una prueba en la que la variable tiempo sea importante. Por este motivo hay que dejar el tiempo suficiente para que todos puedan contestar, a su ritmo todas y cada una de las preguntas.
Realizar adaptaciones si fuera necesario	En el caso de niños de nivel infantil tal vez sería más conveniente emplear otros sistemas de administración a través de fotografías, entre las que el niño escogerá las que prefiere.

*Nota:* Basado en *Sociología de los grupos escolares: sociometría y dinámica de grupos* por J. S. Fernández Prados, 2000, p. 59. Copyright 2000 por Universidad de Almería. Servicio de Publicaciones

### 2.2.5. Análisis de las respuestas I: Elaboración de la matriz sociométrica

El primer paso requerido para el análisis de las respuestas del cuestionario es la elaboración de la matriz sociométrica o sociomatrix. Fernández Prados (2000) da la siguiente definición de este instrumento:

Las matrices sociométricas o sociomatrixes son cuadros de doble entrada en los que tanto en la primera columna como en la primera fila se colocan los nombres de los miembros del grupo y en el resto los resultados del test sociométrico de acuerdo a una serie de criterios (p. 64).

De la anterior definición se deduce que las matrices sociométricas tienen la misma cantidad de filas que de columnas. Al contrario de lo que se afirma en la anterior definición, es poco frecuente colocar en la primera fila y en la primera columna los nombres de los miembros del grupo. En efecto, según Arruga i Valeri (1974), “ordinariamente no se coloca todo el nombre, sino de dos a cinco letras que en un máximo de dos sílabas nos recuerden el nombre completo evitando la posibilidad de confundirse con otros elementos del grupo” (p. 71). De acuerdo con Fernández Prados (2000), además es conveniente colocarlos agrupados para lograr estudiar posibles relaciones entre sujetos de distintas características, por ejemplo, primero las niñas y luego los niños (p. 64).

Los resultados del test sociométrico se cargan en la sociomatrix colocando, en la respectiva fila de cada miembro del grupo, las elecciones y los rechazos que este ha señalado, codificados y ponderados —si corresponde— según lo que se desee medir. Por ejemplo, cuando no es necesario tener en cuenta un orden de preferencia, según Arruga i Valeri (1974), los datos pueden codificarse mediante cualquier tipo de signo, como +, -, x, ( ), etc. (p. 72). Usualmente, las elecciones se capturan con signo positivo o color azul, mientras que los rechazos con negativo o color rojo.

En cambio, en la *Escala de la Distancia Social en el Aula* de Cunningham et al. (1951), lo que se desea medir sí tiene una ponderación, por lo cual los puntajes otorgados van desde el mínimo 1 (menor distancia social, es decir, el valor indica proximidad o afinidad) hasta el máximo 5 (mayor distancia social, es decir, el valor indica lejanía o falta de afinidad). En la figura 7 se muestra una matriz sociométrica cargada con los puntajes otorgados por los 15 miembros de un grupo.

	ADR	CARI	FER	GAVI	PAO	VAN	ALE	ANT	BER	CAVI	DIE	FRA	GASI	PAB	SEB
ADR		1	1	2	1	2	3	1	5	3	3	4	3	4	2
CARI	2		1	2	1	1	3	3	2	4	2	3	3	4	2
FER	1	1		3	1	2	2	4	5	3	3	3	3	4	4
GAVI	2	2	4		2	1	3	4	2	4	5	2	3	3	3
PAO	1	1	2	2		1	3	3	3	2	4	4	2	3	3
VAN	1	1	3	1	2		3	5	4	4	3	3	4	4	3
ALE	3	2	3	4	3	2		2	2	2	1	2	1	2	3
ANT	2	4	5	3	2	4	1		1	1	2	2	3	2	2
BER	4	3	4	2	4	5	2	2		2	2	1	2	3	3
CAVI	3	4	2	5	3	4	3	1	3		5	3	3	3	3
DIE	2	2	3	4	5	3	2	2	1	4		2	3	1	2
FRA	4	3	4	2	4	4	2	3	1	2	1		2	2	1
GASI	3	4	3	5	3	5	1	2	3	3	2	2		2	4
PAB	5	3	5	4	2	2	3	1	3	2	1	3	1		2
SEB	3	2	4	2	3	4	2	3	2	3	2	1	3	3	

Figura 7. Matriz sociométrica con los puntajes otorgados según la *Escala de la Distancia Social en el Aula* de Cunningham et al. (1951). Elaboración propia

### 2.2.6. Análisis de las respuestas II: Cálculo de los valores e índices sociométricos

El análisis de las respuestas finaliza con el cálculo de los valores e índices sociométricos, a partir de los datos contenidos en la sociomatríz. Según Arruga i Valeri (1974), los valores sociométricos son “la cuantificación de los criterios sociométricos investigados en un test” (p. 73), mientras que un índice sociométrico es “la relación existente entre dos valores sociométricos” (p. 105). Fernández Prados (2000) considera que la relación entre un valor sociométrico y el conjunto del grupo también es un índice sociométrico (p. 70). Los principales valores sociométricos se describen en la tabla 6.

Tabla 6

*Los principales valores sociométricos*

Valor sociométrico	Significado
Estatus de elecciones (Sp)	Cantidad de elecciones recibidas
Estatus de elecciones valorizadas (Sp val)	Suma de los puntajes positivos (elecciones ponderadas) recibidos
Estatus de rechazos (Sn)	Cantidad de rechazos recibidos
Estatus de rechazos valorizados (Sn val)	Suma de los puntajes negativos (rechazos ponderados) recibidos
Expansividad positiva (Ep)	Cantidad de elecciones realizadas
Expansividad negativa (En)	Cantidad de rechazos realizados
Elecciones recíprocas (Rp)	Cantidad de veces en que ambos se eligen mutuamente
Rechazos recíprocos (Rn)	Cantidad de veces en que ambos se rechazan mutuamente
Oposición de sentimiento (OS)	Cantidad de veces en que mutuamente uno elige y el otro rechaza

Nota: Basado en *Introducción al test sociométrico* por A. Arruga i Valeri, 1974, pp. 73-77. Copyright 1974 por Editorial Herder S. A.

Existen también valores sociométricos relacionados con el criterio de percepción sociométrica, como Pp (*Percepción de elecciones*) —la cantidad de individuos por los cuales el sujeto se cree elegido— y Pn (*Percepción de rechazos*) —la cantidad de individuos por los cuales el sujeto se cree rechazado—, entre otros.

Además, dos valores pueden medirse con la *Escala de la Distancia Social en el Aula* (*Classroom Social Distance Scale*): por un lado, un valor de *distancia social propia* (*self-social-distance*) que indica el grado de aceptación o rechazo del grupo por parte de un individuo; y por otro lado, un valor de *distancia social del grupo* (*group-social-distance*), que indica el grado de aceptación o rechazo de un individuo por parte del grupo (Cunningham et al., 1951, p. 172).

La distancia social propia es la suma de los puntajes otorgados por cada individuo a los demás miembros del grupo, y la distancia social del grupo es la suma de los puntajes recibidos. Portuondo (1971) proporciona una clara explicación del significado que tienen estos dos valores:

Cuanto más pequeña sea la cifra de la *distancia social del grupo* con respecto a un sujeto, mayor será la aceptación del sujeto por parte del grupo. Cuando la *distancia social del grupo* con respecto a un sujeto cualquiera es muy grande, será un claro índice de que el grupo lo rechaza o al menos de que no lo aprecia adecuadamente . . . Investigando también la *distancia social propia* de un sujeto con respecto a su grupo, es decir, los afectos de este hacia el grupo, es posible ver si estima más o menos al grupo de lo que el grupo lo estima a él (p. 33).

Desde un punto de vista psicopedagógico, es recomendable que los docentes empleen estrategias didácticas que permitan reducir las distancias sociales, ya que esto significaría que hay un mayor aprecio y una mayor aceptación de los estudiantes entre sí.

La distancia social *media* es un índice sociométrico, ya que es la relación entre un valor sociométrico (la distancia social, que es una suma de puntajes) y el tamaño del grupo, según se definió anteriormente.

Existen muchos otros índices sociométricos. Arruga i Valeri (1974) clasifica los más importantes en dos grupos: individuales —aquellos que hacen referencia a los sujetos— y globales —aquellos que hacen referencia al grupo— (p. 105).

Entre los índices sociométricos individuales listados por este autor, pueden mencionarse dos en cuyas fórmulas aparece N (el tamaño del grupo): Pop (*Índice de popularidad*), cuya fórmula es  $Pop = Sp / (N - 1)$  y Ant (*Índice de antipatía*), cuya fórmula es  $Ant = Sn / (N - 1)$ . Otros índices sociométricos individuales, en cambio, solo relacionan dos valores sociométricos entre sí, por ejemplo: CA (*Conexión afectiva*), cuya fórmula es  $CA = Rp / Sp$  y AP (*Atención perceptiva*), cuya fórmula es  $AP = Pp / Sp$ .

Entre los principales índices sociométricos globales, cabe mencionar los índices IC (*Índice de coherencia*) —que se obtiene calculando el cociente entre la cantidad total de elecciones recíprocas y la cantidad total de elecciones recibidas— e IA (*Índice de asociación o de cohesión*) —que es el cociente entre la cantidad total de elecciones recíprocas y la cantidad posible de elecciones recíprocas— (Arruga i Valeri, 1974, pp. 105-113).

### 2.2.7. Síntesis de los resultados I: Elaboración de los sociogramas

Bastin (1966) admite que la matriz sociométrica es un instrumento bastante limitado para representar la interacciones entre los miembros de un grupo:

Las elecciones y rechazos emitidos por todos los miembros del grupo constituyen una red de interacciones de gran complejidad, que de ningún modo se manifiesta a través de los cálculos presentados hasta aquí. Por su parte, la matriz sociométrica solo ofrece una idea muy vaga de dicha red de interacciones, ya que se trata de un instrumento de análisis, no de síntesis (p. 53).

No obstante, también reconoce que Moreno pudo subsanar esta limitación al inventar los sociogramas:

El gran mérito de Moreno consiste en haber propuesto . . . un medio muy simple de obtener informes bastante precisos acerca de la estructura de los grupos; ideó representar gráficamente las relaciones interpersonales, simbolizando cada miembro mediante un círculo o un triángulo . . . y cada relación por una línea que une a círculos y triángulos (p. 53).

En cambio, Arruga i Valeri (1974) es bastante crítico con los sociogramas, los cuales —en su opinión— son una técnica imprecisa si se los compara con la sociomatrix:

Esta técnica, fundamental en los inicios de la sociometría, con el progreso en la elaboración de la sociomatrix y de su significación estadística, está quedando cada vez más como una técnica complementaria. La exactitud de la significación estadística no tiene parangón con la imprecisión con que se realizan las estructuras del sociograma (p. 121)

## 2.2.8. Síntesis de los resultados II: Elaboración del informe sociométrico

El informe es básicamente una recopilación de todos los elementos elaborados en los pasos anteriores, acompañados de explicaciones y orientaciones para la intervención educativa. La tabla 7 lista las partes de que debería constar este informe.

**Tabla 7**

*Las partes del informe sociométrico*

Parte	Elementos
Protocolo	Ficha técnica Listado de alumnos Cuestionario modelo
Informe general del grupo	Sociomatriz y valores sociométricos Sociogramas de grupo Índices globales Estructuras grupales (díadas, triángulos, subgrupos) Explicación y análisis de los resultados obtenidos
Informes por alumno	Valores sociométricos individuales Sociogramas individuales Índices individuales Explicación y análisis de los resultados obtenidos
Orientaciones	Guías para la intervención educativa
Anexos	Cuestionarios utilizados

*Nota:* Basado en *Sociología de los grupos escolares: sociometría y dinámica de grupos* por J. S. Fernández Prados, 2000, p. 72. Copyright 2000 por Universidad de Almería. Servicio de Publicaciones

## 2.3. Validez del test sociométrico

Un test es válido si mide realmente lo que se supone que debe medir. Por ello, para poder ser considerado válido, el test sociométrico debe medir y evaluar las relaciones interpersonales y la estructura del grupo al que se le aplique este instrumento. Según Bastin (1966), “el test valdrá tanto cuanto valga la motivación de los sujetos que lo completan . . . La utilización de un criterio de elección que atraiga al niño y le induzca a responder sin rodeos, es una garantía de buena validez” (p. 104). Portuondo (1971) establece las siguientes condiciones básicas para lograr la validez del test: 1) que todos los miembros hayan entendido previamente las instrucciones; 2) que predominen el deseo de cooperación y la sinceridad; 3) que las preguntas sean adecuadas para el grupo y estén formuladas de manera clara y sencilla; y 4) que los resultados se elaboren de manera adecuada y sistemática (pp. 30-31).

# LA INTELIGENCIA EMOCIONAL

---

El término *inteligencia emocional* (IE) nació en alemán, cuando en 1966 Barbara Leuner lo usó en el título de su artículo *Emotionale Intelligenz und Emanzipation*, en el que “se plantea el tema de cómo muchas mujeres rechazan un rol social a causa de su baja inteligencia emocional” (Bisquerra Alzina, 2009, p. 126). Mucho más tarde, Salovey y Mayer (1990) lo dotaron de una base teórica sólida, definiéndolo como

el subconjunto de la inteligencia social que involucra la capacidad de percibir las emociones y los sentimientos de uno mismo y los de los demás, de discernirlos y de usar esa información para guiar el pensamiento y las acciones propios (p. 189).

Sin embargo, fue recién en 1995 cuando se popularizó el término, al ser usado por Daniel Goleman en el título de su bestseller *Emotional Intelligence: Why it can matter more than IQ*, donde relacionó las emociones, el cerebro humano y la alfabetización emocional de una manera nunca antes vista en la literatura académica (Roy, 2015, pp. 11-12).

Efectivamente, en ese libro, Goleman (1996) señaló que, en ocasiones, más decisiva que el cociente intelectual puede resultar la inteligencia emocional, que definió como

la capacidad de motivarnos a nosotros mismos, de perseverar en el empeño a pesar de las posibles frustraciones, de controlar los impulsos, . . . de regular nuestros estados de ánimo, de evitar que la angustia interfiera con nuestras facultades racionales y, por último —pero no, por ello, menos importante—, la capacidad de empatizar y confiar en los demás (pp. 26-27).

Mayer y Salovey (1997) luego combinaron la idea de que la emoción permite pensar de manera más inteligente, por un lado, con la idea de que uno piensa inteligentemente sobre las emociones, por el otro, y definieron la inteligencia emocional como la capacidad de percibir emociones, de acceder y generar emociones para ayudar al pensamiento, de comprender las emociones y el conocimiento emocional, y de regular reflexivamente las emociones para promover el crecimiento emocional e intelectual (p. 5).

Como se puede observar al analizar la definición propuesta por estos autores en 1990, ellos originalmente veían la inteligencia emocional como una parte de la *inteligencia social*, aunque luego parecen haber querido separar estos dos constructos, ya que en la definición de 1997 no aparece —ni siquiera implícito— el aspecto social.

Por el contrario, Bar-On (2006) afirma que es más preciso referirse a este constructo como ESI (*Emotional-Social Intelligence*), es decir, *inteligencia emocional-social* —como figura en su artículo— o *socioemocional* —que es una denominación más frecuente en español—, en lugar de *inteligencia emocional* o *inteligencia social*, y proporciona la siguiente definición:

La inteligencia emocional-social es un corte transversal de competencias, habilidades y facilitadores emocionales y sociales interrelacionados, que determinan cuán efectivamente nos entendemos a nosotros mismos y nos expresamos, entendemos a los demás y nos relacionamos con ellos, y nos enfrentamos con las demandas diarias (p. 14).

Petrides y Furnham (2001) establecieron una clara división —que persiste hasta hoy en día— entre la inteligencia emocional rasgo (*trait emotional intelligence*) y la inteligencia emocional capacidad (*ability emotional intelligence*), ya que ambas son dos constructos diferentes:

El primero abarca *disposiciones conductuales* y *capacidades autopercebidas* y se mide a través de cuestionarios de autoinforme, mientras que el segundo se refiere a las *capacidades reales* y debe medirse usando tests de rendimiento máximo . . . Debido a que la *inteligencia emocional rasgo* se relaciona con las tendencias conductuales y las capacidades autopercebidas, su estudio debe realizarse dentro del marco de la personalidad. Por el contrario, la *inteligencia emocional capacidad* debe estudiarse principalmente con respecto a la inteligencia psicométrica. Dado que la inteligencia y la personalidad son dominios esencialmente independientes, uno esperaría que la *inteligencia emocional rasgo* se relacionara con la personalidad, pero no con los factores de capacidad. Por otro lado, la *inteligencia emocional capacidad* debería estar relacionada con la capacidad cognitiva (especialmente con *g*), pero también debería correlacionarse con aquellas dimensiones de la personalidad que tienen un núcleo afectivo fuerte (p. 426).

### 3.1. La importancia de evaluar la inteligencia emocional

Independientemente de la definición que se acepte, no hay duda de que la inteligencia emocional es un constructo de gran importancia en innumerables áreas.



Para los educadores, por ejemplo, la importancia de la inteligencia emocional radica en que “influye de forma decisiva en la adaptación psicológica del alumno en clase, en su bienestar emocional e, incluso, en sus logros académicos y en su futuro laboral” (Extremera Pacheco y Fernández-Berrocal, 2004, p. 1).

Según estos autores, las habilidades de la inteligencia emocional deben enseñarse en las escuelas, y su evaluación implica obtener un perfil emocional de los estudiantes:

La evaluación de la IE en el aula supone una valiosa información para el docente en lo que respecta al conocimiento del desarrollo afectivo de los alumnos e implica la obtención de datos fidedignos que marquen el punto de inicio en la enseñanza transversal (p. 1).

En la tabla 8 se resumen las características de los principales instrumentos disponibles para la evaluación de la inteligencia emocional: los tests de ejecución típica (cuestionarios de autoinforme), la evaluación 360° y los tests de rendimiento máximo.

**Tabla 8**

*Principales instrumentos disponibles para la evaluación de la inteligencia emocional*

Instrumento	Descripción
Medidas de evaluación de la IE basadas en cuestionarios de autoinforme (tests de ejecución típica)	En la mayoría de los casos, estos cuestionarios están formados por enunciados verbales cortos en los que el alumno evalúa su IE mediante la propia estimación de sus niveles en determinadas habilidades emocionales a través de una escala Likert. Este indicador se denomina <i>índice de inteligencia emocional percibida o autoinformada</i> y revela las creencias del alumno sobre si puede percibir, discriminar y regular sus emociones.
Medidas de evaluación de la IE basadas en observadores externos	Si la IE implica la capacidad para manejar y comprender las emociones de las personas que nos rodean, ¿por qué no preguntar a nuestras personas más cercanas cómo manejamos nuestras emociones? Este procedimiento utiliza instrumentos basados en la observación externa o evaluación 360°: En ellos se solicita la estimación por parte de los compañeros de clase o del profesor para que den su opinión sobre cómo el alumno es percibido con respecto a su interacción con el resto de los compañeros, su manera de resolver los conflictos en el aula o bien su forma de afrontar las situaciones de estrés.
Medidas de evaluación de la IE basadas en tareas de ejecución (tests de rendimiento máximo)	A fin de determinar si cierto alumno es el mejor pianista del colegio para actuar en una obra musical, no se le solicita que diga cuán bien cree él que toca el piano o no se le pregunta a un amigo si el alumno sabe tocar el piano. La prueba requiere que el propio alumno muestre su habilidad tocando el piano. En general, las medidas de habilidad de la IE consisten en un conjunto de tareas emocionales que evalúan cómo un estudiante resuelve ciertos problemas emocionales comparando sus respuestas con criterios de puntuación objetivos. Por ejemplo, para evaluar la percepción emocional se examina si el alumno es capaz de reconocer emociones en ciertas expresiones faciales. Igualmente, para evaluar la capacidad de manejo emocional se valorará la idoneidad de las estrategias de solución de problemas seleccionadas por el alumno para resolver un conflicto interpersonal.

*Nota:* Basado en “La inteligencia emocional: métodos de evaluación en el aula” por N. Extremera Pacheco y P. Fernández-Berrocal, 2004, *Revista Iberoamericana de Educación*, pp. 1-12. Copyright 2004 por OEI

## 3.2. Algunos modelos teóricos de la inteligencia emocional

A partir de la década de 1990, numerosos estudiosos de la inteligencia emocional han propuesto modelos teóricos para darle sostén a este constructo. A continuación se tratan los principales de estos modelos, incluyendo —en cada caso— algunos instrumentos de evaluación derivados de ellos.

### 3.2.1. Modelos de Salovey y Mayer

Peter Salovey —de la *Yale University*— y John D. Mayer —de la *University of New Hampshire*— publicaron su primer modelo teórico de la inteligencia emocional en 1990. De acuerdo con este modelo, la inteligencia emocional es una capacidad cognitiva compuesta de tres procesos mentales conceptualmente relacionados e involucrados en el procesamiento de la información afectiva: la *valoración y expresión de las emociones* (en uno mismo y en los demás), la *regulación de las emociones* (en uno mismo y en los demás) y el *uso de las emociones* de manera adaptativa (figura 8).

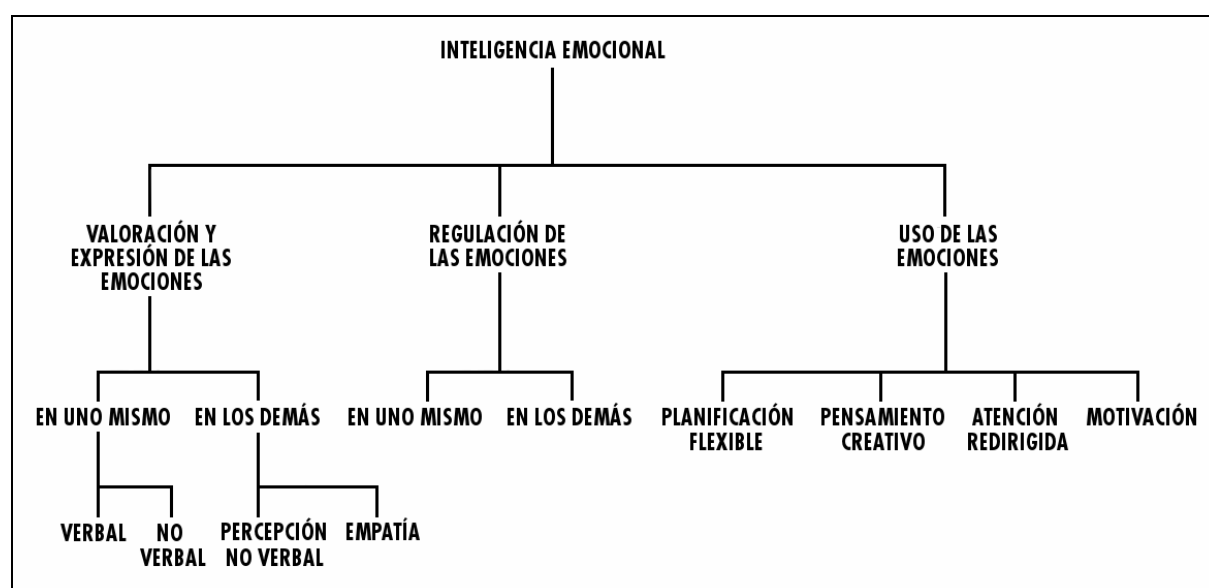


Figura 8. Modelo de inteligencia emocional según Salovey y Mayer (1990). Adaptado de “Emotional intelligence” por P. Salovey y J. D. Mayer, 1990, *Imagination, cognition and personality*, 9(3), p. 190. Copyright 1990 por SAGE Publications

El principal instrumento de evaluación de la inteligencia emocional derivado de este modelo es la TMMS (*Trait Meta-Mood Scale*), un cuestionario de autoinforme que en su versión original consta de 48 ítems, aunque sus autores recomiendan utilizar una

versión reducida de 30 ítems que es más eficiente (Salovey, Mayer, Goldman, Turvey y Palfai, 1995, p. 132). Permite medir tres dimensiones: la *atención a los sentimientos* —*attention to feelings*, es decir, en qué grado las personas creen prestar atención a sus sentimientos—, la *claridad de los sentimientos* —*clarity of feelings*, o sea, cómo las personas creen percibir sus sentimientos— y la *reparación del estado emocional* —*mood repair*, es decir, en qué medida las personas se creen capaces de interrumpir los estados emocionales negativos y prolongar los positivos— (p. 129).

Existen muchas variantes de esta escala, como la TMMS-24 creada en España (Fernández-Berrocal, Extremera Pacheco y Ramos, 2004) y la TMMS-21 desarrollada en la República Argentina (Calero, 2013). En la figura 9 se muestran algunos ítems típicos.

<b>TMMS</b>							
<b>Instrucciones.</b> A continuación encontrará algunas afirmaciones sobre sus emociones y sentimientos. Lea atentamente cada frase y decida la frecuencia con la que Ud. cree que se produce cada una de ellas marcando con una X en la casilla correspondiente							
1	2	3	4	5			
Nunca	Raramente	Algunas veces	Con bastante frecuencia	Muy frecuentemente			
1) Normalmente conozco mis sentimientos sobre las personas			1	2	3	4	5
2) Intento tener pensamientos positivos aunque me sienta mal			1	2	3	4	5
3) Presto mucha atención a cómo me siento			1	2	3	4	5

Figura 9. Ejemplos de ítems del instrumento TMMS. Adaptado de “La Inteligencia Emocional y la educación de las emociones desde el Modelo de Mayer y Salovey” por P. Fernández-Berrocal y N. Extremera Pacheco, 2005, *Revista Interuniversitaria de Formación del Profesorado*, 19(3), p. 75. Copyright 2005 por AUFOP

Más tarde, los propios autores admitieron que el modelo era impreciso en algunos lugares y pobre en el sentido de que solo hablaba sobre *percibir* y *regular* la emoción, y omitía la posibilidad de *pensar* en los sentimientos (Mayer y Salovey, 1997, p. 10).

Por ello, en 1997, Mayer y Salovey publicaron un segundo modelo teórico de la inteligencia emocional (figura 10).

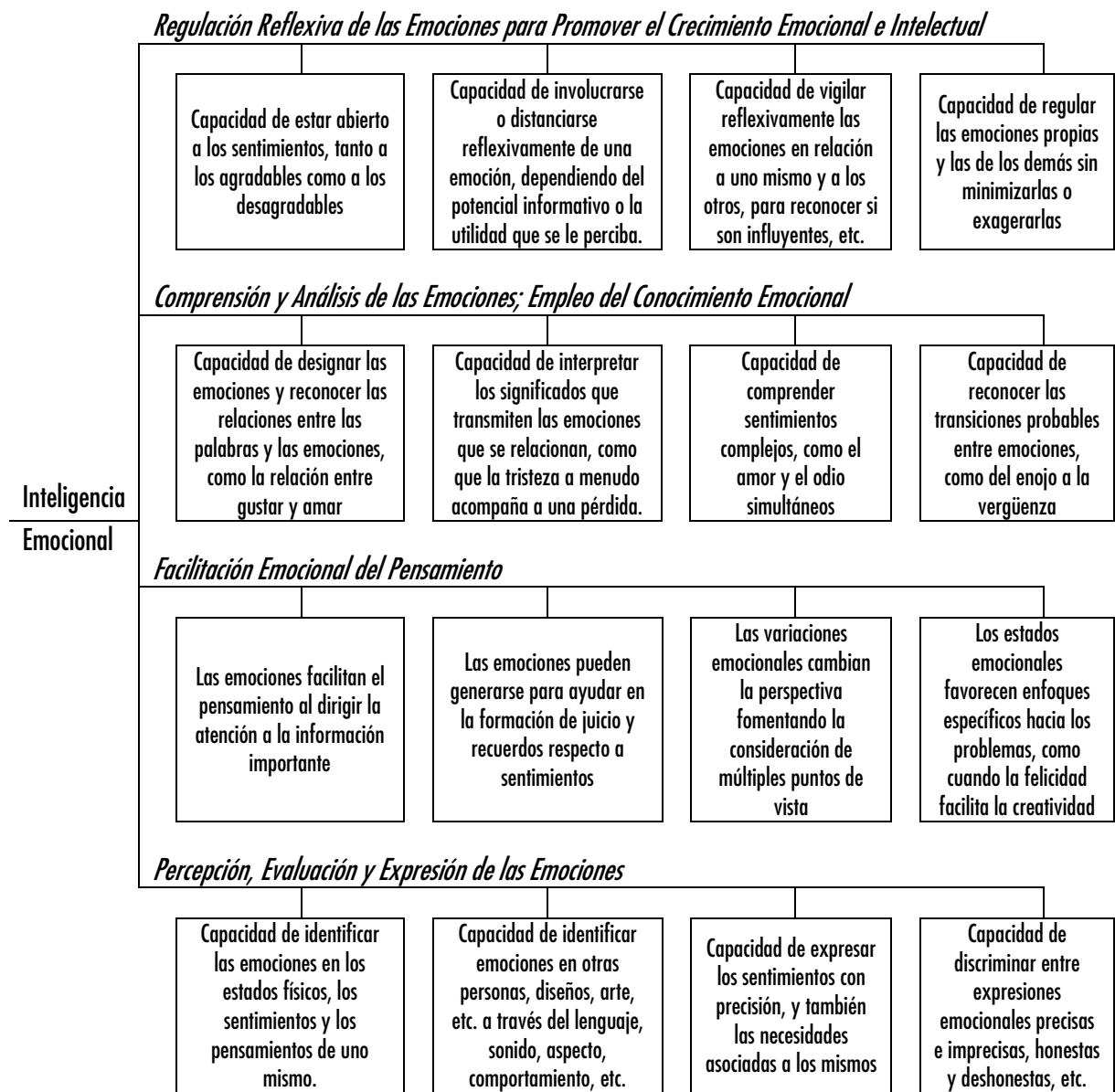


Figura 10. Modelo de inteligencia emocional según Mayer y Salovey (1997). Adaptado de “What is emotional intelligence?” por J. D. Mayer y P. Salovey, 1997, *Emotional development and emotional intelligence: Educational implications* editado por J. D. Mayer y D. J. Sluyter, p. 11. New York: Basic Books. Copyright 1997 por Harper Collins Publishers

En este modelo, la inteligencia emocional se conceptualiza como cuatro capacidades generales estructuradas de manera jerárquica, desde el nivel inferior —la *percepción, evaluación y expresión de las emociones*— hasta el nivel más alto y complejo —la *regulación reflexiva de las emociones para promover el crecimiento emocional e intelectual*—.

El principal instrumento de evaluación de la inteligencia emocional derivado de este modelo es el MSCEIT (*Mayer-Salovey-Caruso Emotional Intelligence Test*), que es una prueba basada en tareas de ejecución o de capacidad que tienen un rendimiento máximo, tal como ocurre en el caso de los tests usados para medir otras inteligencias, ya que, según este modelo, la inteligencia emocional no es más que un sistema de capacidades dedicado al procesamiento de la información emocional. Fue desarrollado por Mayer y Salovey —a quienes se sumó David R. Caruso— en cooperación con *Multi-Health Systems Inc.*, una empresa que comercializa instrumentos de evaluación («MHS Inc.», s. f.).

Existen distintas versiones de este test. En el MSCEIT v. 2.0, los participantes deben resolver ocho tareas emocionales diversas —dos por cada una de las cuatro capacidades generales del modelo—, habiendo en total 141 ítems. Las puntuaciones se obtienen mediante dos criterios: *experto* —basado en la opinión de 21 expertos en el campo emocional— y *consenso* —basado en las respuestas de una muestra heterogénea de más de 5 000 individuos— (Fernández-Berrocal y Extremera Pacheco, 2005, p. 76).

También existen adaptaciones locales, como una desarrollada en España (Extremera Pacheco y Fernández-Berrocal, 2002) y otra creada en la República Argentina (Mikulic, Caballero, Crespi y Radusky, 2013). En la figuras 11 y 12 se muestran algunos ítems típicos.

**MSCEIT**

**Percepción emocional**



**Instrucciones:** ¿En qué medida este rostro refleja las siguientes emociones?

1)	Nada de Felicidad	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">2</td> <td style="padding: 2px 5px;">3</td> <td style="padding: 2px 5px;">4</td> <td style="padding: 2px 5px;">5</td> </tr> </table>	1	2	3	4	5	Felicidad Extrema
1	2	3	4	5				
2)	Nada de Miedo	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">2</td> <td style="padding: 2px 5px;">3</td> <td style="padding: 2px 5px;">4</td> <td style="padding: 2px 5px;">5</td> </tr> </table>	1	2	3	4	5	Miedo Extremo
1	2	3	4	5				

Figura 11. Ejemplos de ítems de percepción emocional del MSCEIT. Adaptado de “La Inteligencia Emocional y la educación de las emociones desde el Modelo de Mayer y Salovey” por P. Fernández-Berrocal y N. Extremera Pacheco, 2005, *Revista Interuniversitaria de Formación del Profesorado*, 19(3), p. 77. Copyright 2005 por AUFOP

<b>MSCEIT</b>				
<b>Manejo emocional</b>				
Luisa acaba de venir de sus vacaciones. Se siente relajada, animada y llena de energía. ¿En qué medida cada una de las siguientes acciones ayudarían a Luisa a mantener esas emociones?				
<b>Acción 1:</b> Luisa empezó a hacer una lista de las cosas que tenía que hacer				
1. Muy ineficaz	2. Algo ineficaz	3. Neutral	4. Algo eficaz	5. Muy eficaz
<b>Acción 2:</b> Empezó a pensar sobre a dónde iría en sus próximas vacaciones				
1. Muy ineficaz	2. Algo ineficaz	3. Neutral	4. Algo eficaz	5. Muy eficaz
<b>Acción 3:</b> Luisa decidió que lo mejor era ignorar esos sentimientos				
1. Muy ineficaz	2. Algo ineficaz	3. Neutral	4. Algo eficaz	5. Muy eficaz
<b>Acción 4:</b> Llamó a una amiga para contarle lo bien que se lo había pasado durante sus vacaciones				
1. Muy ineficaz	2. Algo ineficaz	3. Neutral	4. Algo eficaz	5. Muy eficaz

Figura 12. Ejemplos de ítems de manejo emocional del MSCEIT. Adaptado de “La Inteligencia Emocional y la educación de las emociones desde el Modelo de Mayer y Salovey” por P. Fernández-Berrocal y N. Extremera Pacheco, 2005, *Revista Interuniversitaria de Formación del Profesorado*, 19(3), p. 78. Copyright 2005 por AUFOP

### 3.2.2. Modelo de Petrides y Furnham

Cuando Konstantinos V. Petrides y Adrian Furnham —del *University College London*— señalaron la división existente entre la inteligencia emocional *rasgo* y la inteligencia emocional *capacidad*, establecieron un modelo de IE rasgo mediante la inclusión de diversas dimensiones presentes en los principales modelos de inteligencia emocional encontrados en la literatura científica disponible hasta ese momento:

La inteligencia emocional rasgo abarca diversas disposiciones del dominio de la personalidad, como la *empatía* y la *asertividad* consideradas por Goleman, así como elementos de la *inteligencia social* de Thorndike, la *inteligencia personal* de Gardner y la *inteligencia emocional capacidad* de Mayer y Salovey (Petrides y Furnham, 2001, p. 427).

Metodológicamente hablando, empezaron por allí debido a que “el primer paso para la operacionalización de un constructo psicológico implica la definición previa de su dominio muestral, es decir, determinar qué facetas (elementos o dimensiones) integran dicho constructo” (Pérez-González, Petrides y Furnham, 2007, p. 91).

El dominio muestral que surgió de la comparación de los modelos preexistentes, según estos autores, “incluyó las facetas compartidas pero excluyó aquellas que eran peculiares de un único modelo” (p. 94), resultando en un nuevo modelo formado por 15 facetas que son consideradas todas del mismo nivel de importancia relativa para la formación de la inteligencia emocional global. De esas facetas, dos son independientes y 13 se agrupan en cuatro factores. La representación de este modelo se muestra en la figura 13.

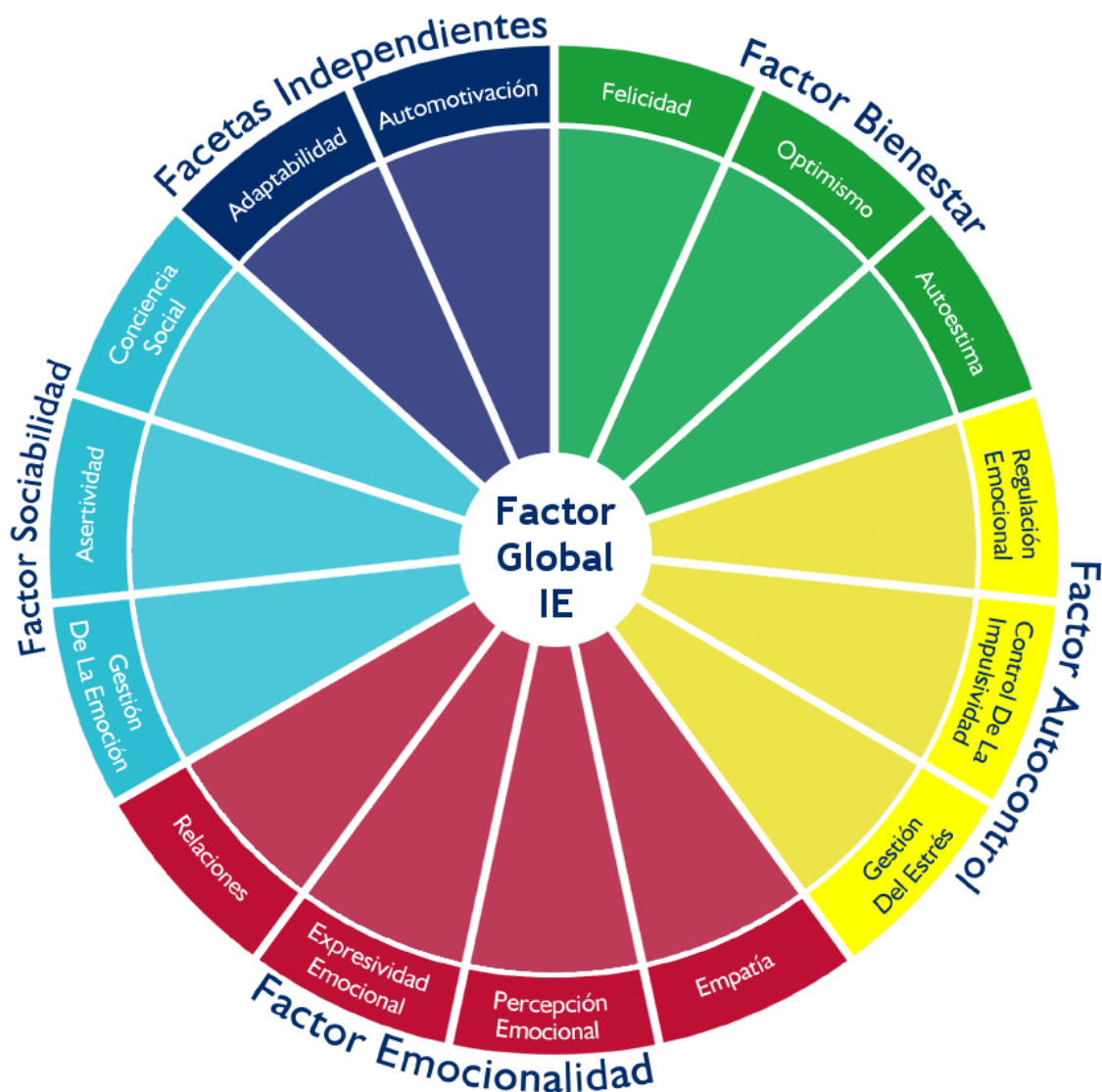


Figura 13. Modelo de inteligencia emocional según Petrides y Furnham. Adaptado del sitio web de Thomas International Ltd. —la empresa que comercializa oficialmente el TEIQue, un instrumento basado en este modelo— («Thomas International Ltd.», s. f.). Copyright 2011 por K. V. Petrides & Thomas International Limited

El *Factor Global IE* mide el funcionamiento emocional general. Los otros cuatro factores son el *Bienestar* —que mide cuán feliz y plena es una persona—, el *Autocontrol* —que mide cuán bien una persona regula el estrés y controla sus impulsos—, la *Emocionalidad* —la capacidad de percibir y expresar emociones y utilizar la percepción de las emociones para desarrollar y mantener relaciones estrechas con los demás— y la *Sociabilidad* —que mide cuán socialmente capaz es una persona y con cuánta confianza puede comunicarse con otros— (Banerji y Tate, 2013, p. 2).

En la tabla 9 se presenta la descripción de las facetas que constituyen el dominio muestral de este modelo.

**Tabla 9**

*Dominio muestral del modelo de Petrides y Furnham*

Factor	Faceta	Quien alcanza valores altos en los cuestionarios de autoinforme se ve a sí mismo...
	Adaptabilidad	...flexible y dispuesto a adaptarse a las nuevas condiciones
	Automotivación	...poco propenso a renunciar frente a la adversidad
	Autocontrol	
	Control de la Impulsividad	...reflexivo y menos propenso a ceder a sus impulsos
	Gestión del Estrés	...capaz de soportar la presión y regular el estrés
	Regulación Emocional	...capaz de controlar sus emociones
Bienestar	Autoestima	...exitoso y seguro de sí mismo
	Felicidad	...alegre y satisfecho con su vida
	Optimismo	...lleno de confianza y propenso a <i>mirar el lado brillante</i> de la vida
Emocionalidad	Empatía	...capaz de tomar la perspectiva de otra persona
	Expresividad Emocional	...capaz de comunicar sus sentimientos a los demás
	Percepción Emocional	...claro acerca de los sentimientos propios y ajenos
	Relaciones	...capaz de mantener relaciones personales satisfactorias
Sociabilidad	Asertividad	...franco y dispuesto a luchar por sus derechos
	Conciencia Social	...tejedor de redes con habilidades sociales superiores
	Gestión de la Emoción	...capaz de influir en los sentimientos de otras personas

*Nota:* Basado en “Psychometric Properties of the Trait Emotional Intelligence Questionnaire (TEIQue)” por K. V. Petrides, 2009, *Assessing Emotional Intelligence* editado por C. Stough, D. H. Saklofske y J. D. A. Parker, 2009, p. 89. Copyright 2009 por Springer Science+Business Media, LLC

Por considerar que la inteligencia emocional es un *rasgo* de la personalidad en lugar de una *capacidad*, Petrides (2009) cuestiona el uso de tests de rendimiento máximo en el ámbito emocional:



¿Hay realmente formas de sentir correctas e incorrectas, de la misma manera que existen, por ejemplo, analogías verbales correctas e incorrectas? ¿Están los expertos mejor posicionados para decirnos cómo se siente una persona adulta típica que la propia persona? Las personas que no pueden adivinar lo que un músico podría estar sintiendo al ejecutar una pieza, ¿son emocionalmente deficientes? (p. 86)

En cambio, defiende la utilización de cuestionarios de autoinforme como forma estándar de efectuar la evaluación de la inteligencia emocional rasgo, debido a que en este modelo se le concede más importancia a la autopercepción del individuo sobre sus habilidades emocionales que a sus capacidades reales.

El principal instrumento de evaluación de la inteligencia emocional derivado de este modelo es el TEIQue (*Trait Emotional Intelligence Questionnaire*), que es precisamente un cuestionario de autoinforme publicado por Petrides en 2001 (Petrides y Furnham, 2003, p. 47). El TEIQue está formado por 153 ítems, de los cuales 144 están relacionados con las facetas. La tabla 10 se muestran algunos de estos ítems característicos.

**Tabla 10**

*Ítems característicos del TEIQue*

Faceta	Cantidad de ítems	Ítem característico (R = escala Likert reversa)
Adaptabilidad	9	Por lo general, me resulta difícil hacer ajustes en mi estilo de vida. (R)
Asertividad	9	Cuando no estoy de acuerdo con alguien, generalmente me resulta fácil decirlo.
Autoestima	11	Creo que estoy lleno de fortalezas personales.
Automotivación	10	Tiendo a sentir mucho placer por el solo hecho de hacer algo bien.
Conciencia Social	11	Puedo tratar de forma efectiva con la gente.
Control de la Impulsividad	9	Tiendo a <i>dejar</i> me llevar fácilmente. (R)
Empatía	9	Me resulta difícil entender por qué ciertas personas se molestan con ciertas cosas. (R)
Expresividad Emocional	10	Otros me dicen que rara vez hablo sobre cómo me siento. (R)
Felicidad	8	La vida es bella.
Gestión de la Emoción	9	Por lo general, soy capaz de influir en la forma en que otras personas se sienten.
Gestión del Estrés	10	Por lo general, soy capaz de lidiar con problemas a otros les parecen molestos.
Optimismo	8	Generalmente creo que las cosas van a salir bien en mi vida.
Percepción Emocional	10	A menudo me resulta difícil reconocer qué emoción siento. (R)
Regulación Emocional	12	Cuando alguien me ofende, normalmente puedo mantener la calma.
Relaciones	9	Generalmente no me mantengo en contacto con amigos. (R)

*Nota:* Basado en “Trait emotional intelligence: behavioural validation in two studies of emotion recognition and reactivity to mood induction” por K. V. Petrides y A. Furnham, 2003, *European Journal of Personality*, 17(1), p. 47. Copyright 2002 por John Wiley & Sons, Ltd.

De los nueve ítems que no están relacionados con ninguna faceta, el último (*Algunas de mis respuestas a este cuestionario no son 100% sinceras*) es un autoinforme de honestidad que permite descartar participantes, en caso de considerárselo necesario.

Los ítems se puntúan según una escala tipo Likert de siete valores, desde *totalmente en desacuerdo* hasta *totalmente de acuerdo*, siendo que cada elección equivale a un valor numérico de 1 a 7 en las preguntas estándar o de 7 a 1 en las preguntas que utilizan la escala reversa.

El valor de cada faceta es la media aritmética de los puntajes asignados a sus correspondientes ítems. El valor de cada factor es la media aritmética de los puntajes asignados a los ítems de sus correspondientes facetas (el valor del Factor Global IE, por lo tanto, es la media aritmética de los puntajes asignados a los 144 ítems que están relacionados con alguna faceta).

Tanto la consistencia interna de este instrumento (alfa de Cronbach en alrededor de 0,85) como su fiabilidad test-retest (entre 0,50 y 0,82; con 0,78 para la puntuación global; periodo de 12 meses), en general, son buenas (Pérez-González, Petrides y Furnham, 2005, p. 131).

El TEIQue —distribuido oficialmente por *Thomas International Ltd.*— es utilizado por empresas de todo el mundo para “conocer las emociones de sus empleados y sus candidatos, saber cómo gestionan las relaciones, seleccionar a candidatos emocionalmente inteligentes, fomentar el compromiso y la lealtad, emplear una comunicación eficaz y ajustar sus habilidades de gestión” («Thomas International Ltd.», s. f.).

Para el uso no comercial —por ejemplo, para el uso con fines académicos—, es posible acceder a la principal documentación sobre este instrumento en el sitio web del *London Psychometric Laboratory* («Psychometric Lab», s. f.) que Petrides dirige en el *University College London*. Allí también hay disponible un sistema *online* que —a partir de los puntajes subidos en una plantilla implementada en Microsoft Excel— genera un fichero con los valores de las facetas y los factores calculados automáticamente.

Actualmente, del sitio web anterior se puede descargar el TEIQue en más de 20 idiomas, incluyendo una versión traducida al español por Juan Carlos Pérez-González —de la UNED (*Universidad Nacional de Educación a Distancia*)—.

Además de la forma estándar del TEIQue —que es la que se ha presentado hasta aquí—, también existen otras que se detallan en la tabla 11.

**Tabla 11**

*Versiones del TEIQue disponibles*

Versión completa		Versión reducida		Tipo de cuestionario	Edad de los destinatarios
Denominación	Cantidad de ítems	Denominación	Cantidad de ítems		
TEIQue	153	TEIQue-SF	30	Autoinforme	Mayores de 17 años
TEIQue-AF	153	TEIQue-ASF	30	Autoinforme	De 13 a 17 años
TEIQue-CF	75	TEIQue-CSF	36	Autoinforme	De 8 a 12 años
TEIQue 360°	153	TEIQue 360°-SF	30	Observador externo	Mayores de 17 años

*Nota:* Basado en “Psychometric Properties of the Trait Emotional Intelligence Questionnaire (TEIQue)” por K. V. Petrides, 2009, *Assessing Emotional Intelligence* editado por C. Stough, D. H. Saklofske y J. D. A. Parker, 2009, pp. 97-98. Copyright 2009 por Springer Science+Business Media, LLC

### 3.2.3. Otros modelos

Rafael Bisquerra Alzina (2009) distingue tres categorías de modelos de inteligencia emocional: los modelos de *capacidad* como el de Salovey y Mayer, los modelos de *rasgo* como el de Petrides y Furnham y, por último, los modelos *mixtos* como el de Bar-On y el de Goleman (p. 134).

Acerca de esta última categoría, Mayer, Caruso y Salovey (1999) advierten que los modelos que engloba deben analizarse cuidadosamente para distinguir los conceptos que forman parte de la inteligencia emocional de los conceptos que se mezclan o se confunden:

La inteligencia emocional a menudo se ha conceptualizado (particularmente en la literatura popular) como algo que involucra mucho más que la capacidad de percibir, asimilar, comprender y manejar las emociones. Estas concepciones alternativas incluyen no solo la emoción y la inteligencia per se, sino también la motivación, una serie de disposiciones y rasgos que no son capacidades, y el funcionamiento personal y social global . . . Tal amplitud parece socavar la utilidad de los términos en consideración. Denominamos a estas concepciones *modelos mixtos* porque combinan muchas ideas diversas. Por ejemplo, el EQ-i (*Emotional Quotient Inventory*) de Bar-On incluye 15 escalas de autoinforme que miden la autoestima de una persona, la independencia, la resolución de problemas, la comprobación de la realidad y otros atributos . . . Tales cualidades, como la resolución de problemas y la comprobación de la realidad, parecen estar más estrechamente relacionadas con la fortaleza del ego o con la competencia social, que con la inteligencia emocional (p. 268).

En la tabla 12 se sintetizan las características de dos de los principales modelos mixtos en uso hoy en día: el modelo de la *Inteligencia Emocional-Social* —descendiente directo de la obra de Reuven Bar-On (1997) y actualizado por la empresa *Multi-Health Systems Inc.* («MHS Inc.», s. f.)— y el modelo de la *Competencia Emocional y Social* —derivado del trabajo de Daniel Goleman (1998) y actualizado por el propio Goleman, junto con Richard Boyatzis y un equipo de la empresa *Hay Group* («Korn Ferry», s. f.)—.

**Tabla 12**

*Modelos de inteligencia emocional mixtos*

Modelo	Dominio muestral	Instrumento	Año	Ítems
ESI ( <i>Emotional-Social Intelligence</i> )	Percepción propia: 1. Autoconcepto 2. Autoactualización 3. Autoconciencia emocional Interpersonal: 4. Relaciones interpersonales 5. Empatía 6. Responsabilidad social Toma de decisiones: 7. Resolución de problemas 8. Comprobación de la realidad 9. Control de impulsos Autoexpresión 10. Expresión emocional 11. Asertividad 12. Independencia Manejo del estrés: 13. Flexibilidad 14. Tolerancia al estrés 15. Optimismo	EQ-i 2.0 ( <i>Emotional Quotient Inventory 2.0</i> )	2012	133
ESC ( <i>Emotional and Social Competency</i> )	Autoconciencia: 1. Autoconciencia Emocional Autogestión: 2. Autocontrol emocional 3. Adaptabilidad 4. Orientación al logro 5. Panorama positivo Conciencia social: 6. Empatía 7. Conciencia Organizacional Gestión de las relaciones: 8. Entrenador y Mentor 9. Liderazgo inspirador 10. Influencia 11. Manejo de conflictos 12. Trabajo en equipo	ESCI 2 ( <i>Emotional and Social Competency Inventory 2</i> )	2011	68

*Nota:* Elaboración propia en base a información del sitio web del Consortium for Research on Emotional Intelligence in Organizations («Consortium for Research on Emotional Intelligence in Organizations», s. f.)

### 3.3. Las competencias emocionales

En 1988, Carolyn Saarni —de la *Sonoma State University*— utilizó el término *competencia emocional* al presentarse en el *Nebraska Symposium on Motivation* con la ponencia *Emotional competence: How emotions and relationships become integrated* (La Competencia Emocional: cómo se integran las emociones y las relaciones).

Sin embargo, este término tardó un década en popularizarse, ya que, según Díaz Matarranz y González Urbano (2011), recién ganó impulso cuando se despertó el interés por la aplicación práctica de la *inteligencia emocional*. Fue así como “el concepto de *competencia emocional* adquirió importancia en principio en el ámbito laboral incluyéndose como elemento a valorar en los aspirantes por los departamentos de selección de personal” (p. 194).

En efecto, fue recién en 1998 —tres años después lanzar su bestseller sobre la inteligencia emocional— que Goleman definió el concepto de *competencia emocional* en su libro *La práctica de la inteligencia emocional*:

Una competencia emocional es una capacidad adquirida basada en la inteligencia emocional que da lugar a un desempeño laboral sobresaliente . . . El hecho de poseer una elevada inteligencia emocional no garantiza que la persona haya aprendido las competencias emocionales que más importan en el mundo laboral sino tan solo que está dotada de un excelente potencial para desarrollarlas (p. 33).

Asimismo, Bisquerra Alzina y Pérez Escoda (2007) definen las competencias emocionales como “el conjunto de conocimientos, capacidades, habilidades y actitudes necesarias para comprender, expresar y regular de forma apropiada los fenómenos emocionales” (p. 69). También destacan su importancia:

Entendemos que las competencias emocionales son un aspecto importante de la ciudadanía efectiva y responsable; su dominio . . . potencia una mejor adaptación al contexto; y favorece un afrontamiento a las circunstancias de la vida con mayores probabilidades de éxito. Entre los aspectos que se ven favorecidos por las competencias emocionales están los procesos de aprendizaje, las relaciones interpersonales, la solución de problemas, la consecución y mantenimiento de un puesto de trabajo, etc. (p. 69).

En consecuencia, es recomendable que los docentes empleen estrategias didácticas que permitan que los estudiantes desarrollen competencias emocionales, ya que esto haría

aumentar las probabilidades de que triunfen en la vida. Es lo que estos autores denominan educación emocional:

Propugnamos la educación emocional entendida como un proceso educativo, continuo y permanente, que pretende potenciar el desarrollo de las competencias emocionales como elemento esencial del desarrollo integral de la persona, con objeto de capacitarle para la vida. El objetivo de la educación emocional es el desarrollo de competencias emocionales (p. 75).

El juego es una actividad tan importante que, ya en 1938, Huizinga (1972) incluso llegó a afirmar que precede a la cultura:

Las grandes ocupaciones primordiales de la convivencia humana están ya impregnadas de juego . . . Mediante el *mito*, el hombre primitivo trata de explicar lo terreno . . . En cada una de esas caprichosas fantasías con que el mito reviste lo existente juega un espíritu inventivo . . . Fijémonos también en el *culto*: la comunidad primitiva realiza sus prácticas sagradas, . . . sus consagraciones, sus sacrificios y sus misterios, en un puro juego . . . Ahora bien, en el mito y en el culto es donde tienen su origen las grandes fuerzas impulsivas de la vida cultural: derecho y orden, tráfico, ganancia, artesanía y arte, poesía, erudición y ciencia. Todo esto hunde así sus raíces en el terreno de la actividad lúdica (p. 16).

Para destacar la importancia del juego, Csíkszentmihályi (1982) sugiere que la invención de los automóviles, los aviones y las computadoras en el siglo XX tuvo más una motivación de carácter lúdico que de cualquier otro tipo (p. 160).

#### 4.1. La naturaleza del juego

La naturaleza del juego es compleja. Debido a ello, a lo largo de la historia han surgido numerosas teorías para intentar explicarla. Según Munné (1980), la concepción actual tiene bases que se remontan a los antiguos filósofos griegos y a Kant:

Platón dice en las *Leyes* que en los jóvenes el juego se debe a que no pueden mantenerse en reposo, por lo que les es placentero saltar, gritar, danzar y jugar unos con otros. Aristóteles, en la *Ética nicomáquea* compara el juego —*παιδιά*— con la felicidad y la virtud; todas ellas son actividades *deseables* por sí mismas y no son necesarias como el trabajo. Pero el juego no es, como aquellas, una actividad *valiosa* por sí misma, sino en la medida en que nos capacita para la actividad seria. Por lo que se refiere a Kant, en la *Kritik der Urteilskraft* define el juego, siguiendo a Aristóteles, como *una ocupación placentera por sí misma, por lo que no necesita de otra finalidad* en contraposición al trabajo que es *una ocupación penosa por sí misma y que solo atrae por el resultado que promete*, por ejemplo, la remuneración (p. 112).

Estableciendo las ocho categorías generales que se detallan en la tabla 13, Munné (1980) recopiló y agrupó las principales teorías que buscan explicar tanto la naturaleza del juego como su génesis y su sentido.

**Tabla 13**

*Principales teorías que buscan explicar la naturaleza, la génesis y el sentido del juego*

Categoría	Representantes	Descripción
Teorías de la homeostasis	F. Schiller (1759-1805), alemán H. Spencer (1820-1903), inglés J. Schaller (1810-1868), alemán M. Lazarus (1824-1903), alemán	Consideran que el juego es una liberación psicofisiológica de energía vital excedente para restablecer el equilibrio del organismo. Esta energía se puede verter también en una forma inferior consistente en los ejercicios físicos (deporte), o en otra superior que produce los sentimientos (arte).
Teoría teleológica del ejercicio preparatorio	F. Fröbel (1782-1852), alemán K. Groos (1861-1946), alemán W. McDougall (1871-1938), inglés	La actividad lúdica no es una simple descarga de energía, sino una auténtica preparación para la vida, que tiene su base en los instintos. Es un pre-ejercicio, un entrenamiento vital y, por ello, altamente educador.
Teoría de la recapitulación	G. S. Hall (1844-1924), estadounidense	Así como el naturalista Haeckel creía que el desarrollo de un individuo (ontogénesis) estaba condicionado por el desarrollo de su grupo ( <i>phylum</i> ) de pertenencia (filogénesis), Hall veía en el juego del niño una reproducción de costumbres ancestrales que representan anteriores etapas del por las que pasó el <i>phylum</i> de la humanidad.
Teoría de los impulsos innatos	J. Dewey (1859-1952), estadounidense	El niño nace con impulsos indefinidos y dispersos que se pueden coordinar en forma de actividades de juego a través de la interacción con otras personas, sobre todo adultos de quienes adquiere la naturaleza humana.
Teoría de la dinámica infantil	F. Buytendijk (1887-1974), neerlandés	La actividad lúdica responde a un dinamismo biológico espontáneo que va de la tensión a la relajación, y cuya esencia se encuentra en la dinámica juvenil en la que no se da la dirección ni la intención. El juego es antiteleológico: su elemento objetivo es lo imprevisible, al final como mera posibilidad; en cuanto a su objeto subjetivo reside en el placer.
Teorías de la autoexpresión	G. Mead (1863-1931), estadounidense E. Mitchell (1889-1983), estadounidense B. Mason (1898-1953), estadounidense J. Piaget (1896-1980), suizo	El niño al jugar representa amplios roles sociales que le permiten dar sus propias respuestas a situaciones creadas por él mismo, autoexpresándose. El juego es una actividad autoformadora de la personalidad del niño, mediante una asimilación de lo que el mundo ofrece al yo.
Teorías psicoanalíticas	S. Freud (1856-1939), austriaco É. Claparède (1873-1940), suizo S. Slavson (1890-1981), estadounidense E. Erikson (1902-1994), estadounidense A. Sabora (1912-2004), estadounidense E. Mitchell (1889-1983), estadounidense	La actividad lúdica en los adultos es una proyección de impulsos sociales no aceptados. Todo el simbolismo del juego opera de catarsis de los impulsos y deseos censurados moral o socialmente por el superyó y reprimidos por el yo. La actividad lúdica, libre y espontánea, es ante todo un medio de descargar la agresividad mediante una regresión, aceptable personal y socialmente, hacia modos del comportamiento infantil.
Teorías de carácter antropológico social y cultural	J. G. Frazer (1854-1941), escocés B. Malinowski (1884-1942), polaco L. Frobenius (1873-1938), alemán J. Huizinga (1872-1945), neerlandés R. Caillois (1913-1978), francés	Frente a las teorías individualistas o psicológicas, los antropólogos han subrayado la dimensión sociocultural del juego. El juego es creador de la cultura; en él manifiestan los pueblos su interpretación del mundo. Ya en sus formas más simples está dotado de significación y en las superiores tiende a la figuración, a representar simbólicamente la realidad.

*Nota:* Basado en *Psicosociología del tiempo libre: un enfoque crítico* por F. Munné, 1980, pp. 112-118. Copyright 1980 por Trillas



Este autor reconoce que las interpretaciones hechas a partir de una única teoría no alcanzan para explicar el carácter complejo del juego:

La existencia de tan distintas concepciones sobre el juego explica y se explica por la importancia y el valor que tiene esta manifestación de la conducta en campos tan diversos como la psicología y la psiquiatría, la pedagogía y el arte, la religión y la política . . . Incluso han podido hacerse aportaciones sobre algunos aspectos del fenómeno en campos tan inesperados como la economía . . . y la lingüística . . . Este carácter complejo hace pensar que cualquier interpretación monista es reductora del fenómeno, insuficiente para explicar sus variadísimas manifestaciones (p. 116-117).

De hecho, no ha mencionado dentro de sus grupos de teorías a algunos importantes intelectuales que se dedicaron al estudio de la naturaleza del juego y que fueron contemporáneos de otros que sí ha incluido. Entre ellos, se destacan los ingleses Donald Winnicott (1896-1971) y Gregory Bateson (1904-1980), los soviéticos Lev Vygotski (1896-1934) —bielorruso— y Daniel Elkonin (1904-1984) —ucraniano—, el canadiense Herbert Marshall McLuhan (1911-1980), el estadounidense Jerome Bruner (1915-2016) y el neozelandés Brian Sutton-Smith (1924-2015), por citar solo a algunos.

Es necesario, por lo tanto, considerar los diversos patrones históricos de interpretación del juego como visiones parciales que se superponen y se complementan entre sí, para así lograr una explicación pluralista de este fenómeno.

## 4.2. El estado de *flujo*

En su obra *Flow: The Psychology of Optimal Experience*, el psicólogo estadounidense —de origen húngaro— Mihály Csíkszentmihályi (1990) se hace las siguientes preguntas:

¿Por qué jugar o practicar deporte nos hace disfrutar, mientras que las cosas del día a día — como trabajar o estar en casa— suelen ser tan aburridas? ¿Y por qué una persona disfrutará incluso aunque esté en un campo de concentración, mientras que otra consigue aburrirse estando de vacaciones en un lugar fantástico? (pp. 71-72).

De acuerdo con este autor, el componente crucial del disfrute es la experiencia de *flujo* (Csíkszentmihályi, 1975, p. 11), que es la sensación holística que las personas sienten cuando se involucran completamente con una actividad (p. 36). Asimismo, explica a qué se debe el nombre de la experiencia:

En este peculiar estado dinámico que es el estado de flujo, la acción sigue a la acción de acuerdo con una lógica interna que parece no necesitar ninguna intervención consciente del actor, quien experimenta como un fluir unificado de un momento a otro, en el que él tiene el control de sus acciones, y en el que hay poca distinción entre el yo y el entorno, entre el estímulo y la respuesta, o entre el pasado, el presente y el futuro (p. 36).

La respuesta a las preguntas anteriores sería entonces, según Csíkszentmihályi (1975), que las personas disfrutan las actividades con las que experimentan el flujo:

Uno de los principales rasgos de las experiencias de flujo es que generalmente son, en mayor o menor medida, *autotélicas*, es decir, las personas buscan el flujo principalmente por sí mismo, no por las recompensas extrínsecas incidentales que pudieran derivarse de él. Sin embargo, uno puede experimentar flujo en cualquier actividad, incluso en algunas actividades que parecen estar menos diseñadas para el disfrute: en el frente de batalla, en una cadena de montaje de fábrica o en un campo de concentración (p. 36).

Para este autor, es obvio que la experiencia de flujo se puede encontrar en el juego, aunque también aclara que se puede presentar en muchas otras actividades que involucran el uso de la creatividad:

El flujo se experimenta más fácilmente en ciertos tipos de actividades . . . Los juegos son *actividades de flujo* obvias, y jugar es la *experiencia de flujo* por excelencia. Sin embargo, jugar no garantiza que uno esté experimentando flujo, así como el hecho de recitar el *Pledge of Allegiance* no es prueba de sentimientos patrióticos. A la inversa, la experiencia de flujo también se puede encontrar en actividades que no son juegos. Una de esas actividades es la creatividad en general, incluidos el arte y la ciencia . . . Casi cualquier descripción de la experiencia creativa incluye relatos experienciales que, en aspectos importantes, son análogos a los que hacen las personas que juegan (Csíkszentmihályi, 1975, pp. 36-37).

Según Csíkszentmihályi (1990), la clave de las actividades de flujo —y, en particular, de los juegos— es que estas hacen más compleja la personalidad de quienes las practican:

Toda actividad de flujo que involucre la competición, el azar o cualquier otra dimensión de experiencia . . . ofrece una sensación de descubrimiento, un sentimiento creativo que transporta a la persona a una nueva realidad. Empuja a la persona hasta los niveles más altos de rendimiento y la conduce a estados de conciencia que no ha experimentado antes. En suma, transforma la personalidad haciéndola más compleja. En este crecimiento de la personalidad está la clave de las actividades de flujo (p. 74).

En la figura 14 se muestra cómo una persona que empieza a jugar, con poca habilidad y desafíos bajos, experimenta el flujo (A1). Si continúa jugando sin aumentar el desafío, su habilidad aumentará, pero —más tarde o más temprano— dejará de disfrutar, dado que el juego le parecerá aburrido (A2). También podría ocurrir que el juego le imponga desafíos demasiado altos para sus habilidades, lo cual le generará ansiedad y frustración (A3). En ambos casos, el deseo de disfrutar lo hará volver a entrar al canal de flujo (A4), ya sea buscando nuevos y mayores desafíos —en el caso de estar aburrido— o practicando para aumentar sus habilidades —en el caso de estar ansioso o frustrado. Así, Csíkszentmihályi (1990) concluye que este aspecto dinámico explica por qué las actividades de flujo conducen al crecimiento y al descubrimiento. Uno no puede disfrutar haciendo la misma cosa al mismo nivel durante mucho tiempo: se sentirá o aburrido o frustrado; y entonces el deseo de disfrutar lo estimulará nuevamente para que ponga a prueba sus habilidades o para que descubra nuevas oportunidades de usarlas (p. 75).

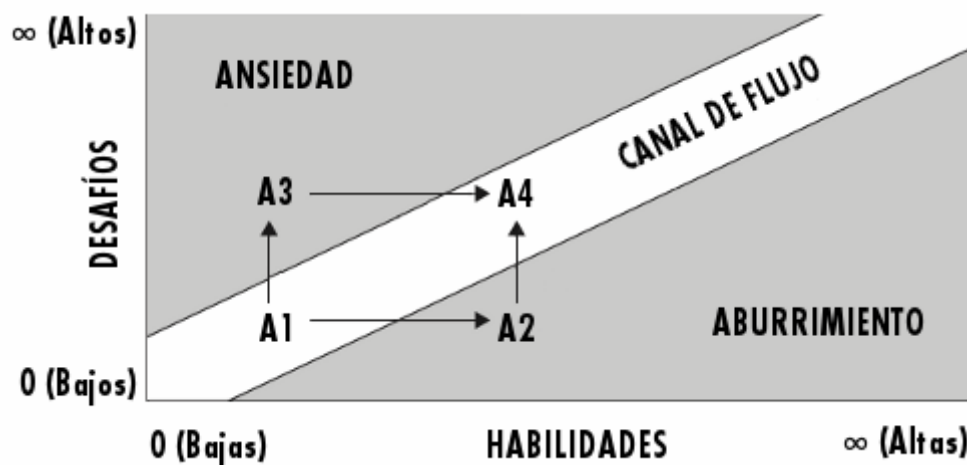


Figura 14. El canal de flujo. Adaptado de *Flow: The Psychology of Optimal Experience* por M. Csíkszentmihályi, 1990, p. 74. New York: Harper & Row. Copyright 1990 por Harper & Row

### 4.3. Los videojuegos

Desde hace varias décadas, los ingresos globales de la industria de los videojuegos no han parado de crecer y, durante los primeros años del siglo XXI, su facturación anual en todo el mundo finalmente superó a la de otras industrias del entretenimiento, como la

discográfica y la cinematográfica, acercándose al negocio de los deportes —que obtiene ingresos por los derechos de transmisión, el patrocinio, el *merchandising* y la venta de entradas— («New Gaming Boom», 2017).

Tal crecimiento sostenido de la industria de los videojuegos solo fue posible gracias a la generalizada adopción que estos tuvieron en prácticamente todo el mundo. Nichols (2013) señala que los principales medios de América del Norte, Europa y Japón publicaron artículos sobre ellos, ya no las tradicionales discusiones sobre la funcionalidad del *hardware* y las cuestiones de los efectos de los videojuegos en los niños, sino discusiones sobre los videojuegos como parte de la vida cotidiana (p. 19).

A continuación se define el término *videojuego*, se detalla la evolución de los videojuegos, se describen los géneros en que pueden ser agrupados y se los encuadra como una forma de arte popular.

#### 4.3.1. Definición del término *videojuego*

En la introducción de la monumental *Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming*, su editor Mark J. Wolf (2012) explica, por oposición a otros términos similares, cómo se usa en esa obra el término *videojuegos*:

El término *videojuegos* (video games) se usa aquí en un sentido muy amplio e inclusivo y es el más amplio de los términos que se refieren a los juegos discutidos aquí. El término *juegos de computadora* (computer games) es un poco más restrictivo; está reservado para juegos que usan computadoras hogareñas (o computadoras centrales) pero no los que se encuentran en sistemas de consola dedicados, aunque pueda decirse que estos últimos sistemas también contienen una computadora (en general, las máquinas que permiten jugar *juegos de computadora* son las que permiten realizar otras funciones de cómputo además de los juegos). Los términos *juegos electrónicos* (electronic games) y *juegos digitales* (digital games), que son incluso más amplios que *videojuegos*, no se utilizan porque abarcan muchos juegos que no utilizan un dispositivo de imágenes y que pueden tener poca o ninguna visualización. Un problema similar surge con el término *juegos portátiles* (handheld games) que parece indicar solo la naturaleza portátil del *hardware* utilizado; los juegos portátiles considerados aquí son solo aquellos que también calificarían como videojuegos debido al uso de alguna tecnología de imágenes. Además, el término se usa típicamente para dispositivos para los cuales el propósito principal es el juego, mientras que los juegos en otros dispositivos portátiles, como los juegos para teléfonos celulares, se suelen denominar *juegos móviles* (mobile games) (pp. xv-xvi).

En el *Diccionario Usual* de la Real Academia Española es posible encontrar dos acepciones del término *videojuego*: “1. m. Juego electrónico que se visualiza en una pantalla” y “2. m. Dispositivo electrónico que permite, mediante mandos apropiados, simular juegos en las pantallas de un televisor o de una computadora” («Real Academia Española. Diccionario Usual», 2018).

En este trabajo, el término *videojuego* se emplea con la primera acepción, ya que la segunda, además de ser una mera referencia al dispositivo —es decir, a la plataforma de *hardware*—, actualmente está un poco incompleta, ya que hoy en día existen muchos otros dispositivos con pantallas que permiten simular juegos —entre los cuales se destacan los teléfonos celulares y las tabletas.

Wolf y Perron (2003) intentan definir el término *videojuego* dividiéndolo en dos partes y explicando cada una por separado:

La primera parte del término —*video*— parece requerir que la acción del juego aparezca de forma visual en una pantalla (aunque originariamente *video* se refería a los tubos de rayos catódicos —CRT, de *Cathode Ray Tubes*— que se utilizaban en los juegos domésticos y *arcades*, ahora los juegos portátiles con visualización basada en píxeles también suelen llamarse videojuegos). La segunda parte del término —*juego*— es más difícil de definir (p. 14).

Por ejemplo, en su libro *Homo ludens* —publicado originalmente en 1938—, Huizinga (1972) definió el concepto de *juego* de la siguiente manera:

El juego, en su aspecto formal, es una acción libre ejecutada *como si* y sentida como situada fuera de la vida corriente, pero que, a pesar de todo, puede absorber por completo al jugador, sin que haya en ella ningún interés material ni se obtenga en ella provecho alguno, que se ejecuta dentro de un determinado tiempo y un determinado espacio, que se desarrolla en un orden sometido a reglas y que da origen a asociaciones que tienden a rodearse de misterio o a disfrazarse para destacarse del mundo habitual (p. 27).

En 1958, Caillois (1986) criticó esta definición por ser a la vez demasiado amplia y demasiado limitada, a tal punto que excluye los juegos de azar:

La parte . . . que considera al juego como una acción desprovista de todo interés material excluye simplemente las apuestas y los juegos de azar, es decir, por ejemplo, los garitos, los casinos, las pistas de carreras y las loterías que, para bien o para mal, ocupan precisamente un lugar importante en la economía y en la vida cotidiana de los diferentes pueblos (pp. 29-30).

Por suerte, en el mismo libro, Huizinga también proporcionó la siguiente definición que, según Caillois (1986, p. 28) es “menos rica pero también menos limitativa”:

El juego es una acción u ocupación libre, que se desarrolla dentro de unos límites temporales y espaciales determinados, según reglas absolutamente obligatorias, aunque libremente aceptadas, acción que tiene su fin en sí misma y va acompañada de un sentimiento de tensión y alegría y de la conciencia de *ser de otro modo* que en la vida corriente (Huizinga, 1972, p. 45).

Para el propio Caillois, el juego se puede definir formalmente como una actividad libre, circunscrita en límites espacio-temporales precisos y determinados por anticipado, incierta, improductiva, reglamentada y ficticia (1986, pp. 37-38). En el juego predomina siempre alguno de los siguientes principios: la competición (*agôn*), el azar (*alea*), el simulacro (*mimicry*) o el vértigo (*ilinx*) (p. 41).

En su libro *Rules of Play: Game Design Fundamentals*, Salen y Zimmerman (2003) comparan las definiciones empleadas en ocho obras publicadas entre 1938 y 2002 y concluyen que un *juego* es un sistema en el cual los jugadores se involucran en un conflicto artificial, definido por reglas, que termina en un resultado cuantificable (p. 80).

Basándose en el trabajo de Salen y Zimmerman, también Juul (2003) comparó diversas definiciones publicadas entre 1938 y 2003, y llegó a la siguiente definición:

Un *juego* es un sistema formal basado en reglas con un resultado variable y cuantificable, donde a los diferentes resultados se les asignan diferentes valores, el jugador ejerce un esfuerzo para influir en el resultado, el jugador se siente afectado por el resultado y las consecuencias de la actividad son opcionales y negociables.

Como se puede observar, esta última definición permite que un mismo juego pueda ser jugado con o sin consecuencias, con lo cual se soluciona la limitación señalada por Caillois en la definición de Huizinga, acerca de la exclusión de los juegos de apuestas y de azar.

Para definir el término *videojuego*, bastaría con ampliar las definiciones anteriores incorporándoles el aspecto tecnológico (dado que, como ya se señaló, los videojuegos son juegos *electrónicos* visualizados en una *pantalla*). Sin embargo, al considerar algunos aspectos más, varios autores han propuesto otras definiciones sumamente interesantes.

Frasca (2001) define el *videojuego* incluyendo la posibilidad de jugar remotamente (en línea):

El término *videojuego* abarca cualquier forma de *software* de entretenimiento basado en una computadora, ya sea textual o con imágenes, que se sirve de cualquier plataforma electrónica, como las computadoras personales o las consolas, y que involucra a uno o a varios jugadores en un entorno físico o en línea (p. 4).

Para Esnaola Horacek (2004), los videojuegos son *narraciones*. Destacando las posibilidades de participación y de inmersión que ofrecen, esta autora formula la siguiente definición:

Los videojuegos son artefactos culturales que expresan, en términos vigotskianos, los relatos de nuestra cultura en clave de hipermedia electrónica. Son narraciones que utilizan las herramientas que la técnica ha desarrollado a través de los tradicionales relatos cinematográficos, pero con un sello propio caracterizado por el despliegue de las posibilidades de participación y de inmersión que permite esta tecnología (p. 240).

Revuelta Domínguez y Guerra Antequera (2012) resaltan el empleo de múltiples tipos de lenguajes en los videojuegos y también la proyección de la personalidad de los jugadores cuando deciden cómo jugarlos:

El videojuego se podría definir como un hiperlenguaje dinámico-proyectivo, es decir, un instrumento que incluye diversos tipos de lenguaje distintos, como son el visual, el sonoro, el literario, gestual... todos ellos encuadrados en un mundo cambiante y dúctil a elección del creador del mismo y de los usuarios, de ahí la parte proyectiva, pues el usuario/a es quien verdaderamente encamina el juego como quiere jugarlo, proyecta su propia personalidad o una personalidad totalmente distinta con matices vinculados con su yo real o totalmente inversos, creando un alter ego irreconocible en el mundo real pero posible en el mundo virtual (p. 2).

Esnaola Horacek (2014) define nuevamente los videojuegos, destacando esta vez el rol que tienen en el desarrollo de la generación *gamer*:

Los hemos definido como un hipergénero artístico emergente con impacto en el desarrollo cognitivo, emocional, kinestésico, que está interviniendo en la construcción de subjetividad en la generación *gamer*. Esta definición que parecería demasiado complicada no es más que definir a los videojuegos en general como un producto cultural acorde a nuestra época de convergencias y de mezclas, donde la imagen digital avanza sobre el privilegio que ha tenido en otra época la palabra escrita (p. 129).

### 4.3.2. El origen de los videojuegos

Los videojuegos han tenido una evolución meteórica desde comienzos de la década de 1970 hasta la actualidad. Sin embargo, esta industria multimillonaria ya se había estado gestando desde varias décadas antes, por lo que —a continuación— se repasan algunos hitos que llevaron a su surgimiento.

Según Kent (2001), el punto de partida de la industria actual de las computadoras y los videojuegos se sitúa en la década de 1930, durante la Gran Depresión, con la creación del juego *Baffle Ball*:

Si un evento allanó el camino para la industria actual de las computadoras y los videojuegos, fue la creación del juego *Baffle Ball* de David Gottlieb en 1931 . . . *Baffle Ball* no usaba electricidad y se parecía poco a los juegos modernos de *pinball* o petacos . . . tenía una sola parte móvil: el émbolo. Los jugadores usaban el émbolo para lanzar bolas en un plano con una pendiente de 7 grados y tachonado con clavijas (*pins*) que rodeaban ocho agujeros que tenían asociados puntajes. Por un centavo, los jugadores podían lanzar siete bolas. *Baffle Ball* no tenía paletas (*flippers*), defensas, ni un dispositivo de puntuación. Los jugadores hacían el seguimiento de los puntajes en sus cabezas. Una vez que lanzaban la bola, solo podían controlar su curso empujando todo el gabinete, una técnica conocida más tarde como *tilting* (inclinarse el gabinete) (pp. 2-3).

A pesar de su impresionante éxito —Gottlieb prácticamente se había convertido en el *Henry Ford* de los *pinballs* (Kent, 2001, p. 3)—, poco tiempo después, en muchas ciudades, estos petacos primitivos y rudimentarios fueron considerados juegos de azar y su explotación en lugares públicos —tiendas, estaciones de autobuses, bares, etc.— fue prohibida (p. 5).

Recién en 1947, con el agregado de seis paletas (*flippers*) accionadas mediante resortes, Gottlieb pudo probar que *Humpty Dumpty*, su más reciente gabinete de *pinball*, no era solo un juego de azar porque los jugadores anotaban la mayor parte de sus puntos al empujar la pelota de regreso al juego con las paletas en lugar de confiar en la suerte y la gravedad (Kent, 2001, p. 6).

Desde entonces, los *pinballs* —que en muchas partes pasaron a ser conocidos simplemente como *flippers*— tuvieron un impacto determinante en la industria del entretenimiento:



La introducción de los *flippers* no solo cambió el paisaje básico de los juegos en sí, sino que realmente cambió la forma en que los jugadores interactuaban con los juegos. Era una forma de entretenimiento totalmente diferente de lo que había sido. Más importante aun, fue un cambio notable para los diseñadores y desarrolladores de juegos. Lo que había sido la manera prescrita para el desarrollo de juegos en la década anterior tuvo que ser alterado de forma dramática. Ya no era una situación en que una persona interactuaba pasivamente con el juego; ahora el jugador tenía una verdadera influencia y un mayor control (Kent, 2001, pp. 6-7).

Además de petacos, la industria del entretenimiento también producía ya en la década de 1940 muchos otros tipos de juegos electromecánicos, como máquinas para medir la fuerza, para jugar al *baseball* y simulaciones de carreras de caballos, de cacerías y de duelos en el lejano oeste (Kent, 2001, pp. 9-10).

En 1947, Thomas Goldsmith Jr. y Estle Ray Mann construyeron y patentaron el *Cathode Ray Tube Amusement Device*, el primer juego electrónico basado en tecnología de TV, del cual, sin embargo, no llegaron a producir más ejemplares (Dillon, 2011, p. xiv).

En 1950, para dar a conocer un uso concreto de un tubo sumador binario desarrollado por Josef Kates, la empresa Rogers Majestic presentó en la CNE (*Canadian National Exhibition*) en Toronto un juego de tres en raya llamado *Bertie the Brain*, posiblemente el primer juego de computadora del mundo (Vardalas, 2001, p. 33).

En 1951, la empresa británica Ferranti presentó en el *Festival of Britain* en Londres la computadora *Nimrod*, una máquina especialmente construida con 480 tubos de vacío contra la cual el público podía jugar al Nim (Borchers, 2001).

En 1952, Christopher Strachey programó un juego de damas (*draughts*) en la Ferranti Mark 1 de la *University of Manchester* y Alexander Douglas implementó un juego de tres en raya (*OXO*) en la EDSAC (*Electronic Delay Storage Automatic Calculator*) de la *University of Cambridge* (Ivory, 2016, p. 3).

En 1958, William Higinbotham —un científico que había trabajado en el equipo de desarrollo de la primera bomba atómica— decidió crear con tecnología de computación analógica y dos osciloscopios de 5 pulgadas un juego interactivo para entretener a los participantes del día anual de visitas públicas del *Brookhaven National Laboratory*. El resultado, *Tennis for Two*, fue popular entre los visitantes, pero solo apareció una vez más en el próximo día anual de visitas (Ivory, 2016, p. 2).

Ivory (2016) considera limitado el impacto que estos primeros juegos electrónicos tuvieron en la evolución de los videojuegos:

*Tennis for Two* y sus predecesores nunca fueron ampliamente jugados o lanzados comercialmente; fueron producidos solo como prototipos de trabajo o exhibidos al público en eventos aislados. El primer videojuego que encontró una gran audiencia y estuvo disponible más allá de una sola exhibición fue *Spacewar!* (p. 3)

*Spacewar!* fue programado por Steve Russell durante 1962 en la DEC PDP-1 (*Programmable Data Processor-1*) del MIT. Anteriormente —cuando era estudiante en el *Dartmouth College*—, junto con su profesor John McCarthy (uno de los pioneros de la inteligencia artificial), Russell había trabajado en la implementación de un intérprete para el lenguaje LISP (Kent, 2001, p. 18). No obstante, a pesar de que Russell tenía ciertos conocimientos de inteligencia artificial, *Spacewar!* era un juego para dos jugadores, pues —en su opinión— no había suficiente potencia de cómputo disponible como para hacer un oponente decente (p. 19). De acuerdo con Ivory (2016), *Spacewar!* fue un verdadero éxito:

*Spacewar!* permitía que dos jugadores se batieran a duelo disparándose torpedos el uno al otro desde naves espaciales que orbitaban un agujero negro. *Spacewar!* era jugado en una pantalla de tubo de rayos catódicos mediante controladores personalizados . . . y presentaba la puntuación en la pantalla, una característica no disponible en un osciloscopio . . . Esta y otras características orientadas a la competencia garantizaron que *Spacewar!* fuera un éxito. Un año después de su presentación . . ., copias y variaciones del programa comenzaron a surgir en los laboratorios de investigación en todos los Estados Unidos, y el juego era jugado no solo en las PDP-1, sino también en otras computadoras que usaban un tubo de rayos catódicos . . . *Spacewar!* logró dos hitos importantes para la escalabilidad del videojuego como medio masivo: fue el primer videojuego que se jugó en más de una máquina y el primer videojuego adaptado para su comercialización (pp. 3-4).

En efecto, el ingeniero Nolan Bushnell —quien en su juventud había trabajado en un parque de diversiones y conocía bien el negocio de los juegos electromecánicos (Kent, 2001, p. 29)— desarrolló, junto con Ted Dabney, una versión de *Spacewar!* para un solo jugador. La empresa Nutting Associates acordó producir esta máquina recreativa que funcionaba con monedas y que denominaron *Computer Space*, permitiendo que retuvieran los derechos de propiedad intelectual y pagándoles una regalía por cada venta realizada

(Dillon, 2011, p. 10). A pesar del diseño futurista de su gabinete de fibra de vidrio, con bordes redondeados y colores vibrantes, *Computer Space* no fue un éxito. Después del lanzamiento en 1971 en los Estados Unidos, Nutting no logró vender la totalidad de las 1500 máquinas originales, ni tampoco construyó más (Kent, 2001, pp. 33-34).

En 1972, la empresa estadounidense Magnavox —que fabricaba y comercializaba televisores, entre otros productos electrónicos— lanzó *Odyssey*, la primera consola hogareña de videojuegos de la historia. Había sido desarrollada desde 1966 por un equipo dirigido por el ingeniero Ralph Baer en la empresa contratista de defensa Sanders Associates, la cual —debido a su naturaleza militar— no podía volcarse al negocio de los juguetes (Kent, 2001, pp. 24-25). A pesar de que al jugador actual pueda parecerle un dispositivo sumamente primitivo, el lanzamiento de la *Odyssey* fue, según Dillon(2011), un hecho verdaderamente revolucionario:

La *Odyssey* era una máquina analógica capaz de mostrar una línea y tres pequeños cuadrados blancos en la pantalla: no tenía capacidades de sonido . . . El juego predeterminado, llamado *Table Tennis*, mostraba una línea que dividía la pantalla en dos partes iguales. Los jugadores manejaban un cuadrado cada vez moviéndolo arriba/abajo e izquierda/derecha con el controlador. El objetivo del juego era hacer rebotar el tercer cuadrado de ida y vuelta el uno al otro. No había visualización de resultados en pantalla. Para hacer más interesantes y significativos los otros juegos, que se iniciaban insertando cartuchos específicos, se colocaban plantillas especiales sobre la pantalla, lo cual permitía visualizar colores, tableros, caminos y objetivos adicionales. La *Odyssey* fue un dispositivo verdaderamente revolucionario . . . como una herramienta para llevar con éxito a la pantalla de TV varios juegos de tablero y de mesa, con la experiencia mejorada también mediante el uso dados, cartas , circuitos, etc. (pp. 12-13).

También en 1972, Bushnell y Dabney fundaron Atari, cuyo primer lanzamiento —*Pong*, una máquina recreativa que funcionaba con monedas, diseñada por Al Alcorn— fue, a la vez, el primer éxito comercial de la empresa:

Unas dos semanas después de la instalación de la máquina en el bar Andy Capps, el propietario llamó a Al porque el juego había dejado de funcionar. Cuando el ingeniero llegó, con gran sorpresa se dio cuenta de que las monedas se desbordaban de la pequeña caja y provocaban la falla. El juego era tan popular que los clientes incluso comenzaban a hacer cola frente al bar antes de la hora de apertura y, una vez adentro, se ponían a jugar a *Pong* sin pedir ningún trago, dejando al dueño del bar completamente perplejo (Dillon, 2011, p. 15).

Puede afirmarse, por lo tanto, que 1972 fue el año del despegue de los videojuegos como una industria. Desde entonces, una enorme cantidad de títulos han sido lanzados para las consolas hogareñas —descendientes de la *Odyssey*—, para las máquinas recreativas que funcionan con monedas —descendientes de *Pong*— y para las muchas otras plataformas —computadoras personales, teléfonos celulares, consolas portátiles— que surgieron a lo largo de los años posteriores.

### 4.3.3. Géneros de videojuegos

Actualmente, los videojuegos son muy diversos, por lo cual es muy útil establecer taxonomías que permitan clasificarlos en distintos géneros, definidos según criterios pertinentes para su estudio. Así, Wolf (2001) descarta la utilización de la iconografía y el tema como criterios para establecer géneros para los videojuegos, y considera más apropiado basarse en la interactividad:

Si bien la *iconografía* y el *tema* pueden ser herramientas apropiadas para analizar películas de Hollywood y muchos videojuegos, otra área, la *interactividad*, es una parte esencial de la estructura de todos los juegos y una forma más apropiada de examinar y definir los géneros de los videojuegos (p. 114).

Este autor explica que, al elaborar una lista de géneros, primero es necesario analizar las interacciones requeridas para cumplir el objetivo principal de cada videojuego:

El objetivo del juego es una fuerza de motivación para el jugador y, al combinarlo con las diversas formas de interactividad presentes en el juego, tenemos un buen punto de partida para construir un conjunto de géneros de videojuegos. El objetivo del juego puede ser múltiple o dividirse, ubicando el juego en más de un género . . . Por ejemplo, el objetivo principal en *Pac-Man* (1980) . . . es comer los puntos amarillos. Para hacerlo con éxito, el personaje-jugador debe evitar a los fantasmas que lo persiguen y también navegar un laberinto. Por lo tanto, mientras que *Pac-Man* puede clasificarse principalmente como un juego de *Recolección*, también podemos clasificarlo —aunque de forma secundaria— como un juego de *Escape* o de *Laberinto*. Comenzando con las interacciones requeridas por el objetivo principal de cada juego, podemos dividir la gran variedad de videojuegos en una serie de géneros (pp. 115-116).

De este modo, tomando la interactividad de los juegos como único criterio para elaborar una taxonomía, Wolf llegó a los 42 géneros que se muestran en la tabla 14.

Tabla 14

*Géneros de videojuegos según su interactividad*

Género	Ejemplos
1. Abstracto ( <i>Abstract</i> )	Arkanoid; Breakout; Q*bert
2. Adaptación ( <i>Adaptation</i> )	Jeopardy; The Price is Right; Virtual Pool
3. Aventuras ( <i>Adventure</i> )	E.T. The Extraterrestrial; Myst; Ultima
4. Vida artificial ( <i>Artificial Life</i> )	AquaZone; Creatures; The Little Computer People
5. Juegos de tablero ( <i>Board Games</i> )	Backgammon; Video Checkers; Video Chess
6. Captura ( <i>Capturing</i> )	Gopher; Hole Hunter; Keystone Kapers
7. Juegos de cartas ( <i>Card Games</i> )	1000 Miles; Eric's Ultimate Solitaire; Montana
8. De atajar ( <i>Catching</i> )	Big Bird's Egg Catch; Lost Luggage; Stampede
9. Persecución ( <i>Chase</i> )	Chase HQ
10. Recolección ( <i>Collecting</i> )	Boulder Dash; Pac-Man; Prop Cycle
11. Combate ( <i>Combat</i> )	Battletech; Battlezone; Combat
12. Demostración ( <i>Demo</i> )	Dealer Demo (Bally Astrocade); Music Box Demo (Coleco ADAM)
13. Diagnóstico ( <i>Diagnostic</i> )	Diagnostic Cartridge FDS100144 del Atari 5200; Final Test Cartridge (Coleco ADAM)
14. De esquivar ( <i>Dodging</i> )	Dodge 'Em; Freeway; Frogger
15. Conducción ( <i>Driving</i> )	Night Driver
16. Educativo ( <i>Educational</i> )	Mario's Early Years: Fun With Numbers; Mario Teaches Typing
17. Escape ( <i>Escape</i> )	Pac-Man; Maze Craze; Mousetrap
18. Lucha ( <i>Fighting</i> )	Avengers; Body Slam; Mortal Kombat
19. Vuelo ( <i>Flying</i> )	Descent; Solaris; Starmaster
20. Juego de apuestas ( <i>Gambling</i> )	Casino; Slot Machine; Video Poker
21. Película interactiva ( <i>Interactive Movie</i> )	Dragon's Lair; Space Ace; Gadget
22. Simulación de gestión ( <i>Management Simulation</i> )	Sid Meier's Civilization; M.U.L.E.; SimCity
23. Laberinto ( <i>Maze</i> )	Pac-Man; Maze Craze; Dig Dug
24. Recorrido de obstáculos ( <i>Obstacle Course</i> )	Boot Camp; Pitfall!; Jungle Hunt
25. Juegos de lápiz y papel ( <i>Pencil and Paper Games</i> )	3-D Tic-Tac-Toe; Effacer; Hangman from the 25th Century
26. Petaco ( <i>Pinball</i> )	Extreme Pinball; Flipper Game; Pinball Fantasies
27. Plataforma ( <i>Platform</i> )	Donkey Kong; Lode Runner; Super Mario Bros.
28. Juegos de programación ( <i>Programming Games</i> )	AI Fleet Commander; RARS (Robot Auto Racing Simulator); Robot Battle
29. Rompecabezas ( <i>Puzzle</i> )	Jigsaw; Sokoban; Tetris
30. Preguntas y respuestas ( <i>Quiz</i> )	\$25,000 Pyramid; Fax; Name That Tune
31. Carreras ( <i>Racing</i> )	Daytona U.S.A.; Mario Kart 64; Pole Position
32. Juegos de rol ( <i>Role-Playing</i> )	Diablo; Ultima; Dungeons & Dragons
33. Ritmo y Baile ( <i>Rhythm and Dance</i> )	Beatmania; Dance Dance Revolution; Guitar Freaks
34. Dispara a todos ( <i>Shoot 'Em Up</i> )	Centipede; Doom; Galaga; Space Invaders
35. Simulación ( <i>Simulation</i> )	Aerobiz; Railroad Tycoon; Spaceward Ho!
36. Deportes ( <i>Sports</i> )	American Football; Atari Baseball; Summer Games
37. Estrategia ( <i>Strategy</i> )	Ataxx; Checkers; Chess; Stellar Track
38. Juegos de mesa ( <i>Table-Top Games</i> )	Pocket Billiards!; Pong; Virtual Pool
39. Objetivo ( <i>Target</i> )	Carnival; Human Cannonball; Shooting Gallery
40. Aventura de texto ( <i>Text Adventure</i> )	The Hitchhiker's Guide to the Galaxy; Planetfall; Zork
41. Simulación de entrenamiento ( <i>Training Simulation</i> )	A-10 Attack; Comanche 3; Flight Unlimited
42. Utilidad ( <i>Utility</i> )	Basic Programming; Home Finance; Infogenius French Language Translator

*Nota:* Basado en "Genre and the Video Game" por M. J. P. Wolf, 2001, *The Medium of the Video Game* editado por M. J. P. Wolf, 2001, pp. 116-136. Copyright 2001 por University of Texas Press

Como se puede observar a simple vista, la taxonomía anterior es demasiado extensa como para tener cualquier aplicación práctica. En realidad, abarca incluso el *software* disponible en los cartuchos de ciertas consolas antiguas, además de los juegos propiamente dichos, ya que propone, por ejemplo, los géneros 13 (Diagnóstico) y 42 (Utilidad).

Aarseth, Smedstad y Sunnanå (2003) mencionan las desventajas de las taxonomías que presentan un número tan grande de géneros:

Los intentos anteriores de clasificar y tipificar los juegos a menudo adolecen de la aparente tendencia a incluir demasiadas categorías arbitrarias, incompatibles o superpuestas. Otro problema es el uso de categorías anticuadas de tecnología sin señalar este factor diacrónico (p. 48).

Estos autores proponen clasificar los juegos a través de un modelo compuesto por 13 dimensiones básicas, cada una de las cuales puede tener distintos valores, como se observa en la tabla 15. En su opinión, así es posible identificar las principales diferencias entre los juegos de forma rigurosa y analítica, con el fin de crear géneros que sean más específicos y menos *ad hoc* que los utilizados por la industria y la prensa popular (p. 48).

También sostienen que este modelo permitiría predecir videojuegos que aún no han sido creados o que podría utilizarse incluso para inventar nuevos juegos, por lo que sugieren que, si fuera necesario modificar el modelo, se lo haga manteniendo su esencia:

La tipología presentada se puede usar para clasificar cualquier juego. En consecuencia, es posible comparar dos juegos similares, y sus diferencias se pueden identificar y describir en detalle. Además, pueden predecirse o incluso construirse juegos nuevos simplemente agregando o cambiando características a lo largo de una dimensión. Así, se podría convertir un juego omnipresente, en tiempo real, multijugador (por ejemplo, *StarCraft*) en un juego omnipresente, en tiempo real, de dos equipos, con suma de experiencia. Se necesita más investigación para decidir si estas dimensiones son lo suficientemente buenas, pero el modelo no necesita ser aceptado o rechazado como un todo: cualquier dimensión puede ser modificada, y se pueden agregar nuevas dimensiones, sin destruir el principio subyacente (p. 53).

**Tabla 15**

*Una tipología multidimensional de juegos*

Categoría	Dimensiones	Valores	Ejemplo
Espacio	1. Perspectiva	Omnipresente (el jugador ve todo lo que ocurre)	Pac-Man
		Errante (el jugador se desplaza pero no ve todo lo que ocurre)	Doom
	2. Topografía	Geométrica (las posiciones del jugador son continuas)	Quake Arena
		Topológica (las posiciones del jugador son discretas)	Chess
	3. Ambiente	Dinámico (el jugador modifica el ambiente)	Lemmings
		Estático (el jugador no modifica el ambiente)	Pac-Man
Tiempo	4. Ritmo	Tiempo real (el jugador actúa todo el tiempo)	StarCraft
		Basado en turnos (el jugador actúa cuando le toca hacerlo)	Golf
	5. Representación	Mimética (dentro y fuera del juego, el tiempo transcurre igual)	Quake III Arena
		Arbitraria (el tiempo no pasa igual dentro y fuera del juego)	Age of Empires
	6. Teleología	Finita (hay una meta y el juego finaliza cuando es alcanzada)	Impossible Mission
		Infinita (no hay cómo ganar el juego)	Tetris
Cardinalidad	7. Cardinalidad	Un jugador	Tetris
		Dos jugadores	Chess
		Multijugador	Quake Arena
		Un equipo	Dungeons & Dragons
		Dos equipos	Counter-Strike
		Multiequipos	Anarchy Online
Control	8. Mutabilidad	Estática (el jugador tiene siempre los mismos poderes)	Paperboy
		Temporal (el jugador gana poderes temporales)	Pac-Man
		Permanente (el jugador gana poderes definitivos)	Baldur's Gate
	9. Grabación del estado	Sin grabación (el estado del juego no puede grabarse nunca)	Counter-Strike
		Condicional (el estado del juego puede grabarse a veces)	Grand Theft Auto III
		Ilimitada (el estado del juego puede grabarse siempre)	Baldur's Gate
10. Determinismo	Determinista (la misma acción genera siempre la misma reacción)	Tetris	
	No determinista (la misma acción no genera siempre la misma reacción)	Chess	
Reglas	11. Reglas topológicas	Tiene (algunas reglas valen solo en ciertas posiciones del ambiente)	Pro Evolution Soccer
		No tiene (todas las reglas son universales)	Pac-Man
	12. Reglas temporales	Tiene (el paso del tiempo determina si un tarea se puede cumplir o no)	Grand Theft Auto III
		No tiene (las reglas del juego son independientes del paso del tiempo)	Pac-Man
	13. Reglas por objetivos	Tiene (el progreso depende del cumplimiento de objetivos específicos)	Heroes III
		No tiene (el progreso no ocurre al cumplir objetivos específicos)	Sims 2

*Nota:* Basado en “A multi-dimensional typology of games” por E. Aarseth, S. M. Smedstad y L. Sunnanå, 2003, *What's in a game? Game taxonomies, typologies and frameworks - Level Up Games Conference*, pp. 48-53. Copyright 2003 por University of Utrecht

Según Egenfeldt-Nielsen, Smith y Tosca (2008) el enfoque anterior tiene la ventaja de categorizar *todos los juegos posibles* que puedan ser concebidos, aunque tiene el inconveniente de tener un uso práctico limitado (p. 40). Por ello, estos autores proponen mejor categorizar los juegos según los requerimientos que tengan para alcanzar el éxito:

En lugar de centrarnos en criterios como el tema o la narración, proponemos un sistema que se centra directamente en una característica importante de los juegos: los objetivos y la forma de cumplirlos. Un ejemplo que ilustra esta distinción es la comparación de dos juegos temáticos de fútbol: *FIFA 2004* y *Championship Manager 4*. En *FIFA 2004* los jugadores deben mover sus joysticks para superar al oponente. En *Championship Manager 4*, el jugador asume el rol de entrenador de fútbol y se concentra en la estrategia de alto nivel en lugar de jugar en los partidos. Por lo tanto, aunque ambos son juegos sobre fútbol, no los consideramos del mismo género (p. 41).

En la tabla 16 se puede observar su taxonomía, la cual solamente presenta cuatro géneros: *juegos de acción, juegos de aventura, juegos de estrategia y juegos orientados a procesos*.

**Tabla 16**

*Géneros de videojuegos según sus requerimientos para el éxito*

Género	Acciones típicas	Requerimientos para el éxito	Ejemplos
1. Juegos de acción	Luchar	Reflejos rápidos	Pac-Man, Half-Life 2
2. Juegos de aventura	Resolver misterios	Capacidad lógica	Maniac Mansion
3. Juegos de estrategia	Construir algo compitiendo con otros	Análisis de variables interdependientes	Dune II, Civilization
4. Juegos orientados a procesos	Explorar y/o entrenarse	Muy diversos, a veces inexistentes	SimCity, Elite

*Nota:* Basado en *Understanding video games: the essential introduction* por S. Egenfeldt-Nielsen, J. H. Smith y S. Tosca, 2008, pp. 43-44. Copyright 2008 por Routledge

La existencia de tantos y tan diferentes sistemas de géneros de videojuegos, según estos autores, se debe a que no hay una forma objetiva de medir las diferencias entre dos cosas. Sugieren, para hacer una analogía, pensar en dos libros, los cuales tendrán muchas características en común (por ejemplo, ambos tienen páginas y son portátiles), pero también muchas diferencias (por ejemplo, las portadas se ven diferentes, tienen diferentes títulos, no pesan lo mismo y no tienen el mismo contenido). Pero no hay una forma objetiva de determinar qué similitudes o diferencias son las más importantes (pp. 40-41).



Para finalizar este asunto y, simultáneamente, para mostrar una aplicación concreta de las taxonomías de géneros de los videojuegos, en la tabla 17 se presentan los sistemas de clasificación utilizados por dos empresas: Google (que en *Google Play* —su tienda en línea— ofrece muchísimas aplicaciones para el sistema operativo Android, entre ellas juegos) y CBS Interactive (que en su sitio web *Metacritic* publica reseñas de álbumes de música, videojuegos, películas, programas de televisión y libros, y en su sitio web *GameSpot* publica exclusivamente reseñas de videojuegos).

**Tabla 17**

*Géneros de videojuegos utilizados en Google Play, Metacritic y GameSpot*

Google Play	Metacritic	GameSpot		
<b>Acción</b>	<b>Acción</b>	2D	Fútbol americano	Pantalla desplazable
<b>Aventura</b>	<b>Aventura</b>	3D	Gratis	<i>Shoot-'Em-Up</i>
Sala de juegos/ <i>Arcade</i>	Juegos de pelea	4X	Juegos de apuestas	Disparos
Juegos de mesa	Disparos en primera persona	<b>Acción</b>	Golf	<b>Simulación</b>
Cartas	Vuelo	<b>Aventura</b>	Juegos de objetos ocultos	Patinaje
Casino	Juego de socialización	Sala de juegos/ <i>Arcade</i>	Hockey	Esquí
Juegos ocasionales	Juegos de plataformas	Béisbol	Caza/Pesca	Fútbol
Educativos	<b>Rompecabezas</b>	Baloncesto	Pistola de luz	<b>Deportes</b>
Música	<b>Carreras</b>	Palizas/ <i>Beat-'Em-Up</i>	Gestión	<b>Estrategia</b>
<b>Rompecabezas</b>	Estrategia en tiempo real	Billar	Coincidir y apilar	Supervivencia
<b>Carreras</b>	<b>Juegos de rol</b>	Bolos/ <i>Bowling</i>	Miscelánea	Táctica
<b>Juegos de rol</b>	<b>Simulación</b>	Boxeo	Multijugador en línea masivo	Por equipos
<b>Simulación</b>	<b>Deportes</b>	Cartas	Arena multijugador en línea	Tenis
<b>Deportes</b>	<b>Estrategia</b>	Compilación	Música/Ritmo	Basado en texto
<b>Estrategia</b>	Disparos en tercera persona	Críquet	Sobre rieles	En tercera persona
Preguntas y respuestas	Estrategia por turnos	Defensa	Juego de mundo abierto	Pista y campo
Juegos de palabras	Juegos de guerra	Conducción/ <b>Carreras</b>	Juego de socialización	Preguntas y respuestas
	Lucha deportiva	Educativos	<i>Petaco/Pinball</i>	Por turnos
		Juegos de pelea	Juegos de plataformas	Combate de vehículos
		En primera persona	<b>Rompecabezas</b>	Realidad virtual
		Bienestar físico	En tiempo real	Surf
		Pantalla fija	Exploración de mazmorras	Lucha deportiva
		Vuelo	<b>Juegos de rol</b>	

*Nota:* Elaboración propia a partir del análisis de los sitios web *Google Play*, *Metacritic* y *GameSpot* (en negrita se han marcado los géneros comunes a los tres sitios)

#### 4.3.4. Los videojuegos como arte popular

Una gran polémica acerca de los videojuegos se generó a partir de un artículo publicado en Newsweek por el crítico Jack Kroll (2000), en el que se mostraba contrario a la idea de considerarlos como una forma de arte:

¿Por qué estos magos del juego no pueden estar satisfechos con . . . sus 7 mil millones de dólares en ventas (y aumentando) . . .? ¿Por qué deben decir que lo que están haciendo es *arte*? . . . Phil Harrison, vicepresidente de investigación y desarrollo de PlayStation, prevé la llegada de *un diseñador de juegos en el futuro que pueda tener el impacto social de un gran director de cine, autor o músico* . . . Es que los juegos pueden ser divertidos y gratificantes de muchas maneras, pero no pueden transmitir la complejidad emocional que es la raíz del arte . . . Lo aterrador es la seducción de este mundo, especialmente hacia los jóvenes . . . Sentados con sus joysticks, aguardan la llegada del salvador previsto por Phil Harrison, alguien que pueda . . . crear un nuevo arte genuino a partir de esos títeres estampados en un mundo de pantallas.

Henry Jenkins —director de la carrera de *Comparative Media Studies* en el MIT— no tardó en unirse a la discusión, respondiéndole a Kroll con otra pregunta:

Incontables espectadores lloran cuando la madre de Bambi muere. Durante la Segunda Guerra Mundial, los soldados sentían verdadera lujuria por las *pin-ups* de Vargas en la revista *Esquire*. Hemos aprendido a sentir ante las criaturas de pigmento lo mismo que sentimos ante las imágenes de personas reales. ¿Por qué con los píxeles debería ser diferente? (H. Jenkins, 2000)

H. Jenkins (2000) defiende que los juegos de computadora sean considerados como arte, un arte popular, un arte emergente, un arte en gran medida no reconocido, pero arte al fin. Su argumentación se basa en la obra del crítico cultural Gilbert Seldes, quien en 1924 publicó el libro *The 7 Lively Arts*, en el que denominaba *artes animadas* a las tiras cómicas, el cine, la comedia musical, el vodevil, la radio, la música popular y las danzas populares —prácticas culturales que así, según Aumont (2001), recibieron su legitimación:

Poco después de . . . la Primera Guerra Mundial, se tuvo la impresión de que una cierta cantidad de prácticas culturales *de masa* podían ser estéticamente tan importantes como las artes establecidas. El cine había abierto la vía a esa empresa de legitimación y no habría de dejar de hacerlo . . . En esta empresa, la legitimación corría pareja a la vituperación de un *gran arte* que Seldes encontraba invadido por producciones mediocres y vulgares (su bestia negra fue Puccini), y que, al fin y al cabo, era estéticamente necesario preferir las artes populares, a las que su popularidad misma preservaba de la esterilidad (p. 277).

Seldes (1924) presenta el núcleo intelectual de su propuesta a través de una revolucionaria serie de proposiciones:

Que no hay oposición entre las grandes artes y las artes animadas. Que ambas se oponen en espíritu a las artes medias o espurias. Que las artes espurias son más fáciles de apreciar, apelan a emociones bajas y mixtas, y ponen en peligro la pureza tanto de las grandes artes como de las artes menores. Que, salvo en los periodos en que las grandes artes florecen con un vigor excepcional, es probable que las artes animadas sean los fenómenos más inteligentes de su época. Que las artes animadas tal como existen hoy en día en los Estados Unidos son entretenidas, interesantes e importantes. Que —con pocas excepciones— estas mismas artes son más interesantes para la inteligencia cultivada de los adultos que la mayoría de las cosas que pasan por arte en la sociedad culta. Que existe una *tradición gentil* sobre las artes que ha impedido cualquier apreciación justa de las artes populares, las que, por lo tanto, han perdido la oportunidad de recibir la crítica correctiva dada a las artes serias, recibiendo en cambio solo abuso. Que, por lo tanto, el intelectual pretencioso es tan responsable como cualquier otro de lo que es realmente absurdo y vulgar en las artes animadas. Que los simples practicantes y simples admiradores de las artes animadas que no están corrompidos por las artes espurias preservan un instinto seguro para lo que es artístico en los Estados Unidos (p. 349).

Los videojuegos representan un nuevo *arte animado*, uno tan apropiado para la era digital como aquellos medios anteriores lo fueron para la era industrial. Abren nuevas experiencias estéticas y transforman la pantalla de la computadora en un campo de experimentación e innovación que es ampliamente accesible. Desbordan de vida frente a la esterilidad de otras iniciativas artísticas contemporáneas. Por ello, para H. Jenkins (2000), los videojuegos son dignos herederos de las artes populares tan apreciadas por Seldes:

Los juegos han sido adoptados por un público que de otro modo no se ha sentido impresionado por gran parte de lo que se considera arte digital. Así como las artes de los salones de la década de 1920 parecían estériles frente a la vitalidad e inventiva de la cultura popular, los esfuerzos contemporáneos por crear una narrativa interactiva a través del hipertexto modernista o del arte de instalación vanguardista parecen pretenciosos y carentes de vida frente a la creatividad que los diseñadores de juegos aportan a su oficio.

Cinco años después de iniciada la polémica sobre si los videojuegos son o no una forma de arte, H. Jenkins publicó un artículo titulado *Games, the new lively art*, en el que narraba la gradual aceptación de los videojuegos como una nueva forma de arte popular:

Mientras que los críticos, como Kroll, se mostraban reacios a atribuir mérito artístico a los juegos, los artistas de otros medios parecían dispuestos a absorber aspectos de la estética del juego en su trabajo. En las escuelas secundarias y universidades de todo el país, los estudiantes discutían los juegos con la misma pasión con la que las generaciones anteriores habían debatido los méritos del *New American Cinema* o la *Nouvelle vague* francesa . . . Al mismo tiempo, los académicos finalmente estaban adoptando los juegos como un tema digno de un tratamiento serio, no simplemente como un problema social, un desafío tecnológico, un fenómeno cultural o una fuerza económica dentro de la industria del entretenimiento, sino también como una forma de arte que exigía una evaluación estética seria (H. Jenkins, 2005).

#### 4.4. Una taxonomía de los jugadores

En 1990, Richard Bartle —de *Multi-User Entertainment Ltd.*, una empresa de diseño y desarrollo de *software* comúnmente conocida como MUSE y especializada en juegos de computadora multijugador en línea— publicó el artículo *Who plays MUAs?* (*¿Quién juega aventuras multiusuario?*) en el cual clasificaba a los jugadores de este género de videojuegos en cuatro tipos: conseguidores (*achievers*), exploradores (*explorers*), socializadores (*socialisers*) y asesinos (*killers*). Posteriormente, retomó esta clasificación en el artículo *Hearts, clubs, diamonds, spades: Players who suit MUDs* (*Corazones, tréboles, diamantes, picas: jugadores que se adaptan a los juegos multiusuarios de mazmorras*) de 1996, donde a cada tipo le asoció un palo de la baraja inglesa, con la intención de crear una regla mnemotécnica para reconocerlos más fácilmente (Bartle, 1996). Según Teixes (2015), “todos tenemos características de cada uno de estos tipos de jugadores. Sin embargo, siempre hay alguna que es predominante”. De acuerdo con este autor, al desarrollar un juego, es importante conocer el perfil de los posibles jugadores para elegir los elementos de diseño correctos:

Cuando se diseña un juego . . . se debe tener muy en cuenta cuál es el perfil de los participantes para decidirse por unas dinámicas y mecánicas de juego concretas. De este modo, los *asesinos* necesitarán de dinámicas más rápidas que los *socializadores*, por ejemplo. Un juego que quiera atraer *conseguidores* deberá tener una curva de dificultad retardadora para mantener su interés en él.

Por estar muy ampliamente difundida, en la tabla 18 se muestra un resumen de esta taxonomía.

Tabla 18

*Tipos de jugadores (Taxonomía de Bartle)*

Tipo	Palo de la baraja inglesa	Justificación del palo asociado	Descripción
Conseguidores ( <i>Achievers</i> )	Diamantes	Siempre están buscando un tesoro	Gusta de actuar en el entorno del juego y lo hace tanto para conseguir los objetivos fijados como para hacerlo correctamente. Ganar no es la prioridad, pero sí hacer un buen desempeño en el juego. Siente inclinación por las recompensas, por completar tareas, por el reconocimiento, por las relaciones sociales y por las clasificaciones.
Exploradores ( <i>Explorers</i> )	Picas (en algunos países, este palo se asocia con la punta de una pala)	Cavan en busca de información	Interactúa con el mundo. Disfruta explorando el entorno del juego, buscando secretos y resolviendo enigmas. Es el más proclive a probar y a inventar nuevas maneras de jugar un juego y compartir sus descubrimientos con los demás jugadores.
Socializadores ( <i>Socialisers</i> )	Corazones	Muestran empatía hacia los otros jugadores	Interactúa con los demás y juega para poder conectar con otros. La conexión social es lo más importante para él en los juegos. Busca la colaboración y el juego colectivo
Asesinos ( <i>Killers</i> )	Tréboles (en inglés, este palo se denomina <i>clubs</i> y significa clavos, que son un tipo de mazas o garrotes)	Es con esta arma que ellos golpean a la gente	Actúa sobre los demás. Le gusta ganar y no descarta hacerlo a costa de los demás. La competición es muy importante para él y se siente confortable en el ámbito social cuando puede hacer gala de sus triunfos.

Nota: Basado en *Gamificación: fundamentos y aplicaciones* por F. Teixes, 2015. Copyright 2015 por UOC

## 4.5. La ludología

El registro más antiguo del uso del término *ludología* data de 1947, cuando el educador argentino José María Lunazzi lo empleó en su artículo *Ludología y Ludoterapia*:

Quando luego de someter a nuestros niños a severa vigilancia, de interrogarlos cuidadosamente, de ensayar diversos procedimientos y de requerir sin resultado la intervención del médico para supuestas deficiencias físicas, nerviosas, digestivas, etc., se nos ocurre (o el ojo clínico nos anuncia) que la falla está por otro lado, es cuando aparece esa nueva ciencia que tantos servicios presta ya a la recomposición de la naturaleza humana . . . Esa ciencia es la *ludología*, o sea la especialidad que investiga cómo ha jugado el hombre, cuáles son las condiciones positivas de su ser lúdrico, cómo en virtud de determinada técnica de juego puede reaccionar contra las propias contrariedades o contra las oposiciones que le ha originado el medio social. Quien procede al desarrollo de esos estudios y alcanza su dominio se constituye en *ludólogo*, y quien aplica procedimientos especiales derivados de los mismos y orienta tales procedimientos a las mejores relaciones de salud orgánica, de psiquismo individual o de relaciones sociales, actúa ludoterapéuticamente, es un *ludoterapeuta* (Lunazzi, 1947, p. 80).

Poco tiempo después, en el *XIV Congreso Internacional de Sociología* que tuvo lugar en Roma del 30 de agosto al 3 de septiembre de 1950, Per Maigaard —un profesor de educación física de Nakskov, Dinamarca— presentó una ponencia titulada *About Ludology* en la cual expresaba el deseo de impulsar esta ciencia, fundando un periódico y una asociación, y limitando su campo de actividad:

Los juegos son y siempre fueron más que pasatiempos. También son medios de educación y, sobre todo, son medios de salud para la mente y el cuerpo . . . Pero el conocimiento de los juegos se divide entre varias ciencias . . . Queremos mucho que los registros —que ahora están a cargo de archivos locales— estén accesibles para uso internacional en uno o más archivos mundiales, traducidos a idiomas internacionales. Queremos catálogos bibliográficos. Queremos un periódico para ludología y una asociación para ludólogos, a la que puedan unirse los científicos de todas las ciencias interesadas, incluidos los ludólogos que estén familiarizados con la tipología, la técnica y la táctica de los juegos. Los primeros objetivos de los ludólogos serán, sin embargo, limitar su campo de actividad y hacer, mediante una clasificación decente de los juegos, que ese campo sea estudiable (Maigaard, 1950, pp. 362-364)

Maigaard también dio la siguiente definición: la *ludología* es la ciencia de los juegos y una parte de la sociología y de las ciencias involucradas (p. 364).

Casi medio siglo más tarde, el diseñador de videojuegos, consultor y catedrático uruguayo Gonzalo Frasca publicó en 1999 en la revista literaria finlandesa *Parnasso* el artículo *Ludologia kohtaa narratologian*, también publicado en inglés con el título de *Ludology Meets Narratology: similitudes and differences between (video)games and narrative*. Según Juul (2009), el término *ludología* fue adoptado para referirse a un movimiento que apoyaba la idea de que el fenómeno interesante en la cultura digital no eran la televisión interactiva o la realidad virtual, sino los *videojuegos*, y buscaba constituir su estudio como una disciplina independiente de la teoría literaria y la narratología, negando que fuera una simple subcategoría del cine, de los estudios narrativos o de nuevos medios:

La *ludología* es el recordatorio continuo de que los videojuegos deben tomarse en serio y de que las teorías de otros campos no pueden aplicarse indiscriminadamente en los juegos. Si la *ludología* temprana era contraria a la aplicación amplia de la teoría narrativa en los juegos, la *ludología* actual debería desconfiar de la aplicación simplista de cualquier campo que intente colonizar los videojuegos. En la *ludología* hay que ser observador y sin prejuicios, tomando las cosas en serio, los jugadores y los juegos (p. 364).

Frasca explica que, a veces, también se utiliza la denominación *game studies* (estudios del juego) en lugar de *ludología*:

Me han preguntado varias veces cuál es la diferencia entre los *game studies* y la *ludología*. La respuesta, hasta donde yo lo veo, es que no hay ninguna diferencia. Ambos términos describen nuestra nueva disciplina y los uso constantemente como sinónimos (Frasca, 2003).

Planells de la Maza (2013) presenta un resumen bastante completo de la posterior evolución de esta disciplina en casi todo el mundo. En el caso de España, señala que la investigación es bastante esporádica y dispersa, concentrada en pocos centros, aunque destaca el asociacionismo de los académicos para el intercambio científico y también el de tipo social, así como el surgimiento de nuevas publicaciones académicas sobre los videojuegos:

La investigación española sobre los videojuegos como objetos comunicativos es esporádica y, por lo general, muy dispersa. Pocos centros aglutinan a los principales investigadores. En Barcelona, la *Universitat Pompeu Fabra* tiene un incipiente grupo de profesionales liderados por Óliver Pérez y el nacimiento de la *Enti (Escuela de Nuevas Tecnologías Interactivas)* augura, al menos, un plan formativo coherente. Valencia cuenta con la *Jaume I*, el equipo liderado por Gómez Tarín y el primer grado en diseño de videojuegos del ámbito público. En Madrid, investigadores como Pilar Lacasa, Salvador Gómez y Alfonso Cuadrado están consolidando el campo de los *Game Studies* junto con los estudios impulsados, con apoyo de las principales empresas del sector, por el *Centro Universitario de Tecnología y Arte Digital (U-tad)* . . . Otra manifestación esencial que demuestra la consolidación de los estudios sobre el juego parte del asociacionismo de los académicos y la creación de cauces adecuados para el intercambio científico . . . En el caso español, la *Asociación Española de Investigadores de Videojuegos (ASEIV)*, nacida en 2012, tiene por objetivos coordinar e impulsar los *Game Studies* en este país, estableciendo puentes y cauces de comunicación entre los investigadores que se encuentran dispersos en todo el territorio. Del mismo modo, España ha sido un país pionero en el impulso del asociacionismo de tipo social, como lo demuestra el trabajo del colectivo artístico *Arsgames* en el seno del Matadero y el *Medialab-Prado* en Madrid . . . Finalmente, destaca el creciente número de publicaciones . . . de corte académico dedicadas exclusivamente a los *Game Studies* . . . En España, la *Universidad de Sevilla* ha sido la pionera al crear la publicación electrónica *LifePlay* (pp. 524-526).

En la República Argentina —al igual que en otros países de la región— el acercamiento a los *Game Studies* apenas se está dando a través de la implantación de

carreras de formación en el desarrollo de videojuegos. La tabla 19 muestra las principales carreras existentes hasta 2017.

**Tabla 19**

*Carreras de formación en el desarrollo de videojuegos dictadas en la República Argentina*

Duración	Carrera	Título	Institución	Localidad
5 años	Licenciatura en Producción de Videojuegos y Entretenimiento Digital	Licenciado en Producción de Videojuegos y Entretenimiento Digital	Universidad Nacional de Rafaela	Rafaela (Pcia. de Santa Fe)
3 años	Diseño y Programación de Videojuegos	Diseñador y Programador de Simuladores Virtuales	Escuela Da Vinci	Ciudad Autónoma de Buenos Aires
3 años	Tecnicatura Superior en Desarrollo de Videojuegos	Técnico Superior en Desarrollo de Videojuegos	Image Campus	Ciudad Autónoma de Buenos Aires
3 años	Tecnicatura Superior en Arte y Animación para Videojuegos	Técnico Superior en Arte y Animación para Videojuegos	Image Campus	Ciudad Autónoma de Buenos Aires
3 años	Tecnicatura Superior en Desarrollo de Videojuegos	Técnico Superior en Desarrollo de Videojuegos	Instituto Privado de la Cámara Argentina de Comercio y Servicios	Ciudad Autónoma de Buenos Aires
3 años	Tecnicatura Universitaria en Desarrollo de Videojuegos	Técnico Universitario en Desarrollo de Videojuegos	Universidad Argentina de la Empresa	Ciudad Autónoma de Buenos Aires
3 años	Tecnicatura Superior en Desarrollo de Simulaciones Virtuales y Videojuegos	Técnico Superior en Desarrollo de Simulaciones Virtuales y Videojuegos	Colegio Universitario IES (Instituto de Estudios Superiores)	Córdoba (Pcia. de Córdoba)
3 años	Tecnicatura Universitaria en Producción y Diseño de Videojuegos	Técnico Universitario en Producción y Diseño de Videojuegos	Universidad Nacional de José C. Paz	José C. Paz (Pcia. de Buenos Aires)
3 años	Tecnicatura en Desarrollo de Video Juegos	Técnico en Desarrollo de Video Juegos	Universidad de La Punta	La Punta (Pcia. de San Luis)
2,5 años	Tecnicatura Universitaria en Desarrollo de Videojuegos	Técnico Universitario en Desarrollo de Videojuegos	Universidad Abierta Interamericana	Ciudad Autónoma de Buenos Aires
2,5 años	Tecnicatura Universitaria en Sistemas Informáticos con Orientación en Videojuegos	Técnico Universitario en Sistemas Informáticos con Orientación en Videojuegos	Universidad de Palermo	Ciudad Autónoma de Buenos Aires
2,5 años	Tecnicatura en Programación de Videojuegos	Técnico Universitario en Programación de Videojuegos	Universidad de Mendoza	Mendoza (Pcia. de Mendoza)
2,5 años	Tecnicatura en Diseño y Programación de Videojuegos	Técnico Universitario en Diseño y Programación de Videojuegos	Universidad Nacional del Litoral	Santa Fe (Pcia. de Santa Fe)

*Nota:* Elaboración propia a partir de la información obtenida de la WWW («Educación», s. f.)



Según comentan Esnaola Horacek, Iparraguirre, Averbuj y Oulton (2015), en la República Argentina también se observa un incipiente asociacionismo:

En el año 2000 un grupo de jóvenes desarrolladores argentinos fundó la Asociación de Desarrolladores de Videojuegos Argentina (ADVA). Esta asociación trabaja activamente en uno de los sectores económicos más dinámicos de América Latina —los videojuegos—, el cual emplea a jóvenes profesionales a cargo de la programación, diseño, ilustración, guión y música (p. 35).

Más recientemente, en 2016 se constituyó la Fundación Argentina de Videojuegos (FUNDAV), cuyo objetivo es el siguiente:

Promover e impulsar el desarrollo económico, social y cultural de los videojuegos, aportando valor para la generación de productos y desarrollos artísticos y/o experienciales, instrumentando el valor del videojuego como canal de comunicación, expresión, investigación y transmisión de conocimientos en toda la República Argentina («Fundación Argentina de Videojuegos», s. f.)

En la tabla 20 se muestran las líneas de trabajo dentro de las que se encuadran los proyectos y programas apoyados por la FUNDAV.

**Tabla 20**

*Líneas de trabajo de la Fundación Argentina de Videojuegos*

Línea	Descripción
Educación	Utilización de videojuegos como componente pedagógico en espacios de aprendizaje formales e informales. Investigación de las prácticas de desarrollo de videojuegos para mayor impacto formador.
Formación	Asistencia en la formación de nuevos profesionales de la industria de desarrollo de videojuegos.
Salud	Utilización del medio para promover o ayudar tratamientos de salud, de prevención y de investigación médica.
Accesibilidad	Herramientas para el desarrollo de juegos y proyectos del medio, con el objeto de pluralizar el acceso.
Diversidad	Inclusión de todo tipo de público a través de videojuegos.
Difusión	Para promover juegos y estudios de desarrollo nacionales, las actividades de la fundación y las líneas de trabajo.
Investigación, desarrollo e innovación	Búsqueda de nuevos procesos, técnicas o tecnologías para el desarrollo de videojuegos.
Expresión cultural	Los videojuegos vistos como una forma de arte. Se busca la innovación y la experimentación en temáticas y formas.
Comunidades	Actividades orientadas a formar y fortalecer una comunidad de desarrolladores de videojuegos en Argentina con varias sedes regionales. El objetivo es permitir y acrecentar la comunicación entre los desarrolladores de todo el país.
Editorial y Publicación	Orientado a la generación y publicación de contenido didáctico en español, relacionado al desarrollo de videojuegos.
Competencias	Se abocará a trabajar con actividades, materiales, conferencias, talleres o eventos de cualquier tipo donde se vea reflejado el carácter de competitividad y/o compañerismo en los videojuegos.

*Nota:* Elaboración propia a partir de la información obtenida de «Fundación Argentina de Videojuegos», s. f.

No cabe duda de que el fenómeno más interesante en la cultura digital son los videojuegos. Su estudio está lejos de ser una simple subcategoría del cine, de los estudios narrativos o de nuevos medios. Es por ello que la *ludología*, como una disciplina independiente de la teoría literaria y la narratología, ha llegado para quedarse y, en algunos años, sin duda estará firmemente establecida en la República Argentina y en el resto del mundo.

# EL APRENDIZAJE MEDIANTE LA UTILIZACIÓN DE VIDEOJUEGOS

Hasta no hace mucho tiempo atrás, según Garzón (2016), en muchas instituciones educativas el *juego* se restringía localmente al patio y temporalmente al recreo:

Durante muchos años se estigmatizó el juego en la educación, no desapareció sino que se lo dejó librado sólo al espacio de recreo, un espacio-tiempo liberador para que más tarde las energías estén centradas en lo más importante: el conocimiento. El recreo para que los cuerpos de los niños descarguen sus energías y así éstos puedan portarse bien y aprender, buenos alumnos, alumnos quietitos, alumnos tranquilos. El paradigma que se instalaba era: Educar o jugar. Entonces cuando te educas, no juegas, y también viceversa, claro, cuando juegas no aprendes (p. 11).

Con el pasar del tiempo, el juego empezó a encontrar un lugar *dentro* de las aulas, inicialmente para aprender música, como señalan DeVries y Kamii (1975):

Cuando los juegos grupales son defendidos en los textos sobre la primera infancia, los autores generalmente limitan su justificación a razones de mera liberación de energía y desarrollo físico o social, o ven tales juegos como justificados solo en el contexto de objetivos musicales (p. 1).

Más tarde, además de su importancia para la educación musical, se pasó a reconocer el valor del uso del juego para la enseñanza de muchas otras materias:

Debido a que muchos estudiantes disfrutaban al jugar, muchos educadores a menudo se han preguntado si el aspecto lúdico no podría combinarse con la instrucción para mejorar el aprendizaje, han explorado la viabilidad de utilizar un formato de juego para complementar o reemplazar la enseñanza de una variedad de temas y han discutido cómo aumentar el interés de los estudiantes con enfoques motivadores como los juegos (Randel, Morris, Wetzel y Whitehill, 1992, p. 261).

Sin embargo, Cañeque, Castro y Greco (1999) señalan que el hecho de que la importancia de los juegos finalmente fuera reconocida no implica que las instituciones los hayan adoptado inmediatamente:

Hoy, la industria, las finanzas, la investigación, están utilizando el juego para comprender situaciones complejas y tomar decisiones. Se lo vislumbra como un medio de comunicación de excelencia para los tiempos venideros. Mientras tanto, ¿qué hace la escuela? Nadie duda de su influencia preponderante en todas las sociedades, como transmisora legítima de objetos culturales, valores, costumbres, creencias. Pero del juego, seriamente, todavía no se pudo ocupar. Es más, parece que le tuviera miedo (p. 34).

Hacia fines del siglo XX, estas autoras argentinas ya defendían el aprendizaje mediante el juego, al que le atribuían las funciones que se muestran en la tabla 21.

**Tabla 21**

*Funciones esenciales del juego*

Función	Descripción
Sirve para descubrir	El juego promueve una sensación continua de exploración y descubrimiento. El sacar la cubierta es el aprendizaje más rico que se da en la persona; es como tender continuamente hacia la resolución de la situación problemática.
Sirve para relacionar a unos con otros	El juego es un factor de permanente activación y estructuración de las relaciones humanas. El individuo logra las relaciones más saludables en campos lúdicos, puesto que, en un clima de juego, el sujeto se maneja y opera con un alto grado de libertad y responsabilidad, más allá de prejuicios, estereotipos u otro tipo de ataduras sociales.
Sirve para equilibrar el cuerpo con el alma	Es una actividad que se presenta en forma natural, como un circuito autorregulable de tensiones y relajaciones. Permite recuperar el equilibrio perdido en la unidad cuerpo-alma.
Sirve para transmitir valores, bienes y productos culturales	Por lo general, gran parte de lo que sucede en la realidad, primero fue gestado en el campo de la fantasía. Los niños imitan reglas, normas, pautas, gestos. Estas conductas vienen transmitidas por los padres y pasan a través de la actividad lúdica.
Sirve para evadirse saludablemente de la realidad	El juego, de cualquier nivel que sea, es un mecanismo muy importante para alejarse de la realidad y después volver a ella fortificado. Esta evasión posibilita el pensamiento inteligente, habilitando la capacidad de centrar y descentrar atributos de dos mundos diferentes, el de la realidad y el de la fantasía.
Sirve para expulsar, expresar, sacar y destrabar	Solo el juego permite convertir lo siniestro en fantástico dentro de un clima de disfrute. Estimula la expulsión del conflicto y abre así nuevos espacios internos para conocer y comprender aquello que pudo haber quedado frenado, reprimido o paralizado.
Sirve para experimentar	El juego reduce la sensación de gravedad frente a errores y fracasos. Explorar y descubrir nuevas alternativas son los mecanismos que se aprenden y se instalan en el área del conocimiento. En el juego, el error y el acierto quedan neutralizados. Se puede volver a probar, y eso eleva enormemente el nivel intelectual de una persona.
Sirve para aprender el manejo de la libertad	El juego posibilita a la persona ser más libre, porque puede operar en un campo donde se ofrecen infinitas variables de elección. Esto le permite estructurarse y desestructurarse frente a una dificultad, recuperar la libertad interna y la conciencia de sí mismo. En un campo ilusorio, la persona puede elegir ser y hacer todo aquello que en la realidad no se anima: pedir, reclamar, defenderse, espiar, atacar, ser poderoso, admirado, etc. Si una persona sabe que puede hacer algo en el juego, queda grabado en su campo vivencial y no lo olvida jamás, se convierte en una conducta magistral.

*Nota:* Basado en “El juego es vida” por H. Cañequé, C. Castro y H. Greco, 1999, *El juego. Debates y aportes desde la didáctica*, pp. 38-41. Copyright 1999 por Ediciones Novedades Educativas

Más recientemente, Esnaola Horacek y Galli (2016) también se manifestaron a favor de la inclusión de juegos en los sistemas educativos:

La idea de incluir juegos, tanto en el sistema formal como en el informal, es . . . reconocer a la actividad lúdica como una actividad propia del ser humano, en la cual se realiza una transformación simbólica de la realidad que permite incorporar, interpretar y asimilar la realidad compleja para la puesta en marcha del desarrollo de capacidades (p. 19).

Según Frasca (2009), el interés formal de las universidades por el juego aumentó precipitadamente a principios del siglo XXI:

La irrupción económica del videojuego es sin dudas la que ha precipitado el interés formal por este aspecto clave de la humanidad . . . El videojuego se perfila como la industria cultural dominante en este nuevo siglo y las universidades, que habían apostado fuertemente por la cultura *online* hasta que la burbuja *punto com* explotó, rápidamente se lanzan a conquistar este terreno poco conocido, con la esperanza de poder utilizarlo para modernizar un poco a las Humanidades y, quizás, hasta tocar algunas de las monedas que esta poderosa industria genera (p. 38).

De acuerdo con Marcano Lárez (2008), el papel de herramientas de aprendizaje que siempre han tenido los juegos, actualmente lo desempeñan los *videojuegos*:

Los juegos han constituido una poderosa herramienta de aprendizaje de conductas y actitudes necesarias para el eficiente desempeño sociocultural. En la actual sociedad digital ese papel lo desempeñan los videojuegos. Estos proveen a los videojugadores de habilidades y destrezas propias de la época y facilitan el aprendizaje de procesos complejos con eficacia. Estos beneficios de los videojuegos se han querido aprovechar para hacer más efectivos los procesos educativos, de entrenamiento e información (p. 96).

El uso de videojuegos con fines educativos forma parte de dos corrientes distintas —el *aprendizaje basado en juegos digitales* y los *juegos serios*— las cuales son tratadas a continuación.

### **5.1. El *Aprendizaje Basado en Juegos Digitales***

El *aprendizaje basado en juegos digitales* es una de las principales corrientes en que se encuadra el uso de los videojuegos en educación. Van Eck (2006) explica los tres factores que llevaron a su popularización:

El peso combinado de tres factores ha resultado en un amplio interés público por los juegos como herramientas de aprendizaje. El primer factor es la investigación realizada por los defensores del DGBL (*Digital Game-based Learning* o aprendizaje basado en juegos digitales). En cada década desde el surgimiento de los juegos digitales, los investigadores han publicado decenas de ensayos y libros sobre el poder del DGBL . . . El segundo factor involucra a la actual *Generación de Internet*, los *nativos digitales*, quienes han perdido el contacto con la instrucción tradicional, requieren múltiples flujos de información, usan más el razonamiento inductivo, quieren interacciones frecuentes y rápidas con contenido . . . —características que se corresponden bien con el DGBL. El tercer factor es la mayor popularidad de los juegos. Los juegos digitales son una industria que factura 10 mil millones de dólares por año (p. 17).

Para profundizar el tratamiento de este tema, se definirá primeramente el propio concepto de *aprendizaje basado en juegos digitales*.

### 5.1.1. Definición del *Aprendizaje Basado en Juegos Digitales*

En su libro *Digital Game-Based Learning*, Prensky (2001) ofrece la siguiente definición:

Dicho de la manera más simple, el *aprendizaje basado en juegos digitales* es cualquier combinación de contenido educativo y juegos de computadora. La premisa detrás de esta modalidad es que es posible combinar videojuegos de computadora con una gran variedad de contenidos educativos, alcanzando resultados tan buenos o incluso mejores que aquellos que se obtienen por medio de métodos tradicionales de aprendizaje. Entonces vamos a definir el aprendizaje basado en juegos digitales como cualquier juego de aprendizaje llevado a cabo en una computadora o en línea (pp. 145-146).

Esta última definición es tan amplia que, según Prensky, además de abarcar —obviamente— los videojuegos, también incluye otras categorías de juegos digitales, por ejemplo, los juegos por correo electrónico:

¿Qué son los juegos por correo electrónico? En pocas palabras, son una variedad de *aprendizaje basado en juegos digitales* donde toda la información recibida, así como toda la información que ingresan los usuarios (como respuestas a preguntas, movimientos u otros) se comunican únicamente a través de mensajes de e-mail. Al igual que con todas las demás categorías de *aprendizaje basado en juegos digitales*, los juegos por correo electrónico vienen en una gran variedad de niveles de sofisticación y costos. Los juegos por e-mail más simples son juegos que cualquier persona que tenga un correo electrónico puede configurar y ejecutar por sí misma: una revisión después de un curso o el juego de *plantearles un enigma a los colegas* (2001, p. 323).

Posteriormente, en su libro *Don't Bother Me Mom — I'm Learning!*, Prensky (2006) explica la diferencia entre los minijuegos (juegos triviales como la mayoría de los juegos educativos en línea y los juegos de *edutainment*) y los juegos complejos, que exigen un aprendizaje mucho mayor —además de la capacidad para investigar y la posibilidad de comunicarse con otros participantes— para poder ser jugados con éxito:

Cuando los padres y los maestros de hoy escuchan la palabra *juego*, por lo general piensan en algo *trivial* . . . Los *juegos educativos en línea* (incluyendo los cientos que se encuentran en sitios web como el de la NASA, UNICEF, National Geographic y muchos otros) y los *juegos de edutainment* (combinación de *educación* y *entretenimiento* en inglés) que pueden comprarse en algunas tiendas, como Carmen Sandiego, Reader Rabbit y Math Blaster, son también muy similares a los juegos de su infancia debido a que no se necesita mucho tiempo para poder jugarlos . . . Los juegos que acabo de describir son lo que yo llamo *minijuegos* y, con raras excepciones, son triviales. Pero aquí está el problema: estos representan solo una fracción de los juegos disponibles. El resto . . . son algo completamente nuevo, se trata de los *juegos complejos*. A diferencia de los minijuegos, para dominar los juegos complejos es necesario invertir un gran número de horas de atención y concentración. Exigen el aprendizaje de múltiples habilidades, así como la capacidad para investigar y comunicarse fuera del juego . . . Cuando los niños hablan . . . acerca de los juegos, casi siempre se refieren a los juegos complejos . . . A los niños les encantan, y . . . claman tenerlos para su aprendizaje escolar (p. 56).

Aunque no los denominó *juegos complejos*, J. P. Gee (2003) también se refirió a estos en su libro *What video games have to teach us about learning and literacy*, donde señala la importancia que tienen los buenos principios de aprendizaje presentes en este nuevo tipo de videojuegos, para poder venderse bien:

Aquí tenemos algo que es largo, difícil y desafiante. Sin embargo, no se puede jugar un juego sin aprenderlo. Si nadie juega un juego, no se vende y la compañía que lo fabrica va a la quiebra. Por supuesto, los diseñadores podrían hacer los juegos más cortos y simples para facilitar el aprendizaje. Eso es a menudo lo que hacen las escuelas. Pero no, en este caso, los diseñadores de juegos siguen haciendo los juegos cada vez más largos y desafiantes . . . y aun así logran que los jugadores los aprendan. ¿Cómo? . . . Si un juego, por cualquier razón, tiene buenos principios de aprendizaje incorporados en su diseño, entonces . . . puede vender muchas copias . . . Si un juego tiene principios de aprendizaje deficientes incorporados en su diseño, entonces . . . no se venderá bien . . . Al final, entonces, los videojuegos representan un proceso darwiniano . . . que conduce a mejores diseños para un buen aprendizaje, . . . un buen aprendizaje de cosas difíciles y desafiantes (pp. 5-6).

Asimismo, Squire (2003) considera que los *juegos complejos* son más consistentes con el paradigma educativo emergente que los tradicionales *minijuegos* (aunque, así como J. P. Gee, tampoco se refiere a ellos con estas denominaciones):

Históricamente, las computadoras han sido utilizadas en la educación sobre todo como herramientas de apoyo al ejercicio y la práctica orientados a recordar hechos . . . Los juegos de ejercicio y práctica (*drill and practice*) como Alga-Blaster, Reader Rabbit o Knowledge Munchers se han vuelto populares porque pueden integrarse fácilmente en un currículo didáctico tradicional como *ejercicios de enriquecimiento* durante el tiempo de estudio independiente. Los buenos juegos de ejercicio y práctica utilizan el *género de acción* de los videojuegos para que los estudiantes se involucren . . . Aunque los juegos de ejercicio y práctica pueden tener un papel importante en entornos de aprendizaje centrados en el estudiante como el aprendizaje basado en problemas, . . . usar videojuegos para apoyar la exploración de micromundos o como herramienta de construcción . . . es más consistente con el paradigma educativo emergente (p. 54).

### 5.1.2. Razones para la eficacia del *Aprendizaje Basado en Juegos Digitales*

A través de una encuesta, Revuelta Domínguez y Guerra Antequera (2012) obtuvieron un listado de aprendizajes y procesos favorecedores del aprendizaje que tienen lugar —según los 115 jugadores que participaron de su estudio— al utilizar videojuegos. Estos aprendizajes y procesos favorecedores del aprendizaje se muestran en la tabla 22 y pueden ser vistos como razones para la eficacia del aprendizaje basado en juegos digitales. De acuerdo con sus autores, el estudio fue innovador por haber sido realizado desde el punto de vista de los videojugadores:

La perspectiva de la adquisición de aprendizajes desde el punto de vista del videojugador es una de las aportaciones innovadoras de la investigación, pues hasta ahora sólo se ha reflexionado desde las ideas preconcebidas de los investigadores y las opiniones vertidas en los medios (Revuelta Domínguez y Guerra Antequera, 2012, p. 1).

Según Prensky, el aprendizaje basado en juegos digitales funciona principalmente por tres razones. La primera es el *involucramiento* agregado que resulta de colocar el aprendizaje en un contexto de juego. La segunda es el *proceso interactivo de aprendizaje* empleado. La tercera es la *forma en que los dos se integran* en el paquete completo (2001, p. 147).



Como puede observarse, hay cierta coincidencia entre los autores anteriores, ya que en el primer caso se menciona la *motivación* y en el segundo el *involucramiento*, en el primero se habla de *feedback* y en el segundo de *proceso interactivo*, etc.

Tabla 22

*Razones para la eficacia del Aprendizaje Basado en Juegos Digitales desde el punto de vista de los videojugadores*

Razón	Explicación
Los videojuegos motivan a los alumnos y favorecen su rendimiento	Los videojuegos provocan una sensación motivadora en el jugador que le empuja a seguir interesado en jugar, gracias a ello podemos conseguir aprendizajes significativos e incrementar el rendimiento.
Los videojuegos permiten aprovechar el binomio lúdico-educativo	Tratando de aprovechar el impulso de la herramienta como lúdica podemos incidir en los elementos de corte didáctico y en sus posibles aplicaciones en la educación formal.
Los videojuegos permiten adquirir habilidades y/o destrezas para la resolución de problemas	La simulación de situaciones y el aprendizaje de habilidades y/o destrezas a través de los videojuegos comerciales supone un aprendizaje no formal aplicable a situaciones de la vida diaria y por tanto de utilidad para los alumnos como nuevos ciudadanos.
Los videojuegos fomentan la socialización y la cooperación	Los videojuegos que implementan contenido multijugador, ya sea cooperativo o competitivo, fomentan la comunicación entre los videojugadores y la coordinación de acciones grupales. El trabajo en grupo es una competencia requerida por las cualificaciones profesionales y el mercado laboral.
Los videojuegos estimulan el aumento de la concentración	Al dotar de múltiples estímulos al jugador, los videojuegos hacen que éste entre en un estado de gran concentración. Trabajar con ese estado para la vida diaria y las tareas escolares sería de gran ayuda.
Los videojuegos ayudan a desarrollar la autonomía personal	La resolución de problemas, el planteamiento de situaciones y la toma de decisiones crean una sensación de autonomía y automotivación que beneficia notablemente al jugador.
Los videojuegos aproximan a maestros y alumnos	Los videojuegos como nexo de unión entre docentes y discentes es un planteamiento novedoso e innovador pues el alumno lleva a su terreno al maestro y no al revés. Se crea un vínculo mayor.
Los videojuegos permiten trabajar contenidos desde una óptica <i>multitarea</i> y <i>multiárea</i>	Trabajar con videojuegos nos permitiría tratar contenidos desde ópticas infinitas, es tener contenidos en vivo y en directo controlados por los propios alumnos, se crearía un aprendizaje multidisciplinar y significativo. Además de su compatibilidad con todas las áreas curriculares del sistema educativo.
Los videojuegos ofrecen una mayor capacidad de interacción que otros medios	La capacidad del alumnado de interactuar con el contenido es limitada hoy en día, no se puede hablar con el libro, en cambio con un videojuego podemos aprender de personajes virtuales inspirados en personajes reales, es la capacidad de tocar, asimilar y vivir el aprendizaje.
Los videojuegos facilitan la asimilación e interconexión de contenidos	Asimilación de contenidos mucho más fácil y productiva, se viven situaciones, se aprende de esas situaciones. Y los contenidos aprendidos se interconectan con otras áreas pues la inmediatez del videojuego permite también la inclusión de material de otras áreas.
Los videojuegos permiten el desarrollo de valores	En el videojuego también podemos enseñar los valores básicos de nuestra sociedad. En los <i>godgames</i> , el jugador toma el poder de un dios omnipresente y omnipotente que dirige las vidas de su creación.
Los videojuegos ofrecen la simulación de situaciones	Los videojuegos permiten vivir el aprendizaje, por lo que podemos enseñar en situaciones reales, como por ejemplo: comprar en un supermercado, localizar algún edificio en una ciudad, etc.
Los videojuegos pueden ayudar a mejorar la toma de decisiones	Al simular situaciones, recaen en los hombros del jugador las consecuencias de sus decisiones. Por ello los videojuegos pueden enseñar qué decisiones son las más adecuadas a cada contexto.
Los videojuegos ofrecen <i>feedbacks</i> inmediatos	Los videojuegos también aportan algo importantísimo para la educación, y eso son los <i>feedbacks</i> , es decir la recompensa o castigo por las conductas llevadas a cabo.

*Nota:* Basado en “¿Qué aprendo con videojuegos? Una perspectiva de metaaprendizaje del videojugador” por F. I. Revuelta Domínguez y J. Guerra Antequera, 2012, RED. Revista de Educación a Distancia, 33. Copyright 2012 por RED. Departamento de Psicología Evolutiva y de la Educación de la Universidad de Murcia

### 5.1.3. Características de los buenos juegos digitales

En el número de julio de 1997, la revista *Next Generation* explicó, en un informe especial, una serie de elementos que siempre se encuentran presentes en los buenos juegos. Es interesante destacar que esta revista ya consideraba, por aquel entonces —es decir, incluso antes de entrar al siglo XXI—, que los videojuegos son una forma de arte:

Los videojuegos, como los videos musicales y los efectos de películas digitales, son un arte moderno . . . Al igual que con cualquier arte, abundan los buenos y malos ejemplos . . . Al considerar los juegos . . . como una forma de arte, es importante reconocer que sus elementos son más fáciles de detectar que de ejecutar, y no estamos sugiriendo que simplemente saber lo que se necesita para hacer un gran juego garantice un buen diseño . . . Conocer y hacer son dos habilidades separadas e igualmente vitales. La inspiración es, obviamente, otro asunto a considerar . . . El buen diseño del juego tiene como meta lograr una cierta armonía mediante el uso de todos los elementos disponibles, y es necesario ser un artista para crear este tipo de armonía («What makes a game good?», 1997, p. 40).

Los elementos señalados en el informe recién mencionado se dividían en dos grupos, según se refirieran a juegos para un único jugador (tabla 23) o para más de un jugador (tabla 24).

**Tabla 23**

*Elementos que siempre se encuentran presentes en los buenos videojuegos para un único jugador*

Elemento	Explicación
Equilibrio	El equilibrio deja al jugador con la sensación de que el juego es desafiante pero justo, y ni demasiado difícil ni demasiado fácil en ningún momento.
Originalidad	Todo buen diseño es creativo, no sigue fórmulas rígidas. Los buenos juegos no son simplemente clones de otros juegos, sino que agregan algo original.
Foco	El foco es descubrir qué es lo divertido de <i>tu</i> juego y darle al jugador la mayor cantidad posible de eso sin distracciones.
Carácter	Todo buen diseño tiene carácter. Es la profundidad y la riqueza del juego. Tanto el carácter como los personajes de un juego, si están completamente desarrollados, son lo que lo hacen memorable.
Tensión	Todo buen diseño tiene tensión. Cada buen juego lo hace a su manera. La forma clásica es hacer que el jugador se preocupe por la meta del juego y luego hacer que esta sea difícil de lograr.
Energía	Esto proviene de cosas como movimiento, impulso y ritmo. La energía del juego es lo que te mantiene jugando toda la noche o te rejuvenece después de un día duro.

*Nota:* Basado en “What makes a game good?”, 1997, *Next Generation*, 31, pp. 40-49. Copyright 1997 por Imagine Media, Inc.

**Tabla 24**

*Elementos que siempre se encuentran presentes en los buenos videojuegos para más de un jugador*

Elemento	Explicación
Velocidad	Los buenos juegos para más de un jugador son rápidos. No necesariamente en la acción del juego, sino en el manejo del escenario multijugador: las actualizaciones enviadas por los clientes y por el servidor a través de la red son eficientes.
Interacción avanzada	Los buenos juegos multijugador ofrecen variadas herramientas inventivas para derrotar a los oponentes humanos y demostrarles que uno es mejor que ellos: con frecuencia, incluso ofrecen recursos para humillar a un enemigo antes de matarlo.
Facilidad de acceso	En los buenos juegos multijugador, los jugadores no necesitan empezar o terminar su participación al mismo tiempo: los jugadores se pueden incorporar en cualquier momento sin ningún problema.

*Nota:* Basado en “What makes a game good?”, 1997, *Next Generation*, 31, pp. 40-49. Copyright 1997 por Imagine Media, Inc.

En su artículo *What makes a game good?*, publicado por primera vez en 2000, Kramer (2015) afirma que el valor percibido de un juego depende en gran medida de las preferencias individuales de quienes lo juegan:

¡Los juegos son una cuestión de gusto! . . . Algunos jugadores prefieren juegos de suerte; otros prefieren juegos de táctica; otros disfrutan al comunicarse con otros jugadores. Después están aquellos a quienes les gustan los juegos basados en la reacción, las habilidades manuales o la memoria, etc. Pero si un juego es considerado bueno o si se lo considera de poco atractivo, no depende por completo de las preferencias personales. También hay criterios objetivos que deben considerarse (p. 84).

En la tabla 25 se explican las características que, de acuerdo con este autor, deben poseer los buenos juegos: originalidad, frescura y rejugabilidad, sorpresa, igualdad de oportunidades, probabilidad de ganar, ausencia del efecto Kingmaker, ausencia de la eliminación temprana, tiempos de espera razonables, control creativo, uniformidad, calidad de los componentes, reglas consistentes, tensión, facilidad de aprendizaje y complejidad equilibrada.

Cabe mencionar que, a diferencia del informe homónimo publicado tres años antes en la revista *Next Generation*, este artículo no se refiere específicamente a los videojuegos. Sin embargo, es interesante observar que hay muchas características de los buenos videojuegos señaladas en el informe de *Next Generation* que también son mencionadas por este autor como características de los buenos juegos en general.

Tabla 25

*Características de un buen juego*

Elemento	Explicación
Originalidad	Todo nuevo juego debe poseer elementos novedosos (o conocidos, pero combinados de forma novedosa).
Frescura y rejugabilidad	El rumbo que tome el juego debe ser tan diferente como sea posible cada vez que se lo juegue.
Sorpresa	Un juego debe ser rico en sorpresas. La repetición en secuencia, progreso y eventos debe ser evitada.
Igualdad de oportunidades	Al comienzo del juego, todos los jugadores deben tener las mismas probabilidades de ganar.
Probabilidad de ganar	Hasta el final, cada jugador debe tener al menos una probabilidad —aunque sea infinitesimal— de ganar.
Sin efecto <i>Kingmaker</i>	Un jugador que ya no tiene ninguna esperanza de ganar no debe poder determinar al ganador.
Sin eliminación temprana	Todos los jugadores deben poder participar en el juego hasta que este casi haya terminado.
Tiempos de espera razonables	Deben evitarse los largos períodos de inactividad mientras los jugadores esperan su turno.
Control Creativo	Los jugadores deben poder influir en su propio progreso y dirección. Un buen juego debe ser un desafío.
Uniformidad	El título, el tema, el formato y los gráficos de un juego deben dar una impresión unificada.
Calidad de los componentes	La durabilidad, la funcionalidad y el atractivo visual del material contribuyen al valor percibido de un juego.
Reglas consistentes	Las reglas deben ser consistentes con el tipo de juego. De un juego de estrategia no debe participar el azar.
Tensión	Cada juego tiene su propia curva de tensión única. Se deben evitar periodos largos de tensión baja.
Facilidad de aprendizaje	El juego debe poder aprenderse fácilmente. Cuanto más claras y simples sean las reglas, mejor.
Complejidad equilibrada	Las reglas complejas son aceptables solo si la influencia de los jugadores en el rumbo del juego es alta.

*Nota:* Basado en “What makes a game good?”, por W. Kramer, 2000, *Game & Puzzle Design*, 1(2), pp. 84-86. Copyright 2015 por Queensland University of Technology

## 5.2. Los Juegos Serios

Los *juegos serios* (o *serious games*) son otra de las corrientes en que se encuadra el uso de los videojuegos en educación. Aunque parezca una contradicción debido a que está formado por dos palabras que —en general— son mutuamente excluyentes, el término *juegos serios* no ha dejado de ganar popularidad desde que Clark C. Abt lo usó como título de su libro *Serious Games* de 1970. A diferencia de los juegos analógicos señalados por Abt, la mayoría de los juegos serios de la actualidad son digitales: se trata principalmente de videojuegos. En 2002, el *Woodrow Wilson International Center for Scholars* creó la *Serious Games Initiative* con el objetivo de promover el desarrollo de juegos para involucrar al público general en la política («About the Serious Games Initiative», 2017). Otros grupos le siguieron, como por ejemplo *Games for Change*, que desde 2004 capacita a creadores de juegos e innovadores sociales para que ayuden a las personas a aprender, a mejorar sus

comunidades y a contribuir a hacer del mundo un lugar mejor, mediante el uso de juegos y tecnología («About Us | Games For Change», s. f.). Como una rama destacada dentro de la industria de los videojuegos, los juegos serios tienen un futuro promisorio.

### 5.2.1. Definición de *Juegos Serios*

Existen diversas definiciones de este término. La formulada por Abt en 1970 es anterior al surgimiento de la industria de los videojuegos:

Los juegos se pueden jugar en serio o informalmente. Nos ocupamos de los *juegos serios* en el sentido de que tienen un propósito educativo explícito y cuidadosamente pensado y no están destinados a ser jugados principalmente por diversión. Esto no significa que los juegos serios no deban ser entretenidos (Abt, 1987, p. 9).

La definición de Zyda (2005) es más actual y, quizá por ello, demasiado restrictiva, ya que solamente abarca a los videojuegos:

Un juego serio es una competición mental, jugada con una computadora de acuerdo con reglas específicas, que usa el entretenimiento para promover la capacitación corporativa o gubernamental, la educación, la salud, la política pública y objetivos de comunicación estratégicos (p. 26).

Michael y Chen (2006) se alinean con Abt cuando definen que un *juego serio* es un juego en el que la educación (en sus diversas formas) es el objetivo principal, más que el entretenimiento (p. 17).

En cambio, Loh, Sheng e Ifenthaler (2015) se oponen a definir el término de una manera tan amplia, pues en su opinión solo pueden ser juegos serios *reales* algunos de los juegos digitales que surgieron después de 2002. Así expresan su cuestionamiento:

Entonces, ¿cuáles son las diferencias entre los *juegos serios* y el *aprendizaje basado en juegos digitales*? Las definiciones demasiado amplias solo complicarán la situación y dificultarán la diferenciación entre los juegos serios *reales* (posteriores a 2002), los primeros *juegos educativos* (como *The Oregon Trail*) y el fallido *edutainment* (p. 7).

Según estos autores, los *juegos serios* reales deben incluir entre sus características la capacidad de llevar a cabo una evaluación apropiada (p. 8), no solo de los conocimientos adquiridos, sino de las diversas competencias que permiten desarrollar:

Los aprendices/jugadores adquieren habilidades y procesos que no se enseñan fácilmente en las aulas, incluidos el pensamiento estratégico y analítico, la resolución de problemas, la planificación y la ejecución, la toma de decisiones y la adaptación al cambio rápido . . . Es obvio que la fortaleza de los *juegos serios* radica en la mejora de las habilidades, del rendimiento y de los procesos de toma de decisiones que promueven, más que en la propagación de mensajes o la difusión de información (p. 9).

Con ellos está de acuerdo Morales Moras, quien no solo afirma que los *juegos serios* son videojuegos, sino que también menciona su uso para desarrollar competencias:

Entendemos por *serious games* un tipo de videojuego que combina su cualidad lúdica con una agenda educativa en un sentido amplio. Esto incluye sobre todo juegos dedicados a promover aprendizajes propios de los currículos académicos, objetivos pedagógicos no reglados, competencias del ámbito profesional y también los que intentan educar de una manera transversal y crear conciencia sobre temáticas de relevancia social, ya sean en el marco de las políticas públicas o las iniciativas ciudadanas (Morales Moras, 2015).

### 5.2.2. Una taxonomía de los *Juegos Serios*

Revisando los sistemas de clasificación preexistentes y analizando sus limitaciones, Djaouti, Alvarez y Jessel desarrollaron un sistema de clasificación general para los *juegos serios*, al que denominaron Modelo G/P/S y que, según ellos, tiene como meta principal servir como una herramienta para guiar a quienes deseen adentrarse en este campo:

La intención de esta clasificación es guiar a las personas a través del vasto campo de los *Serious Games* brindándoles una visión general. Por ejemplo, puede ayudar a los maestros a encontrar juegos que tengan un fuerte potencial educativo, aunque no pertenezcan al campo de los *edugames* (Djaouti, Alvarez y Jessel, 2011, p. 118).

G/P/S permite clasificar *juegos serios* muy rápidamente, marcando cruces en la representación del modelo que se muestra en la figura 15.

En el modelo G/P/S, el *propósito* y el *alcance* de un juego serio están divididos en categorías intuitivamente fáciles de entender. Por el contrario, la *jugabilidad* está dividida en los dos tipos (*paidia* —es decir, el juego libre— y *ludus* —es decir, el juego con meta—) que fueron propuestos en 1958 por Roger Caillois:

Los distintos juegos . . . se pueden situar entre dos polos opuestos. Casi por completo, en uno de los extremos reina un principio común de diversión, de turbulencia, de libre improvisación y de despreocupada plenitud, mediante la cual se manifiesta cierta fantasía desbocada que podemos designar mediante el nombre de *paidia*. En el extremo opuesto, esa exuberancia traviesa y espontánea casi es absorbida o, en todo caso, disciplinada por una tendencia complementaria, opuesta por algunos conceptos, pero no por todos, de su naturaleza anárquica y caprichosa: una necesidad creciente de plegarla a convencionalismos arbitrarios, imperativos y molestos a propósito, de contrariarla cada vez más usando ante ella tretas indefinidamente cada vez más estorbosas, con el fin de hacerle más difícil llegar al resultado deseado. Éste sigue siendo perfectamente inútil, aunque exija una suma cada vez mayor de esfuerzos, de paciencia, de habilidad o de ingenio. A este segundo componente lo llamo *ludus* (Caillois, 1986, p. 41).



Figura 15. Una representación imprimible del modelo G/P/S. Adaptado de “Classifying Serious Games: the G/P/S model” por D. Djaouti, J. Alvarez y J.-P. Jessel, 2011, *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches* editado por P. Felicia, p. 130. Hershey: IGI Global. Copyright 2011 por IGI Global

### 5.3. Tipos de juegos utilizados para aprender programación

La utilización de juegos para la enseñanza y el aprendizaje de la programación cuenta con antecedentes muy diversos. Desde los juegos de mesa no digitales, pasando por los juegos híbridos —con componentes digitales y no digitales—, hasta los juegos digitales —en especial los videojuegos—, todo tipo de juegos ha sido probado para intentar facilitar el aprendizaje de esta disciplina. A continuación se detallan algunos ejemplos destacados.

#### 5.3.1. Juegos no digitales

Alex Baker, Emily Navarro y André van der Hoek —de la *University of California, Irvine*— desarrollaron *Problems and Programmers*, un juego de cartas educativo que simula el proceso de ingeniería de *software* desde la especificación de requisitos hasta la entrega del producto. Según sus autores, este juego proporciona a los estudiantes una experiencia práctica global y de alto nivel del proceso de ingeniería de *software* de una manera lo suficientemente rápida como para ser utilizada de forma viable en un tiempo limitado, es decir, un cuatrimestre o un semestre (Baker, Navarro y Van Der Hoek, 2005, p. 3).

Si bien el juego *Problems and Programmers* no está orientado a enseñar exclusivamente programación sino ingeniería de *software*, debe señalarse que la programación forma parte de la ingeniería de *software*, y el producto que se entrega al final del proceso incluye uno o más programas.

En cambio, Igor Kholodov —un programador de Massachusetts— pensó en facilitar el aprendizaje de los conceptos básicos de lenguajes de programación como Java o C++ cuando creó *C-jump*, un juego de mesa que convierte a los jugadores en esquiadores que deben competir para ver quién baja una montaña de la manera más rápida posible. Con cada tirada del dado, los jugadores deben seguir instrucciones que son similares a los códigos de un programa de computadora. Aunque el juego de aproximadamente 30 minutos es educativo, su autor afirma que es divertido para toda la familia una vez que se entienden las reglas (Cohn, 2005).

#### 5.3.2. Juegos con componentes digitales y no digitales

En 2014, Evolve, Inc. —una empresa fundada por Nader Hamda en California—, lanzó comercialmente su primer robot: *Ozobot*, que tenía el tamaño de una pelota de golf



y se podía programar mediante líneas hechas con marcadores de colores. El éxito de este robot en las escuelas primarias de todo el mundo fue tan grande, que la empresa decidió usar la denominación de *Ozobot* para sí misma y rebautizó el robot con el nombre de *Bit*. Actualmente, la empresa también comercializa otro robot —*Evo*— y se presenta así en su sitio web:

Fabricamos robots programables para la próxima generación de creadores. Nuestra misión es preparar a los niños para un futuro en el que colaborarán con robots todos los días. Nuestros robots unen el juego físico con la tecnología digital y se pueden codificar en el editor *OzoBlockly* o con colores («About Us | Ozobot», s. f.).

### 5.3.3. Minijuegos orientados al aprendizaje de conceptos específicos

Sahar Shabanah y Jim X. Chen —de la *George Mason University*— desarrollaron AVuSG (*Algorithm Visualization using Serious Games*), un enfoque para el aprendizaje y la visualización de algoritmos que utiliza juegos serios de computadora para enseñar algoritmos. AVuSG visualiza cada algoritmo que debe aprenderse en cuatro formas: un texto, un diagrama de flujo, una demostración de juego y un juego. Se han desarrollado juegos para diversos algoritmos, tales como *Binary Search* (búsqueda binaria), *Bubble Sort* (ordenamiento por burbujeo), *Insertion Sort* (ordenamiento por inserción), *Selection Sort* (ordenamiento por selección), *Linked List* (lista enlazada) y *Binary Search Tree* (árbol binario de búsqueda) (Shabanah, 2011, p. 1036). Según sus autores, AVuSG se beneficia del deseo de los jugadores de competir y ganar, y del entretenimiento resultante de jugar, para motivar a los estudiantes a aprender algoritmos. Además, facilita la evaluación de los estudiantes utilizando los criterios para ganar o perder que tienen los juegos de computadora, prescindiendo de preguntas externas (Shabanah y Chen, 2009, pp. 34-40).

Ricardo Botero Tabares —del *Tecnológico de Antioquía*, Colombia— y Helmuth Trefftz Gómez —de la *Universidad EAFIT*, Colombia— crearon *CoquitoDobleO* (Coquito Orientado a Objetos), un “*software* concebido para apoyar la comprensión de conceptos básicos en programación orientada a objetos: clase, objeto, sobrecarga de métodos, herencia y polimorfismo” (Botero Tabares y Trefftz Gómez, 2011, p. 81). Según estos autores, el objetivo de este juego es reforzar los conceptos vistos en las clases magistrales:

El juego CoquitoDobleO . . . se apoya en el Aprendizaje Basado en Problemas dado que, en cada sub-juego, se proponen una serie de problemas que el usuario-jugador resuelve de una forma lúdica. El aprendizaje significativo se da cuando un determinado tópico u objeto de conocimiento, por ejemplo el polimorfismo, adquiere mayor significado para el estudiante luego de interactuar la clase magistral previa y con el juego CoquitoDobleO (p. 83).

Pablo Vera, Edgardo Moreno y Rocío Rodríguez —de la *Universidad Nacional de La Matanza*, Argentina— desarrollaron minijuegos para reforzar los conceptos de programación vistos en la materia *Elementos de Programación* que se dicta en todas las carreras de ingeniería de esa institución:

Estos mini-juegos se encuentran clasificados por unidad y cuentan con distintos niveles de dificultad. Por cada pregunta contestada, el alumno recibe una devolución indicando si es correcto o no lo realizado y, además, en caso de ser incorrecto, se le ofrece una pequeña explicación . . . Los mini-juegos son accesibles desde una plataforma web . . . mediante la cual se podrá ingresar a los distintos juegos y desafíos, realizar el seguimiento de los alumnos y permitir la interacción . . . Las características deseables para esta plataforma son: que los alumnos se logueen en la plataforma para poder realizar un seguimiento de los mismos, guardar los mejores récords por persona y por curso, permitir que un alumno desafíe a otros por medio de un mini-juego, etc. (Vera, Moreno y Rodríguez, 2015).

#### **5.3.4. Videojuegos con personajes programables**

A diferencia de los minijuegos mencionados en el punto anterior —que a veces no pasan de cuestionarios sobre programación revestidos con una estética tomada de los videojuegos—, en los siguientes casos se mezclan la interactividad, la narrativa y la programación, dando como resultado videojuegos de acción y aventuras con personajes programables, los cuales generalmente —aunque no en todos los casos— son robots.

Uno de los primeros videojuegos de este tipo fue *Robocode* (publicado en el año 2001), que ya se mencionó en el capítulo 1. Como se recordará, se trata de un simulador de batallas entre robots con forma de tanques de guerra, programables en Java.

En 2005, la empresa suiza Epsitec SA lanzó *CeeBot*. Esta serie de videojuegos 3D —continuación de su exitoso producto previo llamado *CoLoBoT* (*Colonize with Bots*)— incluye *CeeBot-Teen* —que está dirigido a un público de 10 a 15 años y se basa en la programación de pequeños robots para resolver tareas sobre un escritorio o dentro de una

casa—, *CeeBot-A* —que está orientado a un público de 15 años o más y se basa en la programación de robots de gran tamaño para resolver tareas al aire libre en diversos planetas—, *CeeBot3* —que está dirigido a un público de 10 a 15 años y se basa en la programación de un robot para dibujar o llevar a cabo animaciones, mediante un lenguaje similar a C# o Java— y *CeeBot4* —que está orientado a un público de 15 años o más y se basa en la programación de robots para que resolver los problemas más variados, como salir de un laberinto, correr carreras o jugar al fútbol, también mediante un lenguaje similar a C# o Java— («CeeBot: Have fun programming», s. f.).

*ProBot* —un videojuego desarrollado por Julián Moreno y Edgar Montaña, de la *Universidad Nacional de Colombia*— tiene dos características que lo diferencian de los anteriores: por un lado, “se enfoca en el campo específico de la lógica de programación sin hacer énfasis en ningún lenguaje específico . . . abordando el problema general del diseño de algoritmos y la programación estructurada” (J. Moreno y Montaña, 2009, p. 1) y, por otro lado, su personaje principal es un robot deportista que “debe competir contra una serie de oponentes que emplean cierta serie de movimientos. Para esto el estudiante debe programar al ProBot definiéndole a manera de algoritmo cuáles movimientos debe efectuar para contrarrestar los del oponente y ganarle” (p. 2).

Como ya se mencionó oportunamente, desde 2013 la organización Code.org ofrece en su sitio web numerosas actividades en línea orientadas al aprendizaje de la programación, muchas de las cuales son juegos dirigidos a niños. En la mayoría de estos casos, los jugadores deben programar a los personajes —algunas veces, a los protagonistas de algún videojuego muy conocido, como los pájaros de *Angry Birds*; otras veces, un robot, como el del juego *LightBot* que ya fue mencionado anteriormente; o también a los protagonistas de alguna película famosa, como *Star Wars*, *La Era de Hielo* o *Frozen*— para que resuelvan algún problema, dándoles órdenes mediante bloques o a través de algún lenguaje de programación como Python, Lua o JavaScript («Hour of Code», s. f.).

JavaScript también puede aprenderse jugando a *Hakitzu*, un videojuego comercial lanzado en 2013 por *Kuato Studios* para iOS y Android. Al igual que en su derivado *Code Warriors: Hakitzu Battles* de 2015, se trata de un videojuego épico, con música y efectos gráficos y sonoros muy sofisticados —prácticamente cinematográficos— en el que

el jugador debe comandar un robot gigante en una batalla, introduciendo —cuando es su turno— órdenes expresadas en JavaScript («Code Warriors: Hakitzu Battles», s. f.).

En el extremo opuesto de las opciones para aprender JavaScript —porque no usa recursos gráficos y porque es necesario contar con bastantes conocimientos para jugarlo— se encuentra *Untrusted: The Continuing Adventures of Dr. Eval* («Untrusted», s. f.). Este juego de aventuras diseñado por Alex Nisnevich y Greg Shufflin —quienes eran estudiantes de la *University of California, Berkeley* cuando lo desarrollaron en 2013— ganó ese año el primer premio en el Hackaton de Primavera de la CSUA (*Computer Science Undergraduate Association*) en Berkeley (Nisnevich, 2014). El objetivo del juego es ayudar a Dr. Eval —representado por un carácter @ (arroba)— a huir de una mazmorra llena de obstáculos generados por un código en JavaScript parcialmente editable por el jugador.

Más recientes (de 2015) son dos juegos en los que se aprenden y llevan a la práctica conceptos de la programación en *assembly* (lenguaje ensamblador): *Human Resource Machine* («Human Resource Machine», s. f.) —desarrollado por *Tomorrow Corporation*— y *TIS-100* («TIS-100», s. f.) —desarrollado por *Zachtronics*—. Mientras que en el primero hay personajes (el principal es un oficinista que debe ejecutar órdenes para mover objetos entre una entrada, un área de almacenamiento y una salida), el segundo es más bien la simulación de una computadora típica de la década de 1980, con la que deben resolverse problemas (*puzzles*) de programación en un lenguaje de bajo nivel, sin la participación de ningún personaje.

### **5.3.5. Videojuegos del género *mundo abierto***

Al contrario de los juegos mencionados hasta aquí, que están estructurados como una secuencia lineal de niveles diseñados para favorecer una aplicación escalonada de los conocimientos de programación adquiridos, los videojuegos del género *mundo abierto* tienen estructura de arenero (*sandbox*), para que los jugadores exploren libremente un entorno y aprendan a través del descubrimiento. En términos de Caillois (1958), puede decirse que estos videojuegos se componen más de *paidia* que de *ludus*.

Uno de los representantes más populares de este género es *Minecraft*, lanzado en 2011 por Mojang AB, un estudio de videojuegos sueco adquirido en 2014 por

Microsoft Corporation («¿Qué es Minecraft?», s. f.). El éxito de este mundo virtual 3D construido en red por miles de participantes no deja de ser paradójico, como explican Zorn, Wingrave, Charbonneau y LaViola Jr. (2013):

*Minecraft* es un fenómeno de juego que, paradójicamente, atrae la atención de los jugadores convencionales: sin trama, sin historia, sin objetivo, con un combate simplista y gráficos pixelados. Sin embargo, vendió más de 3,5 millones de copias incluso antes de su lanzamiento. Su principal mecánica es el juego creativo, pero también incorpora colaboración, exploración y aventura. En su forma más simple, consiste en bloques pixelados de un metro cúbico, pero también incluye herramientas, recolección de recursos, modo de supervivencia, servidores multijugador, agricultura, ganadería e incluso lógica booleana programable y movimientos mecánicos. El énfasis en la progresión personal y la presencia de estímulos supernormales hacen que *Minecraft* sea cautivante. También tiene una comunidad de *modding* activa, intercambiando complementos y ampliando el juego de nuevas maneras creativas (p. 352).

Precisamente, la capacidad de modificar el juego (*modding*) es lo que permite su uso como herramienta para el aprendizaje de la programación. Jonathan Peli de Halleux —RSDE (*Research Software Design Engineer*) principal en Microsoft— enseña JavaScript utilizando el editor *MakeCode for Minecraft* y describe así el entusiasmo que esta práctica despertó en sus estudiantes:

Los estudiantes que tomaban mis clases de informática después de la escuela y tenían la suerte de jugar manipulando la codificación en *Minecraft* se volvían completamente locos, en el buen sentido. La capacidad de escribir código e inmediatamente ver los resultados en *Minecraft*, como avatares que pueden saltar 100 bloques de alto, cavar a través de montañas y hacer que lluevan pollos, hacía que mis alumnos corrieran por el aula de pantalla en pantalla para ver lo que hacían sus compañeros y gritando las direcciones IP de sus servidores a través de la sala (de Halleux, 2017).

Además de JavaScript, *Minecraft* se puede usar para enseñar muchos otros lenguajes de programación convencionales, como Lua (Wilkinson, Williams y Armstrong, 2013), Java (Hunt, 2014), Smalltalk (Zabala, Pérez Cerrato, Blanco, Morán y Teragni, 2016), Python (Richardson, 2016) o Kotlin (Norman, 2018), por mencionar solo algunos.

La enorme cantidad de lenguajes de programación disponibles para *Minecraft* lo hacen prácticamente único en su tipo. Los otros representantes del género *mundo abierto* que pueden utilizarse para la enseñanza de la programación solo permiten trabajar con un

número reducido de lenguajes o utilizan lenguajes no convencionales. Por ejemplo, *Second Life* admite los lenguajes LSL —*Linden Scripting Language*— («Second Life», s. f.) y S4SL —*Scratch for Second Life*— (Pellas y Peroutseas, 2016); asimismo, para jugar a *else Heart.Break()* se debe escribir código en el lenguaje Sprak («else Heart.Break()», s. f.). En ambos casos, se trata de lenguajes específicos de esos juegos. En cambio, *AntMe!* soporta C# y Visual Basic («AntMe!», s. f.) —que son lenguajes convencionales— y *Dragon Architect* se programa mediante bloques gráficos (Bauer, Butler y Popović, 2017).

#### 5.4. Impacto de la utilización de juegos en el aprendizaje de la programación

En su artículo *¿Contribuye el Uso de Juegos Serios a Mejorar el Aprendizaje en el Área de la Informática?* del año 2014, Lilia García-Mundo, Juan Vargas-Enríquez, Marcela Genero y Mario Piattini —de la *Universidad de Castilla-La Mancha*— exponen los resultados de un estudio sobre juegos serios que realizaron en base a artículos de revistas, conferencias y talleres publicados entre 2009 y 2012 en *Scopus*, *ScienceDirect*, *Springer*, *Wiley InterScience*, *IEEE Digital Library* y *ACM Digital Library*. De los 1236 artículos encontrados, solo siete presentaban juegos serios usados para enseñar programación, y en la mayor parte de ellos se comprobó un impacto positivo sobre el aprendizaje:

En la mayoría de ellos [seis] se comprobó que se obtuvo una mejora en el aprendizaje . . . en el otro no se comprobó la mejora en el aprendizaje porque el artículo no evalúa un juego en particular sino que propone una metodología de evaluación del aprendizaje aplicada a cualquier juego serio . . . El mecanismo más utilizado para evaluar la mejora del aprendizaje fue cuantitativo a través de cuestionarios de conocimientos de preguntas cerradas (García-Mundo, Vargas-Enríquez, Genero y Piattini, 2014, pp. 305-308).

## EL APRENDIZAJE MEDIANTE EL DESARROLLO DE VIDEOJUEGOS

En 1995, Yasmin B. Kafai —de la *University of California, Los Angeles*— publicó el libro *Minds in Play. Computer Game Design as a Context for Children's Learning*, el cual describe y analiza un experimento bastante inusual, llamado *Game Design Project*, que tuvo lugar en una escuela primaria pública:

16 estudiantes de cuarto grado programaron juegos en Logo para enseñarles fracciones a alumnos de tercero. Los niños transformaron su aula en un estudio de diseño de juegos . . . , aprendieron a programar, redactaron historias, construyeron representaciones de fracciones, crearon diseños de paquetes . . . En este contexto, la programación se convirtió en un medio para la expresión personal y creativa de los niños (Kafai, 1995, p. xiii).

En el prólogo de ese libro, Seymour Papert —quien, como vale recordar, en la década de 1960 fue uno de los inventores del lenguaje Logo— planteó que la reflexión sobre la creación de juegos de computadora por parte de los estudiantes es un medio excelente para resaltar las cuestiones surgidas de la oposición entre el *instruccionismo* y el *construccionismo*:

Todo educador debe haber sentido algo de envidia al ver a los niños jugando videojuegos: ¡si esa energía pudiera movilizarse al servicio de aprender! . . . Esta envidia puede tomar formas diferentes. Los *instruccionistas* muestran su orientación cuando desean juegos que enseñen matemáticas, ortografía, geografía o lo que sea. La mente *construccionista* se revela cuando se imagina a los niños creando los juegos en lugar de verlos solo jugándolos. En vez de querer que los juegos instruyan a los niños, los *construccionistas* anhelan ver a los niños construir juegos (pp. xi-xii).

Para comenzar a tratar el tema del aprendizaje mediante el desarrollo de videojuegos (LMDG — *Learning-by-Making-Digital-Games*) (Earp et al., 2013, p. 440), a continuación se presenta el *construccionismo*, la teoría en la que está basada esta estrategia de aprendizaje.

## 6.1. El construccionismo

Esta teoría del aprendizaje surge inspirada en el *constructivismo* de Jean Piaget, como puede constatarse en las palabras del propio Papert (1980):

Tomo de Jean Piaget un modelo de niño como constructor de sus propias estructuras intelectuales. Los niños parecen ser aprendices innatamente bien dotados y adquieren, mucho antes de ir a la escuela, una enorme cantidad de conocimientos mediante un proceso que denomino *aprendizaje piagetiano* o *aprendizaje sin enseñanza*. Por ejemplo, los niños aprenden a hablar, aprenden la geometría intuitiva necesaria para moverse en el espacio, y aprenden lo suficiente de la lógica y la retórica para persuadir a los padres, todo esto *sin ser instruidos* (p. 7).

La diferencia principal entre el *constructivismo* de Piaget y el *construccionismo* de Papert está en el papel destacado que este último autor le atribuye a la cultura —y al entorno del aprendiz— como fuente de material para la construcción de conceptos:

Todos los constructores necesitan materiales para construir. Donde estoy en desacuerdo con Piaget es en el papel que atribuyo a las culturas circundantes como fuente de estos materiales. En algunos casos, la cultura se los proporciona en abundancia, lo que facilita el aprendizaje constructivo piagetiano. Por ejemplo, el hecho de que tantas cosas importantes (cuchillos y tenedores, madres y padres, zapatos y calcetines) aparezcan en pares es un *material* para la construcción de un sentido intuitivo del número. Pero en muchos casos en los que el desarrollo más lento de un concepto particular, según Piaget, se explicaría por su mayor complejidad o formalidad, yo veo como factor crítico el hecho de que la cultura es relativamente pobre en materiales que harían del concepto algo simple y concreto (p. 8).

Para Papert, las computadoras provistas de entornos de programación como el Logo constituyen la herramienta ideal para la construcción de modelos intelectuales:

En la mayoría de las situaciones educativas contemporáneas donde los niños entran en contacto con las computadoras, estas se utilizan para ponerlos a prueba, para proporcionarles ejercicios de un nivel apropiado de dificultad, para proporcionar retroalimentación y distribuir información. *La computadora programando al niño*. En el entorno de Logo, la relación se invierte: el niño, incluso en edad preescolar, tiene el control: *el niño programa la computadora*. Y al enseñarle a la computadora cómo pensar, los niños se embarcan en una exploración sobre cómo piensan ellos mismos . . . Esta poderosa imagen del niño como epistemólogo captó mi imaginación mientras trabajaba con Piaget. En 1964, después de 5 años en el Centro de Epistemología Genética de Piaget en Ginebra, quedé impresionado por su forma de ver a los niños como los constructores activos de sus propias estructuras intelectuales (p. 19).



Las citas anteriores fueron extraídas del libro *Mindstorms: children, computers, and powerful ideas*, publicado en 1980. Aunque el *construccionismo* no aparece mencionado ni siquiera una única vez en esta obra, sus principios ya estaban presentes allí. Más tarde, en *Constructionism: A New Opportunity for Elementary Science Education* —un proyecto presentado a la *National Science Foundation*—, Papert (1987) lo define formalmente así:

La palabra *construccionismo* es un mnemónico para dos aspectos de la teoría de la educación científica que subyace a este proyecto. De las teorías constructivistas de la psicología tomamos la visión del aprendizaje como una reconstrucción más que como una transmisión de conocimiento. Luego, ampliamos la idea de los materiales manipulativos y llegamos a la idea de que el aprendizaje es más efectivo cuando es parte de una actividad en que el alumno construye un producto significativo.

El término quedó finalmente consagrado en el libro *Constructionism* (Papert y Harel, 1991), en el cual los autores afirmaban tener pruebas estadísticamente contundentes de que la actividad construccionista —que integra las matemáticas con el arte y el diseño, y donde los niños hacen el *software*— aumenta la eficacia de la instrucción dictada por un maestro.

## 6.2. Obstáculos para el aprendizaje mediante el desarrollo de videojuegos

Dos décadas después del lanzamiento de *Minds in Play*, Kafai considera que, aunque aquella iniciativa de implementar el construccionismo a través del desarrollo de videojuegos haya sido un éxito rotundo, su momento fue desafortunado:

Llegó al final del primer intento de llevar la programación a las escuelas, cuando el interés por la codificación escrita por uno mismo ya se estaba desplazando hacia los paquetes de *software* enlatado característicos de las computadoras en las escuelas en las décadas de 1990 y 2000. La llegada de los CD-ROM multimedia y los navegadores de Internet aparentemente negaron la necesidad de aprender el lenguaje de las computadoras; se pensaba que los clics del *mouse*, no el código, eran todo lo que se requería para tomar el control de la máquina. El entusiasmo por el Logo y la programación súbitamente desapareció (Kafai y Burke, 2016, p. 22).

Según Q. Burke y Kafai (2014), afortunadamente la tendencia se ha revertido y, en los últimos años, ha vuelto a crecer el interés por el construccionismo, la enseñanza de la programación y el desarrollo de videojuegos como estrategia pedagógica:

Durante la última década, se han producido dos acontecimientos importantes que ayudan a resucitar la visión constructivista que Papert tuvo inicialmente, de niños diseñando, construyendo y compartiendo sus propios videojuegos. Primero, hay un número creciente de herramientas de fácil acceso para la creación de juegos que están específicamente dirigidas a aquellos jóvenes interesados no solo en jugar, sino también en diseñar y desarrollar sus propios videojuegos . . . El segundo acontecimiento es que, además de las herramientas, ahora también han comenzado a desarrollarse comunidades enteras para compartir y remezclar juegos en línea . . . Estas comunidades en línea ofrecen a los diseñadores y programadores un repositorio para almacenar y un foro para compartir su trabajo, para colaborar con otros, así como también para tomar y remezclar muestras de los videojuegos subidos por otros usuarios (p. 690).

Entre los factores que han impulsado el cambio de rumbo hacia el paradigma constructor de desarrollo de videojuegos, Kafai y Burke (2015) mencionan la iniciativa para promover el pensamiento computacional, la necesidad de ampliar la participación en la informática y la popularización de una cultura del bricolaje (*DIY* o *do-it-yourself*) entre los jóvenes actuales. Sin embargo, afirman que el principal impulso podría deberse al fomento del *modding*:

El ímpetu central para un cambio podría provenir de la propia industria y cultura del juego. Después de todo, algunos de los juegos más populares en el mercado actual incluyen *modding* de niveles y personajes como característica central . . . y fomentan ese *modding* como parte del juego hasta que la próxima versión esté disponible (p. 314).

No obstante, para evitar que el desarrollo de videojuegos como estrategia pedagógica pierda el impulso que ha ganado recientemente, se deberán superar algunos obstáculos: la preferencia de ciertos docentes por el instruccionismo, las dificultades técnicas y cierta falta de interés de la industria. Estos desafíos se tratan a continuación.

### 6.2.1. Preferencia por el instruccionismo

Muchos docentes están acostumbrados al instruccionismo y son reacios a adoptar el enfoque constructor. Al analizar la prácticamente total ausencia de este enfoque en las discusiones sobre la eficacia de los juegos serios, Kafai y Burke (2015) señalan que la primera y más obvia razón se remonta al deseo instructorista de tener un producto didáctico terminado y descargable —el juego en sí— como el responsable (en lugar del instructor) de enseñarle al niño:

Posicionar la tecnología como el *tutor* representa el modo predeterminado implícito de integración tecnológica. En la década de 1980, las computadoras se introdujeron en las escuelas como máquinas de enseñanza, y esta percepción de los dispositivos como instructores subrogantes persiste aún hoy (p. 314).

### 6.2.2. Dificultades técnicas

Este obstáculo que deberá superarse para consolidar el desarrollo de videojuegos como una estrategia pedagógica plausible de ser aplicada tiene dos caras: una dificultad técnica subjetiva —percibida por algunos docentes— y una dificultad técnica objetiva —referida a la falta de las condiciones tecnológicas necesarias para llevar a la práctica el enfoque constructorista en ciertos ámbitos—. Sobre la primera, Kafai y Burke (2015) señalan que otra razón para la impopularidad general del desarrollo constructorista de videojuegos en las escuelas puede deberse simplemente a que los docentes de la Educación Primaria y Secundaria consideran el esfuerzo como demasiado técnico, especialmente si se lo asocia con el aprendizaje de la programación (p. 314).

Los mismos autores consideran que en las escuelas de bajos ingresos esta situación se agrava:

La realidad crítica es que los niños de las escuelas de bajos ingresos ... reciben considerablemente menos acceso a actividades informáticas cuyo foco es la creación real de contenido digital. Las computadoras en estas escuelas de bajos ingresos son más tutores que herramientas para crear contenido, y capacitan e instruyen a los estudiantes con base en algún contenido académico particular (p. 325).

Acercas de la falta de condiciones objetivas para implementar el enfoque constructorista basado en el desarrollo de videojuegos, Q. Burke y Kafai (2014) mencionan que con frecuencia, las escuelas y los clubes extracurriculares carecen del tiempo y de la competencia de llevar a la práctica la colaboración a gran escala (sin mencionar la capacidad técnica) requeridos para facilitar que los niños creen sus propios videojuegos (p. 690).

### 6.2.3. Falta de interés de la industria

Durante años, la falta de interés de la industria fue un gran obstáculo para la estrategia constructivista basada en videojuegos, tal como explican Kafai y Burke (2015):

Hasta hace poco, la industria del juego realmente no quería que los jugadores participaran en ningún diseño o modificación de los juegos que se producían para el mercado. Al fin y al cabo, era su producto y tenía derechos de autor (p. 314).

De acuerdo con E. R. Gee y Tran (2016), este obstáculo afortunadamente ya parece haber sido superado de forma definitiva, pues algunos diseñadores de juegos comerciales fomentan el *modding* lanzando herramientas de *software* que permiten a los jugadores modificar el contenido de sus juegos:

Los fanáticos comparten tales herramientas, tutoriales y *mods* completos, así como también discuten *modding*, a través de comunidades en línea, algunas dedicadas exclusivamente a compartir conocimiento y estrategias para el *modding* de juegos . . . [Por ejemplo] *Little Big Planet* es un juego diseñado principalmente para el entretenimiento en el que los jugadores no solo pueden jugar, sino también crear sus propios juegos con su editor de niveles (p. 252).

### 6.3. Beneficios del aprendizaje mediante el desarrollo de videojuegos

El aprendizaje mediante el desarrollo de videojuegos trae beneficios que hoy en día son muy valorados. Hayes y Games (2008) señalan que el valor de la *mentalidad de diseñador* ha sido reconocido por muchos expertos en educación, que ven tal perspectiva como una habilidad fundamental requerida para la plena participación en la economía del conocimiento:

Las corporaciones actualmente buscan empleados capaces de resolver problemas de manera creativa y efectiva, así como de producir nuevos conocimientos que puedan ayudarlos a adaptarse y a ser competitivos en el mercado global . . . El pensamiento de diseño (*design thinking*) ha llamado cada vez más la atención de los líderes empresariales que buscan mejorar la capacidad de sus empleados y compañías para identificar soluciones creativas a desafíos cada vez más complejos . . . Existe una sólida evidencia de que esta forma de pensar merece —y está recibiendo— una atención seria (pp. 309-310).

Concretamente, Kafai y Burke (2015, pp. 318-325) clasifican los beneficios de esta estrategia pedagógica según pertenezcan a la *dimensión personal* —en la cual se encuadran el aprendizaje de la programación, el aprendizaje de contenidos y el aprendizaje sobre el aprendizaje—, la *dimensión social* —a la que pertenece la promoción del trabajo colaborativo— o la *dimensión cultural* —en la cual se enmarca la reducción de las brechas de género y raciales—. Estos beneficios se tratan a continuación.

### 6.3.1. Aprendizaje de la programación

Según Hayes y Games (2008), el uso más común de la creación de juegos en entornos educativos ha sido con el propósito de ayudar a los estudiantes a aprender herramientas y conceptos de programación (p. 311).

Q. Burke y Kafai (2014) utilizan una retórica claramente constructivista cuando justifican el desarrollo de videojuegos como estrategia para el aprendizaje de la programación:

Al basar la codificación en el juego, los niños aprenden los conceptos fundamentales de la programación mientras participan en una actividad práctica; el desarrollo de un objeto real (en este caso, un juego digital) que tiene valor personal para el usuario ayuda a desmitificar el proceso de codificación, haciéndolo más real y relevante para el aprendiz (p. 691).

Reseñando la investigación que ya había relatado 20 años antes en *Minds in Play*, Kafai destaca que los estudiantes que participaron del *Game Design Project* original mejoraron significativamente en las prácticas computacionales, como la escritura de programas y su depuración, en comparación con los estudiantes que estaban aprendiendo programación solo en el contexto de proyectos independientes más pequeños no relacionados con los juegos, confirmando que el contexto (juegos vs. pequeños programas no lúdicos), la complejidad y el marco de tiempo (programación semanalmente vs. programación a diario) tienen un impacto considerable sobre la capacidad de los principiantes para aprender a codificar (Kafai y Burke, 2015, p. 318).

Estos autores señalan que en experiencias posteriores donde los estudiantes tuvieron que desarrollar videojuegos, se observaron el uso de estructuras de programación simples y la presencia de construcciones más complejas, como las abstracciones creadas por los estudiantes, la ejecución simultánea y el manejo de eventos, todo lo cual indica patrones de orden superior. También el uso de conceptos computacionales clave, como bucles, variables y condicionales, pero usabilidad moderada y bajos niveles de organización y documentación de código (p. 318).

Asimismo, defienden que en contextos educativos se desarrollen videojuegos mediante el remezclado del *software* existente:

Algunas voces han criticado el reciente resurgimiento del enfoque de *codificación para todos* en las escuelas como poco realista y una versión demasiado simplificada de lo que, en verdad, es la programación *real*. Pero nosotros vemos que estas actividades de creación de juegos proporcionan una entrada de bajo umbral al mundo de la programación . . . El remezclado de código es una práctica de programación auténtica utilizada por la comunidad de programadores en general y, de hecho, constituye un enfoque pedagógico válido para proporcionar a los niños una formación básica en ciencias de la computación (p. 318).

### 6.3.2. Aprendizaje de contenidos

Generalmente, los videojuegos no están compuestos solo de código y, por ello, su desarrollo permite aprender mucho más que programación. Según Zimmerman (2013), el diseño de juegos es radicalmente interdisciplinario. Hacer un juego incluye la creación de un sistema formal de reglas, a la vez que se diseña una experiencia de juego humana para un contexto cultural y social particular:

El diseño de juegos incluye matemática y lógica, estética y narrativa, escritura y comunicación, diseño visual y de audio, psicología y comportamiento humanos, y comprensión de la cultura a través del arte, el entretenimiento y los medios populares. Para el diseño de videojuegos, también la informática y la alfabetización tecnológica se vuelven parte de la ecuación . . . El diseño de juegos representa formas de aprendizaje multimodales sobre las que los teóricos de la educación . . . han estado hablando durante años (pp. 162-164).

Q. Burke y Kafai (2014) son partidarios de aplicar el desarrollo de videojuegos en la enseñanza de las más variadas disciplinas, porque así como la creación de juegos tiene el potencial de hacer que el aprendizaje de la programación sea más palpable, también ofrece la oportunidad de basar en actividades prácticas la enseñanza de temas académicos más tradicionales, como son las matemáticas, la lengua, el arte o la historia (pp. 691-692).

Posteriormente, estos autores sugieren que en lugar de conceptualizar las actividades de creación de juegos únicamente como un contexto para aprender programación y habilidades de diseño de *software*, el aprendizaje de la codificación se puede situar dentro de un contexto más amplio: el desarrollo de aplicaciones. Después de todo, el diseño de aplicaciones abarca el diseño de contenido y, por ello, los diseñadores de videojuegos necesitan aprender no solo sobre el contenido y las habilidades que se incluirán, sino también sobre la codificación (Kafai y Burke, 2015, p. 321).

No obstante, también advierten sobre la potencial desconexión entre el mundo del juego desarrollado por los estudiantes —sus historias y su mecánica— y el contenido abordado:

Algunos estudios han observado que cuando se carga a los estudiantes con este doble enfoque de aprendizaje de contenido y codificación, ellos priorizan el mundo del juego y la creación de historias por encima del compromiso con el contenido . . . Esto no es algo inesperado y puede abordarse proporcionando andamios de instrucción que proporcionen una mejor integración del contenido y el juego . . . Estos desafíos y desconexiones entre la mecánica del juego y el contenido no son nuevos; de hecho, se pueden encontrar en muchos juegos de instrucción que solo muestran una integración superficial o extrínseca del juego con el contenido (p. 321).

### 6.3.3. Aprendizaje sobre el aprendizaje

A nivel personal, el aprendizaje mediante el desarrollo de videojuegos ofrece un tercer beneficio más allá del aprendizaje de la propia programación y de los propios contenidos didácticos: el aprendizaje sobre el aprendizaje, el cual —según Kafai y Burke (2015)— corresponde a la idea de que los niños aprendan sobre su propio pensamiento y aprendizaje, lo que se conoce como *reflexión* o *metacognición*, dado que los niños aprenden a articular procedimientos, a reconocer la repetición y a *depurar* su propio pensamiento cuando los programas no funcionan como se espera:

Al comparar experimentalmente a un grupo de estudiantes que juegan con un grupo de estudiantes que hacen juegos, se determinó que estos últimos demostraban un compromiso significativamente más profundo con su aprendizaje y con el uso de estrategias —como el *análisis sistémico*, la *toma de decisiones* y la *resolución de problemas*— que el grupo de estudiantes que solamente jugaban . . . Aunque la escritura de código y el contenido son los conocimientos y habilidades más fácilmente reconocibles que se abordan al hacer juegos, el *diseño*, la *resolución de problemas* o las *habilidades de pensamiento sistémico* . . . también caen dentro de una categoría general más amplia que llamamos *aprendizaje sobre el aprendizaje* . . . Hacer juegos requiere que los diseñadores piensen en una metaestructura en la que se integren la mecánica del juego, las interacciones y el contenido. En particular, la noción de pensamiento sistémico ha recibido mucha atención, tal vez por el creciente interés en utilizar el pensamiento sistémico complejo como un marco para abordar el aprendizaje de la ciencia y la noción de pensamiento computacional como sistema de diseño (p. 321).

### 6.3.4. Promoción del trabajo colaborativo

El aprendizaje mediado por tecnología —por ejemplo, desarrollando videojuegos— puede promover el trabajo colaborativo, lo cual es beneficioso a nivel social. De acuerdo con Roschelle, Pea, Hoadley, Gordin y Means (2000), una línea influyente de investigación sobre el aprendizaje se centra en la base social para el aprendizaje de los niños, inspirada en la investigación fundacional del psicólogo ruso Vygotski. Los contextos sociales les dan a los estudiantes la oportunidad de poner en práctica con éxito habilidades más complejas de las que podrían ejecutar solos:

Realizar una tarea con otros brinda una oportunidad no solo de imitar lo que otros están haciendo, sino también de discutir la tarea y hacer que el pensamiento sea visible . . . A través de conversaciones y gestos sociales informales, los estudiantes y los docentes pueden brindar consejos explícitos, resolver malentendidos y garantizar que los errores se corrijan. Además, las necesidades sociales a menudo son una razón para aprender. Debido a que la identidad social de un niño se mejora al participar en una comunidad o al hacerse miembro de un grupo, involucrar a los estudiantes en una actividad social intelectual puede ser un poderoso motivador y puede conducir a un mejor aprendizaje que depender del trabajo individual (p. 80).

Con el propósito de investigar la viabilidad y el valor agregado que el enfoque denominado *Creación Colaborativa de Juegos Digitales* (CDGM - *Collaborative Digital Game Making*) tiene para el aprendizaje —especialmente para el apoyo de las habilidades transversales de los estudiantes, como la colaboración, la creatividad, la resolución de problemas y la alfabetización en TIC—, entre 2012 y 2014 un consorcio formado por el *Istituto per le Tecnologie Didattiche, Consiglio Nazionale delle Ricerche* (Italia), la *Tampere University of Technology* (Finlandia), la *Manchester Metropolitan University* (Reino Unido) y la *Katholieke Universiteit Leuven* (Bélgica) —en el marco del *Lifelong Learning Programme* de la Comisión Europea— llevó a cabo el proyecto *MAGICAL - Making Games in Collaboration for Learning*. En el informe final, los autores señalan que ven el CDGM como una forma innovadora, centrada en el estudiante, de aprender haciendo, que aprovecha y aumenta las ventajas establecidas del aprendizaje basado en juegos, a la vez que desafía a los equipos de estudiantes a colaborar en el diseño y la creación de sus propios juegos digitales —juegos que producen para sí mismos, para su comunidad de compañeros de clase, amigos y familiares, y para sus redes más amplias.



Asimismo, los autores del informe final mencionan los resultados positivos que ha arrojado el proyecto:

La experiencia adquirida en MAGICAL muestra que la creación de juegos puede integrarse de forma fructífera en la práctica escolar . . . El proyecto ha arrojado resultados positivos, particularmente en lo que respecta a la aceptación, la motivación y el compromiso de los alumnos. En general, estos resultados sugieren que la creación de juegos puede implementarse en una variedad de contextos educativos con el objetivo de mejorar e innovar los procesos de enseñanza y aprendizaje, especialmente en lo que respecta a las habilidades del siglo XXI (European Commission - Education, Audiovisual & Culture Executive Agency, 2014).

En los Estados Unidos, uno de los principales exponentes del aprendizaje mediante el desarrollo colaborativo de videojuegos es Globaloria. Según Kafai y Burke (2015), utilizando lenguajes empleados en la industria, Globaloria es un plan de estudios nacional basado en la creación de juegos que, hasta 2011, ha apoyado a más de 8000 estudiantes de 14 estados en el diseño colaborativo de sus propios videojuegos hechos en la escuela como parte de actividades curriculares. También destacan los resultados de este plan de estudios:

Los resultados son prometedores no solo porque se logra que los estudiantes colaboren de forma más eficaz en proyectos significativos para ellos, sino porque también se aumenta la autoeficacia de los estudiantes con la tecnología . . . Premiando con los *Globey Awards* . . . a los principales prototipos de juegos, Globaloria promueve el espíritu de colaboración interponiendo un espíritu competitivo entre los jóvenes creadores de juegos (pp. 322-323).

### **6.3.5. Reducción de las brechas de género y racial**

En todo el mundo existe una brecha de género en el campo de la informática, a pesar de los esfuerzos gubernamentales para eliminarla o, al menos, para reducirla. Hayes y Games (2008) señalan que las mujeres continúan inscribiéndose en cursos formales de educación informática en números mucho menores que los hombres y mencionan un programa estadounidense concreto lanzado para intentar paliar esta crisis:

A través de fondos específicos como el *Programa para Equidad de Género en Ciencia, Matemáticas, Ingeniería y Tecnología*, la NSF (*National Science Foundation*) ha financiado cientos de proyectos educativos destinados a apoyar la participación de niñas y mujeres en la informática y campos relacionados (pp. 314-315).

Además de la brecha de género, la brecha racial también constituye un problema en la industria de los videojuegos. Según Sheikh (2017), no hay suficientes diseñadores de raza negra o asiáticos trabajando en esta industria:

Las cifras más actuales del empleo en la industria de los juegos provienen de una encuesta de *Creative Skillset*, de 2015. Sugiere que solo el 4% de los diseñadores de juegos en el Reino Unido son de origen negro o asiático, frente a la media nacional del 10% en todos los sectores.

Por suerte, el aprendizaje mediante el desarrollo de videojuegos puede ayudar a reducir estas brechas. De acuerdo con Q. Burke y Kafai (2014), un creciente número de estudios muestra que los estudiantes pertenecientes a los grupos menos representados en la informática —ya sea como curso de estudio o como campo profesional— afirman estar más comprometidos con el aprendizaje de la programación cuando tal actividad se relaciona con el desarrollo y el perfeccionamiento de sus propios juegos digitales (p. 691).

Kafai y Burke (2015) justifican esta potencialidad de los videojuegos resaltando que los mismos son uno de los grandes ecualizadores de la sociedad moderna: son jugados por prácticamente todos, independientemente de la raza, el sexo y, ahora, la edad (p. 325).

#### **6.4. Desarrollo de videojuegos sin escribir código**

Como se acaba de mostrar, el aprendizaje mediante el desarrollo de videojuegos puede ofrecer muchos otros beneficios además del aprendizaje de la programación. Cuando el propósito del desarrollo de un videojuego no es aprender a programar, sino conseguir alguno de los otros beneficios, lo ideal es utilizar herramientas que permitan trabajar sin escribir código. Q. Burke y Kafai (2014) mencionan algunas posibilidades:

Mientras que la programación de un videojuego tradicionalmente requería una extensa escritura de código . . . , muchas herramientas actuales . . . simplemente se basan en un enfoque de *arrastrar y soltar* en el que uno simplemente puede arrastrar el ratón sobre varios objetos y hacer clic y soltar para agregar funcionalidad al juego, como la movilidad del jugador o la puntuación. Para Scratch . . . estos objetos son ladrillos programables, para AgentSheets . . . son hojas digitales en capas, mientras que para GameMaker y GameSalad, los objetos sobre los que se puede hacer clic son ventanas emergentes en cascada (p. 696).

A continuación se presenta un resumen de las principales herramientas actualmente disponibles para desarrollar videojuegos sin escribir código (tabla 26).

Tabla 26

*Principales herramientas actualmente disponibles para desarrollar videojuegos sin escribir código*

Herramienta	Descripción
AgentSheets	<i>Software</i> comercial con interfaz de arrastrar y soltar que mediante <i>Conversational Programming</i> permite a los usuarios crear sus propios juegos y simulaciones basados en agentes y publicarlos en la Web. ► <a href="http://www.agentsheets.com">http://www.agentsheets.com</a>
Alice	<i>Software</i> de código abierto. Es un entorno basado en bloques diseñado para ser la primera exposición a la programación orientada a objetos. Los estudiantes establecen el comportamiento de objetos y personajes en un mundo virtual en 3D. En una ventana contigua, opcionalmente puede verse el código en Java. ► <a href="http://www.alice.org">http://www.alice.org</a>
Gamefroot	Sitio web que permite crear y publicar juegos en línea gratuitamente. El desarrollo se lleva a cabo visualmente. ► <a href="http://gamefroot.com">http://gamefroot.com</a>
GameMaker: Studio	<i>Software</i> comercial con interfaz de arrastrar y soltar (DnD). También permite usar opcionalmente el lenguaje GML. Puede generar juegos para Windows, macOS, Ubuntu, Android, iOS, HTML5, PlayStation 4, Xbox One y otros. ► <a href="http://www.yoyogames.com/gamemaker">http://www.yoyogames.com/gamemaker</a>
GameSalad	<i>Software</i> comercial con interfaz de arrastrar y soltar. Es una plataforma de desarrollo de juegos que permite a cualquiera crear juegos sin escribir código. Puede generar juegos para Windows, Android, iOS, HTML5 y otros. ► <a href="http://gamesalad.com">http://gamesalad.com</a>
Gamestar Mechanic	Juego en línea gratuito, ideado para enseñar los principios del diseño de juegos y el pensamiento sistémico a niños de 7 a 14 años. Posee una comunidad en la Web, donde los usuarios pueden compartir sus juegos. ► <a href="http://gamestarmechanic.com">http://gamestarmechanic.com</a>
Kodu	<i>Software</i> gratuito para crear juegos en la PC y la Xbox a través de un lenguaje de programación visual simple. Las creaciones se pueden compartir en Xbox Live. ► <a href="http://www.kodugamelab.com">http://www.kodugamelab.com</a>
MissionMaker	<i>Software</i> comercial que permite crear rápidamente mundos en 3D visualmente ricos para juegos en primera persona. Los juegos solamente corren en Windows dentro de un entorno denominado <i>MissionPlayer</i> . ► <a href="http://creativeedutech.com/products/missionmaker">http://creativeedutech.com/products/missionmaker</a>
Pocket Code	Aplicación gratuita para celulares con Android. Tiene una interfaz de arrastrar y soltar bloques similar a Scratch. ► <a href="http://share.catrob.at/pocketcode">http://share.catrob.at/pocketcode</a>
RPG Maker	<i>Software</i> comercial para crear juegos de rol sin saber programación, aunque también permite usar JavaScript. ► <a href="http://www.rpgmakerweb.com">http://www.rpgmakerweb.com</a>
Scratch	<i>Software</i> gratuito con interfaz de arrastrar y soltar, basado en bloques, para el desarrollo de juegos, historias y arte, tanto en un editor sin conexión como en línea, en una comunidad que tiene casi 30 millones de usuarios. ► <a href="http://scratch.mit.edu">http://scratch.mit.edu</a>
Snap!	<i>Software</i> gratuito basado en Scratch, con mejoras inspiradas en Scheme. Permite construir bloques propios. ► <a href="http://snap.berkeley.edu">http://snap.berkeley.edu</a>
Sploder	Sitio web que permite crear y publicar juegos en línea gratuitamente. El desarrollo se lleva a cabo visualmente. ► <a href="http://www.sploder.com">http://www.sploder.com</a>
Stencyl	<i>Software</i> gratuito con interfaz de arrastrar y soltar, basado en bloques. También permite usar opcionalmente Haxe, un lenguaje similar a ActionScript. Puede generar juegos gratuitamente (en Flash) o pagando una licencia (otros). ► <a href="http://www.stencyl.com">http://www.stencyl.com</a>
ToonTalk	<i>Software</i> de código abierto para Windows o en línea ( <i>ToonTalk Reborn</i> ). A través de su interfaz de arrastrar y soltar, permite animar dibujos, que luego pueden ser manipulados para crear rompecabezas y simulaciones. ► <a href="http://www.toontalk.com">http://www.toontalk.com</a>
Tynker	Plataforma de programación para niños que utiliza una interfaz de arrastrar y soltar. Ofrece cursos pagos. ► <a href="http://www.tynker.com">http://www.tynker.com</a>

*Nota:* Basado en “Decade of Game-Making for Learning: From Tools to Communities” por Q. Burke y Y. B. Kafai, 2014, *Handbook of Digital Games* editado por M. C. Angelides y H. Agius, 2014, pp. 694-695. Copyright 2014 por The Institute of Electrical and Electronics Engineers

## 6.5. Formas de aprender programación mediante el desarrollo de videojuegos

Existen distintas formas de aprender a programar desarrollando videojuegos. Por un lado, es posible encontrar en el mercado una gran cantidad de libros didácticos en los cuales los conceptos de los lenguajes enseñados son aplicados al desarrollo de uno o más videojuegos. Por otra parte, es frecuente escuchar relatos de docentes que exponen conceptos de lenguajes de programación de propósito general en clase y hacen que sus estudiantes después pongan a prueba —a través del desarrollo de videojuegos— los conocimientos adquiridos. Por último, otros docentes relatan que enseñan la disciplina usando herramientas de programación que facilitan la creación de videojuegos. Estas tres variantes se tratan a continuación.

### 6.5.1. Libros que enseñan lenguajes a través del desarrollo de videojuegos

El típico libro que enseña a programar en algún lenguaje mediante el desarrollo de videojuegos generalmente está dirigido a aprendices que, leyendo por su cuenta, reconstruirán el camino que el autor del libro siguió cuando creó los juegos propuestos. A veces, el código fuente de un juego se encuentra impreso en el libro, comentado y acompañado de capturas de pantalla, por lo cual cabe al lector decidir si efectivamente probará el juego o si, por el contrario, preferirá estudiarlo solamente en el libro.

En *The essential guide to HTML5: Using games to learn HTML5 and JavaScript* (J. Meyer, 2010), la autora presenta los siguientes minijuegos, cada uno acompañado de capturas de pantalla y del código fuente comentado: Craps (Pase inglés); Tiro de cañón; Memotest; Quiz (Países y capitales); Laberintos; Piedra, papel o tijera; Ahorcado y Blackjack. Aunque son todos juegos muy sencillos, tienen una aceptable calidad gráfica.

En cambio, en *Beginning C++ through game programming* (Dawson, 2011), el autor presenta los siguientes juegos que solo corren en *modo texto* (en la consola de Windows), también acompañados de capturas de pantalla y del código fuente comentado: *Lost fortune* (el jugador ingresa datos y el programa crea una historia); *Guess my number* (el jugador debe adivinar un número entre 1 y 100); *Word Jumble* (el jugador debe adivinar una palabra viendo sus letras mezcladas); Ahorcado; Tres en raya; *Critter caretaker* (el jugador debe cuidar una mascota virtual) y Blackjack.

A diferencia de los anteriores, *Learning Java through games* (Stanchev, 2014) no incluye el código fuente comentado de todos los juegos que presenta, sino solamente el de los siguientes: *Multiplication game* (el jugador debe ingresar el producto de dos números enteros); *Guess my number* (el jugador debe adivinar un número entre 0 y 999); *Trading Game* (el jugador compra o vende peras o manzanas a lo largo de 10 días, intentando maximizar su ganancia); Generala; Batalla naval; Cara o cruz; Breakout; *Typing game* (el jugador debe presionar rápidamente las teclas de los caracteres que van apareciendo en la pantalla); *Think of a number* (el programa adivina un número entre 1 y 1000 pensado por el jugador) y Tres en raya. Además, el autor incentiva al lector para que modifique los juegos presentados y le propone que también implemente los siguientes: Ahorcado; Nim; 3x3 Puzzle; Reversi y Tetris. La calidad gráfica de los juegos va aumentando a lo largo del libro, comenzando con juegos que solo corren en *modo texto* y terminando con juegos que usan gráficos animados.

En *Python: An Introduction to Programming* (Parker, 2017), el autor empieza mostrando versiones simplificadas de los juegos *Guess my number* (el jugador debe adivinar un número entre 1 y 100) y Piedra, papel o tijera. Después les va agregando características avanzadas hasta llegar a las versiones finales. Más adelante presenta los juegos completos Primo o no primo (el jugador debe indicar si un número es o no es primo); Craps (Pase inglés); Sticks (si el jugador retira el último palito, pierde); Nim; Jeopardy!; Cat-A-Pult (el jugador debe lanzar gatos para eliminar un ratón) y Breakout. Al igual que en el libro anterior, la calidad gráfica de los juegos va aumentando: los primeros juegos solo corren en *modo texto* y los últimos utilizan animaciones.

Como puede observarse, cuando el foco está puesto en enseñar programación y no diseño de videojuegos, en los libros se usan casi siempre los mismos juegos, los cuales son, en su mayoría, minijuegos de computadora que no incluyen animaciones ni sonidos, pues la prioridad es mostrar las características de los lenguajes y la lógica de la programación.

### **6.5.2. Desarrollo de videojuegos usando lenguajes de propósito general**

En numerosas publicaciones académicas es posible encontrar testimonios de docentes narrando las experiencias de sus estudiantes que aprendieron programación y

desarrollaron videojuegos utilizando lenguajes y entornos de propósito general, es decir, aquellos que no son específicos para la creación de juegos.

Feldgen y Clúa (2004) decidieron exigir que sus estudiantes de una materia de programación en Pascal del primer año de una carrera universitaria llevaran a cabo dos proyectos: primeramente, programar un juego de mesa para jugarlo fuera de línea y, después, reimplementarlo para que pudiera ser jugado en la Web en un entorno que registrara a los jugadores y sus puntajes. Así manifiestan el motivo de su decisión:

Su mundo de la computación son los sistemas informáticos multiprogramados con impresionantes interfaces humanas para juegos y la Web. Los estudiantes no pueden relacionar estos sistemas con los ejercicios de programación clásicos que requieren crear programas formados por un único subproceso que realiza una secuencia de cálculos. Ante esto, decidimos iniciar a nuestros estudiantes en la resolución de problemas utilizando lo que ellos ven como problemas del mundo real: los juegos y la programación Web (p. 11).

La retención de los estudiantes, el fomento de la colaboración entre ellos y la introducción de mayores proyectos de programación fueron los objetivos que llevaron a Garlick y Akl (2007) a innovar en un curso de programación en C++ del segundo semestre de una carrera universitaria. Sin embargo, en lugar de llevar a cabo proyectos grupales con división de tareas y colaboración —como se suele hacer habitualmente—, estos autores tomaron otro camino: hicieron que sus estudiantes compitieran entre sí individualmente programando agentes de *software* para que se enfrentaran en un torneo. Así lo explican:

La tarea consistía en crear una clase de C++ que interactuaría con un motor de juego creado por el instructor. El juego era *TankWar*, y cada jugador proporcionaría el código para operar autónomamente su tanque en el campo de batalla. El motor del juego pasaría información sobre el estado del juego a cada jugador por turno . . . y recibiría la reacción del agente de cada jugador en función de este estado. El tanque de cada jugador, en cada turno, podría dispararle al enemigo, plantar una mina, barrer y remover minas adyacentes al jugador, avanzar un espacio o detenerse. En cada batalla participarían dos jugadores (p. 4).

Burguera y Guerrero (2014) necesitaban aumentar la motivación de sus estudiantes de una materia —del segundo año de una ingeniería informática— en la cual se trabaja con lenguaje ensamblador. De esta manera explican la solución que encontraron:

Se decidió dotar a la parte práctica de la asignatura de un hilo conductor: el desarrollo de un videojuego . . . Este hilo conductor es especialmente motivador para los estudiantes, y aumenta notablemente su interés . . . Por un lado, porque persiguen un objetivo atractivo: diseñar un videojuego tiene un componente a la vez creativo y cotidiano, dado que el mundo de los videojuegos es de sobra conocido por nuestros alumnos. Por otro lado, porque a medida que se avanza en el proceso de programación, los estudiantes son conscientes de que los conocimientos que adquieren trascienden claramente los límites de la propia asignatura (p. 36).

Soares, Fonseca y Martin (2015) rediseñaron su curso de introducción a la programación con Java en una carrera universitaria, centrándolo en el aprendizaje basado en problemas y en el diseño de juegos. Así narran cómo lo hicieron:

Hemos rediseñado el curso incorporando aprendizaje basado en problemas y diseño de juegos . . . El curso culmina con un proyecto grupal orientado a crear un juego que requiere que los estudiantes usen los conceptos aprendidos a lo largo del semestre . . . Sin embargo, los estudiantes también aprenden nuevos conceptos por sí mismos cuando quieren implementar ciertas características en sus juegos . . . Aproximadamente en la novena o la décima semana, cuando se ha cubierto la mayoría de los conceptos esenciales de programación necesarios para crear un juego, los estudiantes envían una propuesta para el proyecto de juego final de su grupo. Por lo general, los estudiantes intentan recrear algunos de sus juegos favoritos o proponen un juego que combinaba características de dos o más juegos conocidos (p. 132).

Con el propósito de aumentar la motivación de sus estudiantes de una materia de introducción a la programación en una carrera de ingeniería, Blas, Hauque, Re y Castellaro (2017) les solicitaron la implementación de un juego (denominado King Coder) en C++, para ser jugado por sus compañeros de clase y por los estudiantes de futuros cursos. Así describen estos autores los tres módulos que fueron desarrollados:

Al elegir el primer módulo de King Coder, el programa imprime en pantalla un código C++ simple y corto con todas sus líneas numeradas. Seguidamente pide al usuario que indique los números de líneas en las cuales considera que existen errores de sintaxis . . . En base a los valores indicados, el programa determina la correctitud o no de la solución. En el caso de ejecutar el segundo módulo, el programa imprime la consigna de un problema de programación junto con una propuesta de solución. Ante este ejercicio el usuario debe analizar cuidadosamente el código y determinar la veracidad de la consigna dada . . . Finalmente, en el juego de opciones múltiples (*Code Quiz*) la herramienta imprime un código C++ acompañado de una pregunta y 4 posibles respuestas. El usuario debe seleccionar una única opción (p. 9).

### 6.5.3. Herramientas de programación que facilitan la creación de videojuegos

Hoy en día, existen herramientas de programación que facilitan la creación de videojuegos, evitando que el desarrollador tenga que codificar desde cero cada parte de una nueva producción. Estas herramientas son las API (*Application Programming Interfaces*), los SDK (*Software Development Kits*), las bibliotecas (*libraries*), los marcos de trabajo para videojuegos (*videogame frameworks*) y los motores de videojuego (*videogame engines*).

La más específica de estas herramientas es la API, que no es más que una interfaz para interactuar con un servicio. Según Herman (2014), muchos sistemas de *software* tienen sus propias formas de interactuar con los usuarios y de mostrar y administrar los datos. Sin embargo, podría haber características adicionales que los diseñadores no incorporaron al sistema:

Algunos sistemas proporcionan una *interfaz de programación de aplicaciones* (API) para que los usuarios que saben cómo programar puedan interactuar con el sistema utilizando algunas funciones y organizaciones de datos definidas . . . Las llamadas a la API permiten a los usuarios mejorar y administrar el sistema de *software* de maneras novedosas (p. 219).

Dos API destacadas son *Direct3D* («Direct3D», s. f.) —utilizada para la generación de imágenes tridimensionales en Microsoft Windows y Xbox— y *Java 3D* («Java 3D API», s. f.) para la plataforma Java.

Menos específicos son los SDK, ya que, de acuerdo con Herman (2014), estos pueden incluir una o varias API:

Más allá de las API, algunos sistemas y productos tienen un *kit de desarrollo de software* (SDK), que puede incluir herramientas, *software* y *hardware*, además de una o más API. Al usar un SDK, un desarrollador puede ampliar las capacidades e interfaces de un sistema e incluso producir una versión mejorada de valor agregado del producto original (p. 219).

Un ejemplo de SDK es kit de desarrollo de *DirectX* («DirectX SDK», s. f.) —utilizado para llevar a cabo tareas relacionadas con multimedia—, el cual actualmente es distribuido por Microsoft como parte de Windows SDK.

Una biblioteca (*library*) es una colección de funciones específicas para llevar a cabo tareas en determinadas áreas. Según Sherrod (2015), en el contexto del desarrollo de juegos, una biblioteca es, por ejemplo, una colección de comportamiento específico:



Piense en la STL (*Standard Template Library*) de C++, una colección de estructuras de datos y algoritmos escritos en el lenguaje C++. Las listas enlazadas, los arreglos, las colas, las tablas *hash* y muchos más forman parte de esta biblioteca de plantillas estándar. Puede pensar en una biblioteca como una colección de códigos estándar, como clases, que definen comportamientos específicos (p. 38).

Simões (2016) destaca que las bibliotecas agrupan funcionalidades según áreas y que cabe a cada programador elegir las y hacerlas trabajar juntas:

Hace algunos años, las aplicaciones se desarrollaban utilizando uno o más lenguajes de programación y, cuando se necesitaba alguna funcionalidad específica, una o más bibliotecas externas. Estas bibliotecas incluían desde funciones matemáticas o científicas, operaciones para la lectura y escritura de archivos de diversos formatos, herramientas de procesamiento de imágenes, algoritmos de aprendizaje automático, gráficos por computadora y una amplia variedad de otras tareas. Lo mismo sucedió en el desarrollo de los juegos de computadora. Durante años, podíamos encontrar bibliotecas para dibujar gráficos, reproducir sonido, simular la física, realizar operaciones de red, etc. Sin embargo, la programación de juegos requería de prácticamente todos estos aspectos y, por lo tanto, los programadores necesitaban elegir una biblioteca para cada una de estas tareas (gráficos, sonido, inteligencia artificial, física), y luego hacerlas encajar y trabajar juntas (pp. 223-224).

Algunas bibliotecas muy populares entre los desarrolladores de videojuegos son *LWJGL* («*LWJGL - Lightweight Java Game Library*», s. f.) para crear juegos en Java, *SFML* («*SFML - Simple and Fast Multimedia Library*», s. f.) —la cual está disponible para numerosas plataformas y permite programar en un gran número de lenguajes—, *pygame* («*Pygame*», s. f.) para desarrollar juegos en Python y basada, a su vez, en *SDL* («*SDL - Simple DirectMedia Layer*», s. f.) —la cual es multilenguaje y multiplataforma.

Para evitar que el desarrollador tenga que elegir las bibliotecas individualmente, surgieron los marcos de trabajo para videojuegos. De acuerdo con Simões (2016), los *frameworks* son un conjunto de bibliotecas elegidas y configuradas para trabajar juntas de forma inmediata (*out of the box*). Por lo tanto, un *framework* para videojuegos incluye, al menos, una biblioteca para la generación de imágenes y un conjunto de otras bibliotecas (probablemente más pequeñas) para algunas otras tareas:

Algunos de estos marcos de trabajo son muy completos e incluyen bibliotecas de reproducción de sonido, simulación de física, inteligencia artificial y conexión a redes. Otros son algo más limitados e incluyen solo la biblioteca de generación de imágenes y un par de funciones o módulos auxiliares para otras tareas. Los *frameworks* de juegos más recientes incluyen características aun más interesantes. Algunos permiten desarrollar juegos simultáneamente para diferentes plataformas y sistemas operativos (p. 224).

Un aspecto importante que caracteriza a los *frameworks* es la *inversión del control*, es decir, no es el programa del usuario el que invoca las funciones —como en la programación tradicional—, sino el , y cabe al usuario implementar las funciones. Según Herman (2014), un *framework* es un *software* que está incompleto, pero diseñado para ser completado y personalizado por su usuario:

Un ejemplo típico es un servidor web. Depende del usuario configurarlo, escribir código y usar herramientas para hacer que un servidor web sea utilizable . . . En algunos *frameworks*, el usuario escribe funciones que el *framework* invoca para guiar las decisiones y controlar la apariencia de la salida y la entrada (p. 219).

Entre los más populares *frameworks* para videojuegos se destacan *libGDX* («LibGDX», s. f.) para Java, *Phaser* («Phaser», s. f.) para JavaScript o TypeScript, *MonoGame* («MonoGame», s. f.) para C#, *LÖVE* («LÖVE», s. f.) para Lua y *Cocos2d* («Cocos2d», s. f.) para varios lenguajes —ya que el mismo es, en realidad, una familia de *frameworks* compuesta por *Cocos2d(Python)*, *Cocos2d-x* para C++, *Cocos2d-JS* para JavaScript, *Cocos2d-XNA* para C# y *Cocos2D-Swift* para Objective-C.

De acuerdo con Sherrod (2015), las más completas de las herramientas que facilitan la creación de videojuegos son los motores de videojuego:

Un *motor de videojuego* es una colección de herramientas que, juntas, se usan para hacer un videojuego . . . Se utilizan para abstraer los detalles comunes de un juego para que puedan ser reutilizados. Para los juegos en 2D, por ejemplo, un desarrollador no tiene que volver a escribir el código de generación de imágenes para cada proyecto nuevo. Es posible alimentar un sistema de generación abstracto con los datos del juego necesarios para producir el resultado deseado. Esto se conoce como desarrollo dirigido por datos (*data-driven development*) . . . Los motores de juego pueden incluir, entre otros, estos componentes: sistema de generación de imágenes; física y detección de colisiones; control y reproducción de audio; detección y captura de entrada; lenguaje de *scripting*; inteligencia artificial . . . y editores (pp. 36-37).

Algunos motores de videojuego destacados son *Unity* («Unity», s. f.) para programar en C#, *Unreal Engine* («Unreal Engine», s. f.) para C++ y *jMonkeyEngine* («jMonkeyEngine», s. f.) para Java.

En varias publicaciones académicas se pueden encontrar testimonios de docentes narrando las experiencias de sus estudiantes que aprendieron a programar usando algunas de las herramientas anteriores para desarrollar videojuegos.

Simões (2016) opina acerca de enseñar Haskell en la cátedra de programación funcional de una carrera universitaria no orientada al diseño de videojuegos, haciendo que los estudiantes utilizaran las bibliotecas gráficas *X3DOM* («X3DOM», s. f.) y *gloss* («Gloss», s. f.) para implementar como proyecto final los videojuegos *LightBot* y *Sokoban*:

Aunque estos proyectos . . . resultaron en algunos productos interesantes, realizados por los mejores estudiantes, la mayor parte de la cursada se dedicaba a implementar funciones simples para procesar cadenas (filtros que, desde un tablero y un movimiento, generaran un nuevo tablero) y solo sobre el final los estudiantes realmente veían algo en movimiento. En consecuencia, cuando los mejores estudiantes llegaban al proyecto final, su motivación aumentaba exponencialmente, pero aquellos que tenían dificultades nunca llegaban a ver nada sofisticado y se desmotivaban. Todavía considero que este enfoque es mejor que el empleado en años anteriores, pero no es el mejor —aunque probablemente sea el mejor posible cuando se aprende un lenguaje como Haskell— (p. 222).

Este mismo autor también opina acerca de enseñar C# en una carrera universitaria orientada al diseño de videojuegos, haciendo que los estudiantes usaran el marco de trabajo *MonoGame* para implementar los videojuegos *Tetris* y *Pac-Man*:

Cuando les enseñé programación a estos estudiantes que estaban altamente motivados para el desarrollo de videojuegos, utilicé un *framework*, y parece que su uso podría ser un buen enfoque para enseñar conceptos básicos de programación orientada a objetos . . . Y esta afirmación no se basa solamente en el *feedback* vistoso que los estudiantes reciben, sino porque al desarrollar un videojuego es fácil enfocarse en los conceptos estándar que se deben aprender (p. 222).

Otros autores que narran sus experiencias con estas herramientas son Camilo Hernandez et al. (2010) —quienes observaron una mejora en el desempeño académico de los estudiantes que desarrollaron una versión del clásico 1942 de Capcom usando el

motor *GameMaker* en un curso universitario cuyo objetivo era enseñar principios básicos de programación sin poner el foco en ningún lenguaje en particular, a pesar de lo cual algo de código en GML se debió escribir— y Vidal Duarte (2013) —quien notó un aumento en la motivación de los estudiantes que usaron la biblioteca *pygame* para desarrollar una versión de Pac-Man en un curso universitario de introducción a la programación con Python—.

# LA LUDIFICACIÓN DEL APRENDIZAJE (*GAMIFICATION*)

Como todos los años, en 2011 la consultora estadounidense Gartner, Inc. evaluó centenas de tecnologías en cuanto a su madurez, sus beneficios empresariales y su futura evolución, y publicó los resultados de sus análisis en forma de ciclos de sobreexpectación (*Hype Cycles*), un tipo de gráfico que resalta el patrón común de exceso de entusiasmo, desilusión y realismo eventual que acompaña a cada nueva tecnología («Gartner’s 2011 Hype Cycle Special Report Evaluates the Maturity of 1,900 Technologies», s. f.).

Específicamente en el ciclo de sobreexpectación para las tecnologías emergentes (*Hype Cycle for Emerging Technologies*), y describiendo una tecnología que recién estaba entrando al pico de expectativas sobredimensionadas con una previsión de 5 a 10 años hasta su adopción generalizada, ese año apareció un nuevo término: *Gamification* (fig. 16).

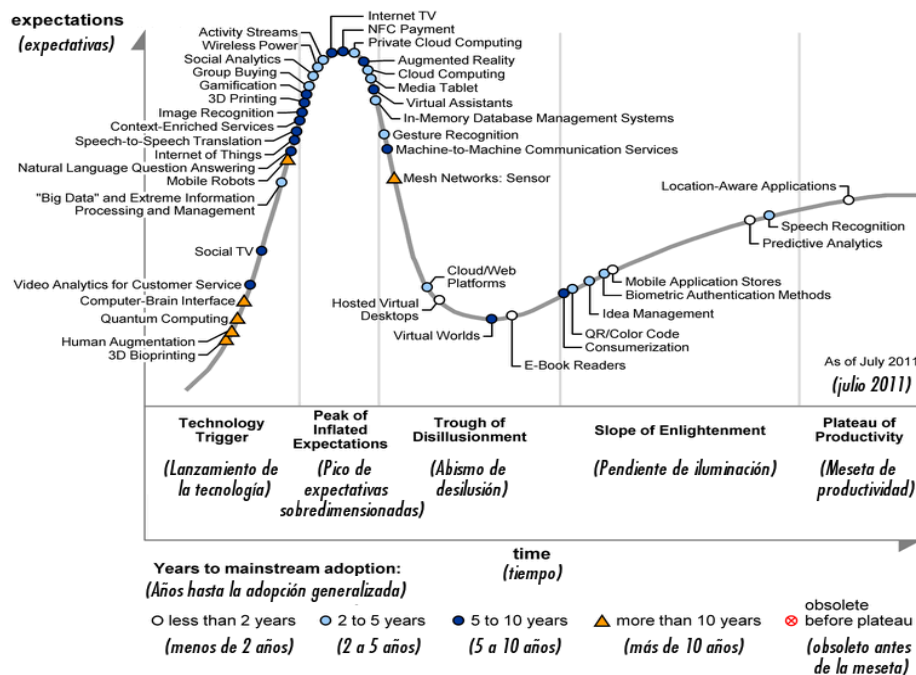


Figura 16. Ciclo de sobreexpectación para las tecnologías emergentes para el año 2011. Adaptado de «Gartner’s 2011 Hype Cycle Special Report Evaluates the Maturity of 1,900 Technologies», s. f. Copyright 2011 por Gartner, Inc.

Dos años más tarde, en el *NMC Horizon Report: Edición sobre Educación Superior 2013* —que es “una fuente de información objetiva que ayuda a . . . comprender fácilmente la influencia de las tecnologías emergentes clave en la educación, y cuándo es probable que pasen a un uso generalizado” (Johnson et al., 2013)—, la *gamification* apareció mencionada como una tecnología a seguir de cerca, con un plazo de adopción generalizada de dos a tres años (p. 5).

Si las previsiones anteriores se cumplieron, actualmente la *gamification* debería estar masivamente difundida —o, en todo caso, debería estar mucho más próxima de estarlo—. Pero, a fin de cuentas, ¿qué es la *gamification*? A continuación se define este concepto.

### **7.1. Definición de la ludificación o *gamification***

El término *gamification* fue acuñado en 2002 por el programador de videojuegos y periodista investigativo británico Nick Pelling, quien en 2003 lo hizo público en la página web de su consultora Conundra Ltd., como afirma en la siguiente entrada de su blog *Funding Startups (€ other impossibilities)*:

En algún momento a fines de 2002 . . . comencé a preguntarme si el tipo de interfaz de usuario de juegos que había estado desarrollando durante tanto tiempo podría usarse para turbocargar todo tipo de transacciones y actividades en dispositivos electrónicos comerciales —video a bordo en vuelos, cajeros automáticos, máquinas expendedoras, teléfonos móviles, etc.—. Como era de esperarse, este fue el momento en que acuñé la palabra deliberadamente fea *gamification*, para referirme a la aplicación de un diseño de interfaz de usuario similar a un juego, con el propósito de hacer que las transacciones electrónicas sean divertidas y rápidas. Tenga en cuenta que en ese momento estaba mucho más interesado en aplicar la idea de *gamification* a los dispositivos electrónicos que a la Web: al igual que ahora, quería *construir* cosas físicas y hacerlas divertidas y efectivas de usar. Por todo ello es que en 2003 (9 de abril) fundé una consultora llamada Conundra Ltd. específicamente con la idea de llevar a la práctica la *gamification*, aunque finalmente la cerré en 2006 (17 de octubre) después de no despertar interés significativo por parte de los clientes (Pelling, 2011).

Casi una década más tarde, en 2011 la palabra *gamification* súbitamente ganó tanta difusión entre los hablantes del idioma inglés, que estuvo entre las finalistas en la elección de la palabra del año llevada a cabo por *Oxford Dictionaries* («SQUEEZED MIDDLE is named Oxford Dictionaries Word of the Year 2011», 2011).

En su libro *For the win: how game thinking can revolutionize your business*, Werbach y Hunter (2012) sostienen que *gamification* es una palabra engorrosa:

Muchos desarrolladores e investigadores de juegos se preocupan —con buena razón— de que esta palabra trivializa las complejidades del diseño del juego. De todos modos, el término se ha establecido . . . El problema es que no existe una definición de *gamification* universalmente aceptada (p. 26).

Desde un principio, entre los hablantes del español no ha habido consenso sobre cómo utilizar esta palabra, lo que llevó a la *Fundación del Español Urgente* (Fundéu BBVA) a publicar la siguiente recomendación, que incluye una breve definición del término:

*Ludificación* podría ser la alternativa adecuada en español para traducir el término inglés *gamification* en lugar del calco *gamificación*. En inglés, *gamification* se refiere a la aplicación de mecánicas de juego a entornos no lúdicos; para denominar esta técnica en español se ha empezado a utilizar el calco *gamificación*, que, sin embargo, no parece adecuado, ya que los derivados de *juego* se forman a partir de la raíz latina *ludus* (*lúdico*, *ludoteca*, *ludópata*, etc.). Se propone entonces utilizar el término *ludificación* tanto como traducción del término inglés, como en un significado amplio referido a las iniciativas que pretenden la motivación a través de la diversión en ambientes no propios para ello (como las empresas, la educación o la salud) («Ludificación, mejor que Gamificación como traducción de Gamification», 2012).

La definición anterior es similar a la propuesta por Deterding, Dixon, Khaled y Nacke (2011), para quienes la ludificación es el uso de elementos del diseño de juegos en contextos no lúdicos (p. 10). Estos autores señalan que el *uso en contextos no lúdicos* es una característica que la ludificación y los *serious games* tienen en común, y resaltan que la diferencia entre ambos conceptos radica en que, en la ludificación, solo se usan *elementos del diseño de juegos*, no juegos enteros —como sí lo son los *serious games*— (p. 13).

Sin embargo, la coincidencia entre ambas definiciones no es total, ya que la recomendación de Fundéu BBVA también menciona la existencia de un significado más amplio del término, referido a la motivación. Este significado sí se encuentra en la definición de Zichermann y Cunningham (2011), quienes definen la ludificación como el proceso de usar el pensamiento lúdico y la mecánica de los juegos para involucrar a los usuarios y resolver problemas (p. 14).

Mientras que la definición anterior solamente menciona a los usuarios, Huotari y Hamari (2012) también hacen mención a los servicios, cuando establecen que la ludificación es un proceso de mejora de un servicio mediante ofrecimientos (*affordances*) para experiencias lúdicas con el fin de apoyar la creación de valor del usuario (p. 19).

## 7.2. Objetivos de la ludificación

Aunque las definiciones anteriores ya incluyen algunas metas de la ludificación (*involucrar a los usuarios y resolver problemas o apoyar la creación de valor del usuario*), estos objetivos pueden ser expresados de una forma mucho más precisa. En su bestseller *Gamify: How Gamification motivates people to do extraordinary things*, el consultor Brian Burke —vicepresidente de investigación de Gartner, Inc.— señala que la ludificación tiene tres propósitos principales: cambiar conductas, desarrollar habilidades e impulsar la innovación para tres públicos meta: clientes, empleados y comunidades de interés (B. Burke, 2014). A continuación se tratan detalladamente estos tres objetivos.

### 7.2.1. Cambio de conductas

Dando como ejemplos de aplicaciones de la ludificación, entre otros, la línea de productos *Nike+* —lanzada en 2006 para motivar a las personas que usan un iPod durante sus actividades físicas—, la solución *F12* —proyectada por el sector de TI de DirecTV para promover que sus empleados compartan sus fracasos, permitiendo así que otros aprendan de los errores ajenos— y los eventos *HeartChase* —organizados desde 2012 por la AHA (*American Heart Association*) para recaudar fondos e impulsar un estilo de vida más saludable, durante los cuales los participantes de esta comunidad de interés compiten en equipos de dos a cinco miembros y registran sus logros en sus teléfonos celulares—, B. Burke (2014) explica que ya sea que el público meta esté constituido por clientes, empleados o una comunidad de interés, todos son personas y responden a estímulos similares para cambiar conductas:

Los hábitos se forman a lo largo del tiempo y, una vez establecidos, pueden ser difíciles de cambiar. El verdadero problema es cómo reconectar nuestros cerebros y reemplazar los viejos hábitos con otros nuevos . . . El cambio es difícil, y la ludificación se puede usar para involucrar y motivar a las personas a realizar cambios de comportamiento . . . positivos en sus vidas.



### 7.2.2. Desarrollo de habilidades

También el desarrollo de habilidades puede facilitarse a través de la ludificación. Utilizando como ejemplos, entre otros, la *Healthcare University* —una solución lanzada en 2013 por la compañía de seguros *Capital Blue Cross* para ayudar a los clientes de seguros de salud en los Estados Unidos a entender mejor los conceptos y beneficios de este sistema—, el juego *Ignite Leadership* —lanzado en 2012 por NTT Data para identificar a los líderes entre sus empleados y desarrollar sus habilidades— y la *Khan Academy* —una organización sin fines de lucro movilizada para desarrollar habilidades en una gran comunidad de interés, en este caso, estudiantes de enseñanza primaria y secundaria—, B. Burke (2014) manifiesta que el aprendizaje ocurre todos los días y en todas partes, pues los humanos están programados para clasificar las experiencias y asimilar el conocimiento:

Casi podemos aprender sin esfuerzo una nueva receta o memorizar un horario de autobuses. Eso es porque la motivación está estrechamente ligada a la habilidad adquirida. El acto de aprender la receta resulta en la satisfacción de una gran comida, y aprender un nuevo horario de autobuses puede reducir los tiempos de espera, por lo que hay una línea de visión clara entre la motivación y el aprendizaje. Pero a veces el aprendizaje es más difícil, particularmente cuando la gratificación del logro se retrasa. Pasar años en la escuela preparándose para una carrera es un trabajo largo y difícil. La capacitación corporativa a menudo es aburrida y no se conecta fácilmente con los objetivos de los empleados. Ya sea que se trate de educación formal, capacitación corporativa o aprendizaje informal, la ludificación puede proporcionar el camino y agregar motivación a las actividades de aprendizaje.

### 7.2.3. Impulso de la innovación

El tercer propósito de la ludificación es impulsar la innovación, algo que B. Burke (2014) ejemplifica citando, entre otros, los casos de la tarjeta de crédito *Barclaycard Ring* —lanzada en 2012 por la compañía de tarjetas de crédito y proveedora de servicios de pago Barclaycard para combinar la ludificación con la cocreación de productos y servicios entre la empresa y los clientes—, el sistema *Idea Street* del DWP (*Department for Work and Pensions*) del Reino Unido —implementado en 2008 y relanzado en 2012 para permitir que los empleados de esta entidad pública pudieran colaborar y, al mismo tiempo, recibir un reconocimiento por su participación en la innovación del sistema de jubilaciones y pensiones británico— y la empresa *Quirky* —que desde 2009 provee la infraestructura y el

apoyo para refinar, producir y distribuir productos surgidos de una amplia comunidad de inventores que envían ideas para nuevos productos de consumo—. Este autor afirma que se acabaron los días en que la responsabilidad de innovar se asignaba a un pequeño grupo de grandes cerebros:

Las organizaciones han pasado a usar un modelo de innovación basado en la colaboración masiva (*crowdsourcing*). Aprovechar la sabiduría de la multitud aporta muchas perspectivas para el desafío de la innovación . . . Pero, ¿cómo innova una multitud? Abandonadas a su propia suerte, las personas optimizarán su condición dentro de las limitaciones del entorno . . . Este fenómeno . . . es todo lo contrario de la planificación central de los grandes cerebros: son las acciones colectivas de muchas personas, limitadas por reglas, quienes se autoorganizan en sistemas complejos y adaptativos, a menudo con resultados radicalmente novedosos . . . La ludificación puede proporcionar la estructura para involucrar, motivar y enfocar las actividades de innovación de la multitud . . . Las soluciones ludificadas les brindan a los jugadores el espacio de juego y crean los objetivos, las reglas, las recompensas y otros aspectos . . ., pero no definen el resultado: los jugadores son libres de innovar dentro de ese espacio.

### 7.3. Elementos usados en la ludificación

Dado que los elementos usados en la ludificación provienen del diseño de juegos, se pueden estudiar a partir del enfoque MDA (*Mechanics, Dynamics, and Aesthetics*), que sus autores Robin Hunicke, Marc LeBlanc y Robert Zubek describen como un enfoque formal para la comprensión de los juegos, que intenta cerrar la brecha entre el diseño y el desarrollo del juego, la crítica del juego y la investigación técnica del juego:

Creemos que esta metodología aclarará y fortalecerá los procesos iterativos de los desarrolladores, los académicos y los investigadores por igual, haciendo que sea más fácil para todos ellos descomponer, estudiar y diseñar una amplia gama de diseños y artefactos de juegos (Hunicke, LeBlanc y Zubek, 2004).

Las tres partes que componen el enfoque MDA (en español: *mecánica, dinámica y estética*) tienen, según estos autores, las siguientes funciones:

La *mecánica* describe los componentes particulares del juego, a nivel de representación de datos y algoritmos. La *dinámica* describe el comportamiento en tiempo de ejecución de la mecánica que actúa sobre las entradas del jugador y las respectivas salidas a lo largo del tiempo. La *estética* describe las respuestas emocionales deseables evocadas en el jugador cuando interactúa con el sistema del juego (Hunicke et al., 2004).

Ramírez Cogollor (2014) afirma que las tres partes del enfoque MDA deben estar perfectamente sincronizadas. Según este autor, “mecánicas, dinámicas y *aesthetics* deben funcionar como un todo, como si de un engranaje perfecto se tratara, pues cualquier fallo en alguna de ellas echaría por tierra la estrategia de gamificación” (p. 80).

Blohm y Leimeister (2013) asocian los elementos del diseño de juegos (mecánica y dinámica) con los motivos del jugador (tabla 27). Por ejemplo, una mecánica de juego como los *rankings* induce una dinámica de *competición* que relaciona las actividades de los usuarios con un grupo de referencia y debería, por lo tanto, satisfacer la búsqueda por *reconocimiento social* (p. 276).

**Tabla 27**

*Elementos del diseño de juegos y motivos*

Elementos del diseño de juegos		Motivo
<b>Mecánica de juego</b>	<b>Dinámica de juego</b>	
Documentación de comportamiento	Exploración	Curiosidad intelectual
Sistemas de puntos, insignias, trofeos	Colección	Logro
<i>Rankings</i> o tablas de clasificación	Competición	Reconocimiento social
Rangos, niveles, puntos de reputación	Adquisición de estatus	Reconocimiento social
Tareas grupales	Colaboración	Intercambio social
Presión temporal, tareas, búsquedas	Desafío	Estimulación cognitiva
Avatares, mundos virtuales, comercio virtual	Desarrollo/organización	Autodeterminación

*Nota:* Basado en “Gamification: Design of IT-based enhancing services for motivational support and behavioral change” por I. Blohm y J. M. Leimeister, 2013, *Business & Information Systems Engineering (BISE)*, 5(4), pp. 275–278. Copyright 2013 por Springer Gabler

Los elementos del diseño de juegos se pueden utilizar en la ludificación de aplicaciones informáticas de muchos tipos —redes sociales (por ejemplo, para incentivar a los usuarios a que mantengan actualizados sus perfiles), plataformas de *e-learning* (por ejemplo, para motivar a los estudiantes y reducir la deserción), *wikis* (por ejemplo, para aumentar las contribuciones)—. No obstante, Deterding et al. (2011) señalan que aunque la abrumadora mayoría de los ejemplos actuales de ludificación son digitales, el término no debe limitarse a la tecnología digital. La convergencia de los medios y la informática ubicua están difuminando cada vez más la distinción entre lo digital y lo no digital, por lo cual los juegos y el diseño del juego son categorías transmediales (p. 11).

En efecto, estos elementos se pueden encontrar además en muchos otros contextos no digitales. Charles *Chuck* Coonradt —apodado el abuelo de la ludificación (Krogue, 2012)— fundó en 1973 la consultora *The Game of Work* y publicó en 1984 el bestseller *The Game of Work: How to enjoy Work as much as Play*. Aunque el término *gamification* aún no había sido acuñado en aquel entonces, no caben dudas de que Coonradt ya tenía como objetivo aplicar al ambiente laboral algunos elementos provenientes de los juegos, como se explica en el sitio web de la empresa:

*The Game of Work* ha tomado las técnicas ganadoras de los deportes y la recreación y las ha aplicado al lugar de trabajo para aumentar la rentabilidad, la productividad, el compromiso de los empleados y el éxito . . . Examinamos el fenómeno de que las personas trabajarán más y gustarán más energía en deportes y otras actividades atléticas que en sus trabajos diarios. ¿Por qué? Porque en los deportes, un jugador tiene una retroalimentación constante sobre cómo lo está haciendo, se conoce el puntaje, y el esfuerzo necesario para ganar está establecido. En el trabajo, la retroalimentación es poco confiable, inconsistente y, a menudo, inexistente. En el trabajo, rara vez se conoce el puntaje o lo que se necesita para ganar («Game of Work», s. f.).

Otro ámbito donde la ludificación se ha hecho presente con bastante fuerza es en los contextos educativos, donde, según Rodríguez y Santiago (2015), “cuando hablamos de Gamificación estamos pensando en motivar a chavales y resulta más fácil conseguir nuestros objetivos cuando conectamos con la tecnología” (p. 11). Sin embargo, estos autores admiten que la ludificación sin tecnología también es posible:

Cuando un profesor juega en clase con sus alumnos al bingo usando la tabla periódica, ha encontrado una dinámica de juego para motivar a sus alumnos y hacer que aprendan los elementos químicos de una forma divertida. Este profesor no es consciente de que está gamificando su clase de Química, y sin incorporar tecnología en el proceso (p. 11).

En la tabla 28 se describen algunos de los elementos que, de acuerdo con Ramírez Cogollor (2014), provienen del diseño de juegos y se utilizan en la ludificación: PBL (*Points, Badges, Leaderboards*), es decir, puntajes, insignias y *rankings*; niveles; retos y desafíos; avatares; así como también monedas para el comercio virtual y bienes virtuales.

Tabla 28

*Elementos del diseño de juegos y descripciones según Ramírez Cogollor (2014)*

Elementos del diseño de juegos	Descripción
PBL ( <i>Points, Badges, Leaderboards</i> ) o puntajes, insignias y <i>rankings</i>	<p><b>Puntajes:</b> La acumulación de puntos es el objetivo inicial de los jugadores, por ello es importante establecer un sistema de puntuación coherente. Cada acción que realice el jugador lleva asociada una cantidad de puntos que se pueden conseguir. Existen distintos tipos de puntos: <i>de experiencia</i> (se ganan a medida que se avanza en el juego), <i>intercambiables</i> (pueden ser intercambiados por algo, como bienes, regalos, etc.), <i>de habilidad</i> (destacan las habilidades específicas del jugador), <i>de karma</i> (sirven para premiar comportamientos), <i>de reputación</i> (dan prestigio). Todo jugador debe conocer perfectamente la manera de ganar puntos.</p> <p><b>Insignias:</b> Son un sistema sencillo y directo de recompensas. Su finalidad es premiar al jugador por hacer cosas concretas. Deben tener un diseño muy atractivo y es recomendable que tengan un color determinado según el nivel del jugador. Es importante que sorprendan: el jugador solo sabrá cómo obtenerlas una vez que las haya conseguido, nunca antes. Se recomienda que tengan un nombre sonoro, que atraiga, para que los jugadores deseen tenerlas.</p> <p><b>Rankings:</b> Son tablas de clasificación que se hacen con la puntuación obtenida por los jugadores. El objetivo es doble: aumentar la competitividad entre los jugadores y motivarlos para que quieran estar en los primeros puestos de la lista. Permiten que cada jugador vea su puesto en relación con el de los demás. Al conocer la forma de ganar puntos, el jugador puede definir una estrategia para lograr llegar a los primeros puestos de la clasificación. Esto es motivador, incluso para los jugadores que aparecen en los puestos de cola, pues querrán salir de abajo lo antes posible.</p>
Niveles	<p>Los niveles indican progresión en el juego. Se alcanzan a través de puntos o determinadas metas, y se implementan para hacer más atractivo el reto de obtener puntos, pues el jugador puede ver un progreso. Se debe intentar que el sistema de niveles no sea lineal, con el objetivo de exigir una mayor progresión en determinados momentos. No se puede exigir lo mismo al jugador experto que al novato. Es importante controlar mucho la curva de dificultad de un nivel a otro: un salto grande entre niveles provocará que el jugador medio no se adapte a ninguno y termine abandonando.</p>
Retos y desafíos	<p>Conviene proponer retos y desafíos (por ejemplo, <i>búsquedas</i>) en determinados momentos para impedir que pasado un tiempo la experiencia se vuelva monótona o aburrida. Los retos y desafíos suelen aportar <i>presión temporal</i> e introducen el factor sorpresa, lo que incita a jugar más.</p>
Avatares	<p>La personalización afecta la forma en que el jugador ve el juego. Es muy común en los juegos de rol que el jugador en lugar de jugar con un personaje predeterminado pueda crear el suyo propio, su alter ego virtual o avatar. Esto provoca que el jugador se involucre más con el juego.</p>
Monedas para el comercio virtual	<p>Además de ganar puntos, en ciertas ocasiones se ganan monedas virtuales, que sirven para comprar — a veces a los otros jugadores — bienes virtuales.</p>
Bienes virtuales	<p>Se trata de bienes y objetos que todos los jugadores desean obtener, bien porque ofrecen unas ventajas determinadas, bien porque sirven como autoexpresión del jugador refrendando su estatus. Son típicos de los juegos de rol: espadas con poderes especiales, objetos exóticos difíciles de encontrar, elementos que sirvan para continuar el juego, escalar posiciones o pasar de fase, etc.</p>

*Nota:* Basado en *Gamificación: mecánicas de juegos en tu vida personal y profesional* por J. L. Ramírez Cogollor, 2014, pp. 81–94. Copyright 2014 por Alfaomega

Kevin Werbach y Dan Hunter —de la escuela de negocios *Wharton School* de la *University of Pennsylvania*— en su libro *For the win: how game thinking can revolutionize your business* detallan, de una manera mucho más breve, 15 componentes de juegos que consideran importantes (tabla 29).

**Tabla 29**

*Elementos del diseño de juegos y descripciones según Werbach y Hunter (2012)*

Elementos del diseño de juegos	Descripción
1. Logros	Objetivos definidos
2. Avatares	Representaciones visuales del personaje de un jugador
3. Insignias	Representaciones visuales de los logros
4. Boss Fights	Desafíos especialmente difíciles en la culminación de un nivel
5. Colecciones	Conjuntos de elementos o insignias para acumular
6. Combate	Una batalla definida, generalmente de corta duración
7. Desbloqueo de contenido	Aspectos disponibles solo cuando los jugadores alcanzan objetivos
8. Regalos	Oportunidades para compartir recursos con otros
9. Tablas de clasificación	Visualizaciones de la progresión y el logro del jugador
10. Niveles	Pasos definidos en la progresión del jugador
11. Puntos	Representaciones numéricas de la progresión del juego
12. Misiones	Desafíos predefinidos con objetivos y recompensas
13. Gráficos sociales	Representación de la red social de los jugadores dentro del juego
14. Equipos	Grupos definidos de jugadores que trabajan juntos para un objetivo común
15. Bienes virtuales	Activos de juego con valor de dinero real o percibido

*Nota:* Basado en *For the win: how game thinking can revolutionize your business* por K. Werbach y D. Hunter, 2012, p. 80, Philadelphia: Wharton Digital Press. Copyright 2012 por Kevin Werbach y Dan Hunter

## 7.4. Un marco de diseño para la ludificación

Así como el caso del pintor —que se enfrenta a un lienzo en blanco y, mezclando colores, lo transforma en un cuadro—, hay muchos otros casos en los que un diseñador empieza con un lienzo vacío y lo transforma en algo que funciona como un todo cohesivo, utilizando de manera sistemática un conjunto de patrones y componentes interrelacionados, es decir, un *marco de diseño*.

Recientemente, Mora Carreño, Riera, González González y Arnedo-Moreno (2015) llevaron a cabo una vasta revisión bibliográfica e identificaron 18 marcos de diseño para la ludificación:

Se realizó una revisión bibliográfica de trabajos indexados principalmente en Google Scholar, Scopus y Web of Knowledge. Las palabras clave fueron *gamification, game, design, framework, models, methods* y *engagement* . . . Se revisaron inicialmente 22 marcos candidatos, sin embargo, solo 18 de ellos cumplieron con los requisitos del estudio. Estos requisitos se refieren a la definición de la estructura formal de un soporte que sirva como base para algo que se está construyendo.

En la figura 17 se presenta el marco de diseño conocido usualmente como 6D, el cual —según los autores anteriores— es el más conocido. Fue ideado y publicado por Werbach y Hunter (2012, p. 86) eligiendo palabras que empiezan con la letra D.

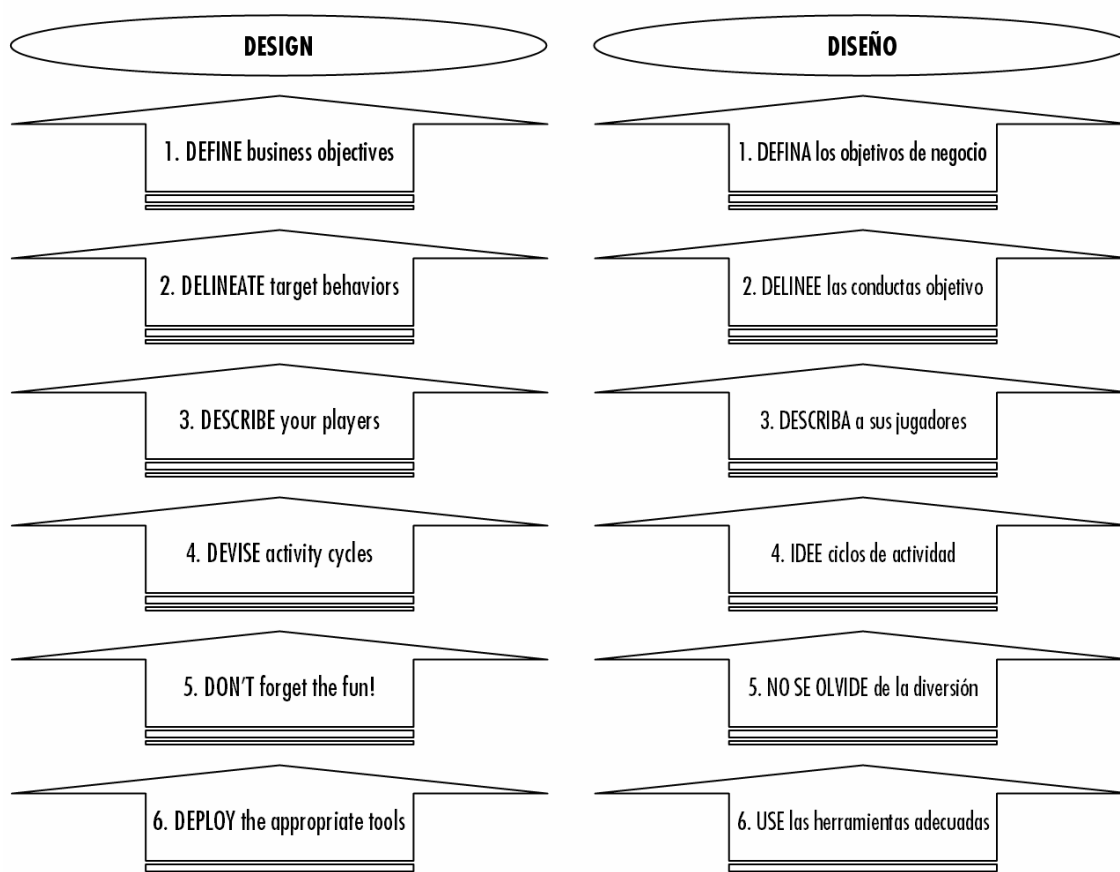


Figura 17. El marco de diseño 6D. Elaboración propia. Basado en *For the win: how game thinking can revolutionize your business* por K. Werbach y D. Hunter, 2012, p. 86, Philadelphia: Wharton Digital Press. Copyright 2012 por Kevin Werbach y Dan Hunter

## 7.5. Tipos de ludificación

Dependiendo del criterio adoptado para clasificar las diversas soluciones que hacen uso de esta estrategia, es posible definir diferentes tipos de ludificación. Considerando su contexto de aplicación, Werbach y Hunter (2012, pp. 20-25) distinguen entre ludificación *interna*, ludificación *externa* y ludificación *social* (de cambio de conducta). En cambio, teniendo en cuenta la motivación en que se basa la solución, Nicholson (2012) diferencia entre ludificación *BLAP* y ludificación *significativa*. A continuación se explican estos diferentes tipos de ludificación.

### 7.5.1. Ludificación interna vs. ludificación externa vs. ludificación social

La *ludificación interna* se utiliza para aumentar la productividad, fomentar la innovación y mejorar la camaradería dentro de las empresas, por lo que se la conoce también como *ludificación empresarial*. Tiene, según Werbach y Hunter (2012), dos características distintivas:

Primero, los jugadores ya son parte de una comunidad definida: la compañía. La empresa sabe quiénes son e interactúan entre sí de forma regular . . . Comparten puntos de referencia, como la cultura corporativa y el deseo de progreso y estatus dentro de la organización . . . Segundo, la dinámica motivacional de la ludificación interactúa con las estructuras de gestión y recompensa existentes . . . como cuando se gana un premio de empleado destacado o la oportunidad de aprender nuevas habilidades (p. 21).

En cambio, la *ludificación externa* se utiliza para involucrar a los clientes en función de objetivos de *marketing*. De acuerdo con estos autores, este tipo de ludificación no es más que otra forma de comercialización apoyada en herramientas motivacionales:

La ludificación aquí es una forma de mejorar las relaciones entre las empresas y los clientes, produciendo una mayor participación, identificación con el producto, mayor lealtad y, en última instancia, mayores ingresos . . . Como una forma de comercialización, la ludificación externa puede aprovechar toda la sofisticación de las prácticas de *marketing* modernas basadas en datos, agregando herramientas para estimular la motivación del cliente (pp. 22-23).

Finalmente, la *ludificación social* busca formar nuevos hábitos beneficiosos en la población, basándose en que un cambio de conducta sostenido requiere de una gran motivación, y aprovechando que los juegos son herramientas de motivación poderosas.



Werbach y Hunter (2012) dan algunos ejemplos de hábitos beneficiosos que la población puede adoptar, motivada por la ludificación social:

Este tipo de ludificación puede involucrar desde animar a las personas a tomar mejores decisiones de salud, como comer mejor o ejercitarse más, hasta rediseñar el aula para que los niños aprendan más mientras disfrutan de la escuela o construir sistemas que ayuden a las personas a ahorrar más dinero para la jubilación . . . En general, estos nuevos hábitos producen resultados sociales deseables: menos obesidad, menores gastos médicos, un sistema educativo más efectivo, mejores decisiones financieras. Los programas de ludificación de cambio de conducta a menudo son patrocinados por organizaciones sin fines de lucro y gobiernos (p. 23).

### **7.5.2. Ludificación BLAP vs. Ludificación significativa**

La *ludificación BLAP*, según Nicholson (2012), es la que utiliza insignias, niveles, *rankings*, logros y puntajes (en inglés: *badges*, *levels*, *leaderboards*, *achievements*, *points*, cuyas iniciales forman el acrónimo BLAP) como una forma de recompensar a alguien por la realización de alguna actividad. Este autor reconoce que, en ciertos casos, este tipo de ludificación puede ser efectivo:

Para algunas situaciones, la gamificación BLAP es una elección apropiada. Las recompensas externas son formas fáciles de atraer a los clientes para que interactúen con un producto y, mientras los sistemas de recompensa externa se mantengan, pueden ser efectivos . . . Si el objetivo es adquirir una habilidad con beneficios en la vida real, como atar un zapato o aprender a usar un programa, entonces la gamificación BLAP puede ser efectiva (p. 1).

No obstante, en un artículo posterior, señala que es preferible que una persona realice una tarea por una motivación intrínseca, en lugar de hacerlo motivada por recompensas externas:

Muchas tareas que se llevan a cabo todos los días no se realizan por una recompensa, sino por alguna otra razón importante para el sujeto. El impulso de hacer algo sin una recompensa externa se conoce como *motivación intrínseca* . . . Realizar tareas por razones intrínsecas pone a alguien en un estado mental más saludable que cumplirlas por recompensas extrínsecas (Nicholson, 2015, p. 2).

Por ello, propone que se utilicen elementos del diseño de juegos para aumentar la motivación intrínseca, en lugar de usarlos como recompensas externas, definiendo así lo que él denomina *ludificación significativa*:

Obtener una buena puntuación es solo una de las razones por las que las personas juegan: los jugadores interactúan con los juegos para explorar la narrativa, tomar decisiones interesantes y jugar con otras personas. Hay otros elementos del diseño de juegos . . . que pueden generar un aumento en la motivación intrínseca. El uso de elementos del diseño de juegos para ayudar a construir la motivación intrínseca y, por lo tanto, el significado en entornos no relacionados con el juego se conoce como *ludificación significativa* (p. 4).

Este autor presenta una lista de seis elementos inspirados en el diseño de juegos y cuyas iniciales forman la palabra inglesa *recipe* (en español: *receta*), y que pueden usarse como una receta para conseguir implementar la ludificación significativa (tabla 30).

**Tabla 30**

*RECIPe: Una receta para lograr la ludificación significativa*

Elemento del diseño de juegos	Instrucción
Reflection (Reflexión)	Ayudar a los participantes a encontrar otros intereses y experiencias pasadas que puedan profundizar la participación y el aprendizaje.
Exposition (Exposición)	Crear, para los participantes, historias que estén integradas con el entorno del mundo real y permitirles crear la suya propia.
Choice (Elección)	Desarrollar sistemas que pongan el poder en las manos de los participantes.
Information (Información)	Usar conceptos del diseño y de la visualización de juegos para permitir a los participantes aprender más sobre el contexto del mundo real.
Play (Juego)	Facilitar la libertad de explorar y fallar dentro de límites predeterminados.
Engagement (Compromiso)	Alentar a los participantes a descubrir y aprender de otros interesados en el entorno del mundo real.

*Nota:* Basado en “A RECIPE for Meaningful Gamification” por S. Nicholson, 2015, *Gamification in Education and Business* editado por T. Reiners y L. C. Wood, p. 5. Copyright 2015 por Springer International Publishing

## 7.6. Posibles problemas de la ludificación

Así como una moneda tiene dos caras, la ludificación también tiene un lado negativo, al cual se asocian posibles problemas como —entre otros— la tendencia a la puntificación (*pointsification*) y su uso como medio de explotación (*exploitationware*). A continuación se tratan estos problemas.

### 7.6.1. Tendencia a la puntificación (*pointsification*)

Según Werbach y Hunter (2012), la forma más fácil de echar a perder el potencial de la ludificación es centrarse demasiado en las recompensas y no lo suficiente en el atractivo de la experiencia (p. 105). Agregarle puntos a cualquier proceso o sistema y

suponer que los usuarios van a interactuar con él solo por amor a la recopilación de puntos, es un problema que Margaret Robertson denomina puntificación (*pointsification*):

Podrás saber cuándo algo ha sido ludificado porque tendrá puntos e insignias. Y este es el meollo del problema . . . Lo que hoy llamamos *ludificación* es, de hecho, el proceso de tomar lo que es menos esencial para los juegos y representarlo como el núcleo de la experiencia . . . La palabra para lo que está pasando en este momento es *puntificación*. Hay cosas que deberían ser puntuadas. Hay cosas que deberían ser ludificadas. Hay cosas que deberían ser ambas cosas. Hay muchas cosas que no deberían ser ninguna de las dos cosas (M. Robertson, 2010).

Kapp (2012) también coincide con lo anterior:

No pienses en la gamificación solo como el uso de insignias, recompensas y puntos; en su lugar, piensa en los elementos motivadores por los que las personas juegan: no es solo por los puntos, sino por el sentido del compromiso, la retroalimentación inmediata, el sentimiento de logro y el éxito de luchar contra un desafío y superarlo (p. xxii).

### **7.6.2. Uso como medio de explotación (*exploitationware*)**

Ian Bogost —un destacado diseñador de juegos y teórico del *Georgia Institute of Technology*— es uno de los principales opositores de la ludificación, a la que considera una farsa, cuando los puntos que los consumidores ganan en el juego no se pueden traducir en beneficios reales:

He sugerido el término *exploitationware* como un nombre más preciso para el verdadero propósito de la ludificación . . . *Exploitationware* captura las intenciones reales de los ludificadores: un juego de estafadores, surgido para sacar provecho de un momento cultural, a través de servicios sobre los que tienen conocimientos cuestionables, para producir resultados destinados a durar solo el tiempo suficiente para llenar sus cuentas bancarias antes de que llegue la próxima tendencia (Bogost, 2011).

En lugar de criticar la ludificación desde la perspectiva del consumo, Patella-Rey (2012) lo hace desde la producción, cuando afirma que la ludificación es un engaño perpetrado por empresas que inducen a las personas a hacer trabajo gratuito.

Werbach y Hunter (2012) advierten que las relaciones entre los trabajadores pueden ponerse cada vez más tensas a medida que el entorno laboral se vuelve más competitivo, y sugieren que al evaluar las iniciativas de ludificación se considere si las ganancias de eficiencia compensan el costo en la insatisfacción de los trabajadores (p. 115).

## 7.7. Implementación de la ludificación para el aprendizaje

En busca de estudios sobre la ludificación en la educación, Dicheva, Dichev, Agre y Angelova (2015) llevaron a cabo una exhaustiva revisión bibliográfica:

Se realizaron búsquedas en las siete principales bases de datos científicas: ACM Digital Library, IEEE Xplore, ScienceDirect, Scopus, Springer Link, ERIC y Google Scholar. Después de buscar en las bases de datos con las palabras clave *gamification*, *gamify* y *gameful* y de eliminar los duplicados, . . . en base a los resúmenes, filtramos por primera vez todas las publicaciones que no estaban relacionadas con la educación . . . Esto fue seguido por una segunda ronda de filtrado en la que, según el texto completo, eliminamos las publicaciones relacionadas con la aplicación de la gamificación para tareas que no estaban directamente relacionadas con el aprendizaje . . . El conjunto resultante contenía 34 artículos que presentaban estudios empíricos para ser analizados y clasificados (p. 77).

En los 34 artículos que finalmente analizaron, encontraron cinco niveles de implementación de la ludificación para el aprendizaje, desde la ludificación sin tecnología digital hasta el desarrollo de *software* propio para la ludificación (tabla 31).

**Tabla 31**

*Niveles de implementación de la ludificación para el aprendizaje*

Nivel	Descripción
Ludificación sin tecnología digital	No se utiliza ninguna plataforma de <i>e-learning</i> u otro <i>software</i> . El profesor administra el juego y publica —regularmente en una pizarra— una tabla de clasificación con los puntajes actualizados.
Uso de tecnología digital no específica para ludificación	Los datos relacionados con el desempeño de los estudiantes se recogen manualmente y se procesan con un programa de computadora —por ejemplo, con <i>Microsoft Excel</i> — para generar una tabla de clasificación con los puntajes actualizados que se publica en la Web o en un grupo de <i>Whatsapp</i> .
Ludificación de plataformas de <i>e-learning</i>	Se agrega al sistema de gestión del aprendizaje o LMS ( <i>learning management system</i> ) un módulo para soportar la ludificación. Por ejemplo, en <i>Moodle</i> («Moodle», s. f.) es posible así dividir las actividades en niveles que se alcanzan mediante puntos de experiencia, encontrar ítems al explorar las actividades, ganar dinero virtual, etc.
Uso de <i>software</i> específico para ludificación	Se utiliza <i>software</i> específico para implementar algún aspecto de la ludificación —por ejemplo: <i>ClassDojo</i> («ClassDojo», s. f.) para otorgar puntos y recompensas, <i>Accredible BadgeBuilder</i> («Accredible BadgeBuilder», s. f.) para crear insignias, <i>Kahoot!</i> («Kahoot!», s. f.) para implementar tareas competitivas, <i>Voki</i> («Voki», s. f.) para crear avatares, <i>rise.global Scorecard</i> («rise.global Scorecard», s. f.) para implementar rankings, etc.— o un <i>software all-in-one</i> (todo en uno) como <i>Classcraft</i> («Classcraft», s. f.).
Desarrollo de <i>software</i> propio	Se desarrolla <i>software</i> propio para implementar la ludificación. Esto es poco justificable si se considera la gran cantidad de <i>software</i> ya existente para este fin.

*Nota:* Basado en “Gamification in Education: A Systematic Mapping Study” por D. Dicheva, C. Dichev, G. Agre y G. Angelova, 2015, *Educational Technology & Society*, 18(3), pp-81-82

## 7.8. La ludificación del aprendizaje de la programación

Son bastante numerosos y sumamente variados los artículos científicos que informan sobre la aplicación de técnicas de ludificación para el aprendizaje de la programación. A continuación se detallan algunos de ellos.

Kumar y Khurana (2012) proponen la utilización de juegos como material didáctico, en reemplazo de los libros y tutoriales disponibles en Internet:

Un estudiante que esté buscando una carrera en el área de la programación debe cursar su carrera en el campo de la informática o de las aplicaciones con un completo conjunto de conocimientos y habilidades en lenguajes de programación como C, C++, C# o Java. Para comenzar a aprender estos lenguajes como principiante o avanzado, es necesario seguir ciertos libros y tutoriales disponibles en Internet. En el proceso basado en la ludificación, sin leer mucho de un libro, los estudiantes pueden obtener la comprensión completa de los conceptos mientras juegan juegos específicamente diseñados por el instructor para este propósito (p. 49).

A primera vista, esta propuesta podría clasificarse como una aplicación del *aprendizaje basado en juegos digitales* o de los *juegos serios*. Sin embargo, en el propio título del artículo se la considera como un caso de ludificación: *Gamification in education – Learn computer programming with fun*. Es que, en realidad, además de incorporar juegos, estos autores proponen reemplazar la nomenclatura tradicional del curso por una nomenclatura ludificada, usando *puntos* en lugar de *créditos*, *niveles* en lugar de *unidades*, *insignias* en lugar de *notas*, etc.:

Para evaluar a los estudiantes-jugadores sobre la base de los conceptos que han aprendido, el instructor puede establecer puntos, así como también modos de dificultad para cada nivel. Los modos de dificultad pueden ser *principiante*, *intermedio*, *avanzado* o *experto* . . . Como factor de estímulo, se puede diseñar una interfaz en el juego para que el estudiante pueda compartir sus puntajes con otros compañeros del mismo curso. Se pueden diseñar insignias como un factor de motivación para aumentar la participación en los juegos. También se pueden agregar mecánicas de juego como logros y bonificaciones (p. 50).

Aunque no se haya llevado a cabo una comparación experimental para evaluar esta propuesta, una encuesta realizada con 207 estudiantes de posgrado de una carrera en desarrollo de aplicaciones indicó una clara preferencia hacia el uso de la ludificación para adquirir conocimientos de programación (p. 46).

Por otro lado, Caton y Greenhill (2013) sugieren entregar premios —para incorporar en el aula el aprendizaje basado en la competencia— y tarjetas de penalización sobre las calificaciones —para fomentar la asistencia (si son emitidas ante una ausencia) u otras conductas deseadas (si son emitidas ante ciertas infracciones)—. Esta propuesta fue evaluada experimentalmente con dos cursos del tercer año de una carrera de producción de juegos, donde los estudiantes trabajaron en equipos —cada uno formado por tres programadores y dos artistas— para desarrollar un prototipo de juego de computadora. (p. 88). Los premios fueron entregados en base al voto popular:

Los premios se otorgaron en las siguientes categorías: *Mejor concepto original*, *Mejor diseño*, *Mejor programación*, *Juego más divertido*, *Mejor cohesión del equipo*, *Juego del año*. El objetivo era fomentar la participación de los equipos y aumentar la motivación mediante la introducción de una competencia amistosa entre los equipos. Los proyectos preseleccionados se exhibieron en un corredor muy visitado y se alentó a todos los estudiantes a votar. A los equipos ganadores les tomaron fotos, y las exhibieron en un pizarrón y en el sitio web de la universidad (p. 92).

Según estos autores, el grupo experimental (n=74) tuvo una mejor asistencia (83%) que el grupo de control (n=62; 76%), y su desempeño tanto individual como grupal también fue mejor (pp. 92-94).

C. Li, Dong, Untch y Chasteen (2013) describen *PeerSpace*, un entorno de aprendizaje en línea desarrollado para mejorar el aprendizaje, fomentando la creación de redes de apoyo entre los estudiantes universitarios matriculados en cursos de informática inicial (p. 72). Posteriormente le agregaron a *PeerSpace* algunos elementos de ludificación:

Varias técnicas de diseño de juegos se han implementado en *PeerSpace* para fomentar la participación en actividades sociales y de aprendizaje: puntos de participación, un sistema de niveles basado en puntos de participación, una barra de progreso, tablas de clasificación, programación colaborativa para la construcción comunitaria y juegos casuales (p. 73).

Muy destacable en este entorno es la *programación colaborativa para la construcción comunitaria*. Se trata de foros con problemas de programación similares a los de la competición ACM-ICPC. Cualquier estudiante puede publicar sus propias soluciones, ver las de los demás, comentarlas y hacer sugerencias. Esto es una excelente manera de que los estudiantes participen, colaboren, mejoren su nivel y, lo que es más importante, aprendan habilidades de programación unos de otros (pp. 74-75).

Estos autores compararon en PeerSpace la participación de dos grupos de estudiantes de la materia *Computer Science II* —dictada en ambos casos por el mismo docente, y basada en el mismo libro de texto y las mismas tareas—. Mientras que el grupo experimental usó el entorno en su totalidad, el grupo de control no utilizó los elementos de ludificación. Como resultado de ello, en PeerSpace la cantidad de publicaciones del grupo experimental fue casi tres veces mayor que la del grupo de control (p. 76).

Mitchell, Danino y May (2013) presentan el 4WC (*Four Week Challenge*), un desafío de un mes de duración —común a todas las carreras de computación—, que tiene lugar antes del comienzo del dictado normal de las materias, apenas los estudiantes ingresan a la universidad. Estos autores resaltan la importancia de formar equipos que tengan todas las mismas probabilidades de ganar:

Los estudiantes trabajan en equipos de seis en una serie de desafíos incrementales. El objetivo final para los equipos es construir con App Inventor un sofisticado juego para teléfonos móviles, comercializarlo en línea y presentarlo en un contexto académico. Además, el módulo en sí se ejecuta como un juego, con equipos que compiten entre sí no solo para construir el mejor juego, sino también para ser el equipo más efectivo. Importante para lograr que los estudiantes adopten la idea del módulo como un juego, fue inculcar la creencia de que los equipos tenían . . . las mismas posibilidades de éxito (p. 394).

Si bien la cantidad de casos de deserción en el primer año de las carreras no cambió significativamente después de la implementación del 4WC, según estos autores sí ocurrió un cambio en el momento en que se producen esos casos: quienes abandonan la carrera lo hacen mucho antes, lo cual sugiere que el 4WC es una ayuda para decidir si la carrera elegida es la correcta. Además, mediante una encuesta respondida por los estudiantes antes y después de participar del 4WC, se determinó que habían disminuido sus preocupaciones financieras, sus preocupaciones sobre hacer amigos, y el miedo al fracaso académico y a la pérdida del apoyo familiar inmediato (p. 399).

A su vez, Orejuela, García, Hurtado y Collazos (2013) plantean el proceso *ChildProgramming-G*, una versión ludificada de su modelo *ChildProgramming*, el cual está basado en la aplicación de prácticas ágiles, colaborativas y cognitivas para el desarrollo de software de calidad, exploradas y empíricamente evaluadas con equipos de niños de edad escolar, entre los 9 y los 11 años de edad (p. 12).

Si bien reconocen que las oportunidades que ofrece la ludificación pueden explotarse incluso en la enseñanza tradicional, Swacha y Baszuro (2013) consideran que —debido a la creciente demanda de educación y capacitación en la era de la información— es en la enseñanza y en el aprendizaje en línea donde esta estrategia debería aplicarse más (p. 122). A causa de las deficiencias que tienen los jueces en línea —que son capaces de verificar y ejecutar los programas escritos por los estudiantes, evaluando su correctitud y su eficiencia (y, a veces, también su estilo de codificación), reconociendo patrones de diseño e incluso detectando casos de plagio, pero que carecen de las capacidades de proporcionar material de instrucción a los estudiantes y de comunicarse con ellos— y los sistemas de gestión del aprendizaje o LMS (*learning management systems*) como, por ejemplo, Moodle —que permite a los docentes distribuir materiales electrónicos de aprendizaje a los estudiantes, hacer un seguimiento de su progreso y comunicarse con ellos individualmente y en grupos, pero que no son capaces de evaluar automáticamente los programas de los estudiantes—, estos autores proponen desarrollar un nuevo sistema de gestión del aprendizaje o LMS especializado en la enseñanza de la programación, con capacidades de ejecución y de evaluación de código, y que emplee mecánicas de ludificación (pp. 123-124).

Por su parte, Agapito, Martínez y Casano (2014) presentan *Xiphias*, el cual que es esencialmente un sistema competitivo que facilita la ludificación de la evaluación de programadores novatos de C++. Para participar en la competición, desde sus terminales los estudiantes inician una sesión en el sistema, utilizando sus nombres de usuario y contraseñas y, a medida que van resolviendo ejercicios, obtienen recompensas en forma de insignias, y se actualiza un *ranking* público —que puede ser proyectado en el pizarrón— (p. 11). La corrección de los ejercicios no es automática y se lleva a cabo desde la cuenta del docente (p. 12). El estatus de los jugadores se obtiene como una combinación entre la cantidad de ejercicios resueltos y el tiempo empleado (p. 13).

González González y Mora Carreño (2015) aplicaron técnicas de ludificación en una materia del tercer año de una carrera universitaria. Primero definieron el objetivo de la actividad, que sería “la realización del diseño y prototipado de una aplicación para móviles sobre Semana de Halloween en La Laguna aplicando correctamente los patrones de



diseño” (p. 32) y establecieron cuatro fases necesarias para lograrlo: análisis y concepto, propuestas de adaptación (Android, iOS o Web adaptativa), prototipado y, por último, presentación (p. 35). Finalmente aplicaron los elementos de juego (mecánicas) a las tareas que componen cada fase:

Las mecánicas pueden ser individuales (puntos, insignias, niveles, restricciones, etc.) o sociales (competición o colaboración, por ejemplo, una tabla de clasificaciones u objetos, entre otras mecánicas). En nuestro caso, hemos utilizado ambos tipos de mecánicas. Las mecánicas individuales consistieron en puntos por participación en la comunidad de Google+, recompensas en forma de golosinas en clase de distinto valor según puntuación obtenida, etc. Las mecánicas sociales fueron de competición entre grupos y de colaboración grupal. Se usaron tablas de clasificaciones y recompensas grupales en forma de objetos virtuales (p. 36).

Estos autores concluyeron que la utilización de la ludificación fue beneficiosa, pues “tuvo una acogida positiva por los estudiantes, aumentando durante el concurso la asistencia a clases presenciales de teoría y problemas, donde se iban viendo los contenidos que debían ir aplicando en el desarrollo de la actividad” (p. 36).

Por otro lado, Hakulinen, Auvinen y Korhonen (2015) incorporaron insignias de logro al sistema de gestión del aprendizaje *TRAKLA2* utilizado en la materia universitaria *Estructuras de Datos y Algoritmos*. En este sistema, los estudiantes resuelven ejercicios interactivos que son corregidos automáticamente. Después realizaron un experimento en el que los estudiantes (n=281) fueron separados aleatoriamente en un grupo experimental y uno de control, con y sin insignias de logro. A los estudiantes del grupo experimental se les otorgaron insignias de logro, por ejemplo, por resolver ejercicios correctamente en el primer intento, hacer ejercicios temprano o resolver todos los ejercicios en una ronda con los puntajes máximos. Sus conclusiones hablan a favor de la ludificación mediante insignias de logros:

Nuestros resultados muestran que las insignias de logros se pueden usar para afectar el comportamiento de los estudiantes. Se observaron diferencias estadísticamente significativas en el tiempo utilizado por ejercicio, número de sesiones, tiempo total y número total de insignias normalizado. Además, la mayoría de los estudiantes informaron estar motivados por las insignias. Con base en nuestros hallazgos, las insignias de logros parecen ser un método prometedor para motivar a los estudiantes y alentar las prácticas de estudio deseadas (p. 18).

A su vez, Campaña et al. (2016) adoptaron un modelo mixto, pues solo ludificaron la parte de evaluación continua de la asignatura universitaria *Fundamentos de Programación*, manteniendo intactos la realización de proyectos y los exámenes de evaluación:

Para motivar al alumnado y favorecer la competitividad desde el punto de vista lúdico, las actividades a realizar en tiempo limitado, como por ejemplo resolución de problemas en parejas o individuales, pueden contener puntos adicionales para aquellos que terminen la tarea en menor tiempo. Esto que en principio puede parecer intrascendente, en la práctica hemos comprobado que activa bastante la dinámica en clase y fomenta el trabajo. En cursos previos a la utilización de esta metodología, las clases de prácticas eran desaprovechadas sistemáticamente por el alumnado y no finalizaban las tareas asignadas. Con la introducción . . . del sistema de puntuación, el ambiente está mucho más enfocado al trabajo (p. 249).

Paiva, Leal y Peixoto de Queirós (2016) presentan la arquitectura y el diseño de *Enki*, un IDE para el aprendizaje de lenguajes de programación en MOOC (*Massive Open Online Courses*). De acuerdo con estos autores, un LMS típico como Moodle puede iniciar este IDE utilizando la API LTI (*Learning Tool Interoperability*). A través de la Web, en *Enki* el estudiante tiene acceso a diferentes herramientas integradas para visualizar tutoriales, resolver ejercicios de programación que se evalúan automáticamente, etc. *Enki* usa varias técnicas para motivar a los estudiantes, como, por ejemplo, la secuenciación de recursos educativos: el contenido del curso (videos, PDF, ejercicios de programación) es revelado progresivamente al estudiante a medida que completa con éxito los ejercicios. Esto es similar a lo que ocurre en un juego, donde los niveles nuevos se desbloquean cuando los anteriores se completan, lo que contribuye al sentido de logro (p. 295). Además, también se utilizan los servicios de ludificación genéricos proporcionados por *Odin* (Leal, Paiva y Peixoto de Queirós, 2015).

Por último, Vera, Moreno, Rodríguez, Vázquez y Valles (2016) describen una plataforma que permite aplicar técnicas de ludificación a la enseñanza universitaria de la programación. En este modelo, el docente prepara enunciados junto con un lote de casos de prueba que se usarán para corroborar automáticamente si el código ingresado por el estudiante mediante un portal web es correcto. El puntaje máximo se va decrementando según los errores detectados y tiempo empleado. La asignación de puntos permite aplicar distintas técnicas de ludificación, como *rankings*, premios, desafíos, insignias, etc. (p. 1).

**SEGUNDA PARTE**

**DESARROLLO EMPÍRICO**

---



## METODOLOGÍA Y DISEÑO DE LA INVESTIGACIÓN

La investigación es una actividad que los seres humanos llevan a cabo con frecuencia. Por ejemplo, al planear un viaje hacia un lugar desconocido, además de determinar las condiciones del transporte y el hospedaje, también es común indagar qué atractivos posee la región, cómo estará el clima durante la estadía, etc.

En particular, la *investigación científica* no es más que tipo específico de investigación, la cual se caracteriza por ser sistemática, empírica y crítica. Según Hernández Sampieri, Fernández Collado y Baptista Lucio (2014):

Que sea *sistemática* implica que hay una disciplina para realizar la investigación científica y que no se dejan los hechos a la casualidad. Que sea *empírica* denota que se recolectan y analizan datos. Que sea *crítica* quiere decir que se evalúa y mejora de manera constante. Puede ser más o menos controlada, más o menos flexible o abierta, más o menos estructurada, pero nunca caótica y sin método (p. xxiv).

A continuación se presenta la metodología empleada en este trabajo, se define el diseño de la investigación, se identifican las variables, se enuncian las hipótesis, y se establecen y describen las muestras y los instrumentos utilizados para la recolección de los datos.

### 8.1. Método

Hasta hace relativamente poco tiempo atrás, solo existían dos aproximaciones a la investigación: el *enfoque cuantitativo* —entre cuyos procesos se encuentran la recolección de datos numéricos y su análisis mediante técnicas estadísticas, para intentar generalizar a un universo los resultados encontrados en una muestra— y el *enfoque cualitativo* —en el cual se recolectan datos no numéricos (como descripciones de situaciones, emociones, conductas, etc.) que se interpretan en base al entendimiento que se tenga sobre ellos, sin pretender generalizar los resultados—.

Sin embargo, en los últimos años, un tercer enfoque se ha establecido como una opción viable en prácticamente todos los ámbitos de investigación: los *métodos mixtos*. De acuerdo con Creswell (2014), la premisa fundamental de esta forma de investigación es que la combinación de los enfoques cualitativo y cuantitativo proporciona una comprensión más completa del problema de investigación que cualquiera de estos enfoques por separado (p. 4).

Hernández Sampieri, Fernández Collado y Baptista Lucio (2014) explican que el desarrollo y la aceptación de los métodos mixtos se deben a que estos, en realidad, no son algo nuevo, y a que “diversos fenómenos han sido abordados desde siempre bajo la óptica mixta de manera natural” (p. 534). Dan como ejemplo de ello la investigación de la escena de un crimen:

Se toman en cuenta técnicas *cuantitativas* (análisis de huellas, sangre y ADN, propiedades químicas de objetos, patrones de salpicadura de la sangre y otras pruebas forenses) y técnicas *cualitativas* (entrevistas a testigos y observación) y distintas clases de evidencia (fotografías, videos, grabaciones de audio, levantamiento de muestras físicas, etcétera) (p. 534).

El sustento epistemológico de los métodos mixtos es el *pragmatismo*, definido como la “postura que consiste en usar el método más apropiado para un estudio específico. Es una orientación filosófica y metodológica, como el positivismo, pospositivismo o constructivismo” (p. 539).

Asimismo, las funciones de los métodos mixtos señaladas por estos autores como justificaciones para su adopción se enumeran en la tabla 32.

Como ya se mencionó en la introducción de este trabajo, el objetivo científico perseguido en esta investigación es evaluar el impacto que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tiene sobre tres dimensiones específicas: la adquisición de los conocimientos específicos de la disciplina, las relaciones interpersonales en el aula y la inteligencia emocional de los estudiantes.

Debido a la naturaleza compleja de este objetivo, se decidió basar esta investigación en métodos mixtos.

Tabla 32

*Justificaciones y razonamientos para el uso de los métodos mixtos*

Justificación	Se refiere a...
1. Triangulación o incremento de la validez	Verificar convergencia, confirmación o correspondencia al contrastar datos cuantitativos y cualitativos, así como a corroborar o no los resultados y descubrimientos en aras de una mayor validez interna y externa del estudio.
2. Compensación	Usar datos y resultados cuantitativos y cualitativos para contrarrestar las debilidades potenciales de alguno de los dos métodos y robustecer las fortalezas de cada uno. Un enfoque puede visualizar perspectivas que el otro no, las debilidades de cada uno son subsanadas por su <i>contraparte</i> .
3. Complementación	Obtener una visión más comprensiva sobre el planteamiento si se emplean ambos métodos, así como un mayor entendimiento, ilustración o clarificación de los resultados de un método sobre la base de los resultados del otro.
4. Amplitud (visión más integral del fenómeno)	Examinar los procesos más holísticamente (conteo de su ocurrencia, descripción de su estructura y sentido de entendimiento).
5. Multiplicidad (diferentes preguntas de indagación)	Responder a un mayor número de diferentes preguntas de investigación y más profundamente.
6. Explicación	Mayor capacidad de explicación mediante la recolección y análisis de datos cuantitativos y cualitativos y su interpretación conjunta.
7. Reducción de incertidumbre	Un método (cuantitativo o cualitativo) puede ayudar a explicar los hallazgos inesperados del otro método.
8. Iniciación	Descubrir contradicciones y obtener nuevas perspectivas y marcos de referencia, y también a la posibilidad de modificar el planteamiento original y resultados de un método con interrogantes y resultados del otro método.
9. Expansión	Extender el rango de la indagación usando diferentes métodos para distintas etapas del proceso investigativo. Un método puede expandir o ampliar el conocimiento obtenido en el otro.
10. Desarrollo	Usar los resultados de un método para ayudar a desplegar o informar al otro método en diversas cuestiones, como el muestreo, los procedimientos, la recolección y el análisis de los datos. Incluso, un enfoque puede proveerle al otro de hipótesis y soporte empírico. Por ejemplo, facilitar el muestreo de casos de un método, apoyándose en el otro.
11. Credibilidad	Al utilizar ambos métodos se refuerza la credibilidad general de los resultados y procedimientos.
12. Contextualización	Proveer al estudio de un contexto más completo, profundo y amplio, pero al mismo tiempo generalizable y con validez externa.
13. Ilustración	Ejemplificar de otra manera los resultados obtenidos por un método.
14. Utilidad	Mayor potencial de uso y aplicación de un estudio (puede ser útil para un mayor número de usuarios).
15. Descubrimiento y confirmación	Usar los resultados de un método para generar hipótesis que serán sometidas a prueba a través del otro método.
16. Diversidad	Obtener puntos de vista variados, incluso divergentes, del fenómeno o planteamiento bajo estudio. Distintas ópticas ( <i>lentes</i> ) para estudiar el problema. Además de lograr una mayor variedad de perspectivas para analizar los datos obtenidos en la investigación (relacionar variables y encontrarles significado).
17. Claridad	Visualizar relaciones <i>encubiertas</i> , las cuales no habían sido detectadas por el uso de un solo método.
18. Argumentación	Consolidar los razonamientos y argumentaciones provenientes de la recolección y análisis de los datos por ambos métodos.
19. Producción metodológica	Generar nuevos métodos de recolección y análisis. Por ejemplo, desarrollar un instrumento para recolectar datos bajo un método, basado en los resultados del otro método, logrando así un instrumento más enriquecedor.

*Nota:* Basado en *Metodología de la investigación* (6ª ed.) por R. Hernández Sampieri, C. Fernández Collado y P. Baptista Lucio, 2014, pp. 538-539, México, D.F.: McGraw-Hill Education. Copyright 2014 por McGraw-Hill / Interamericana Editores, S.A. de C.V.

## 8.2. Diseño de la investigación

De acuerdo con Creswell (2014), hay muchos diseños de métodos mixtos para elegir, y deben considerarse las diferentes posibilidades al decidir cuál es el mejor para determinado estudio (p. 215). Según este autor, el *diseño explicativo secuencial* atrae a individuos con un fuerte historial cuantitativo o desde campos relativamente nuevos a los enfoques cualitativos (p. 234). Las fases del diseño utilizado en este trabajo —que está fuertemente inspirado en el diseño explicativo secuencial— se muestran en la figura 18.

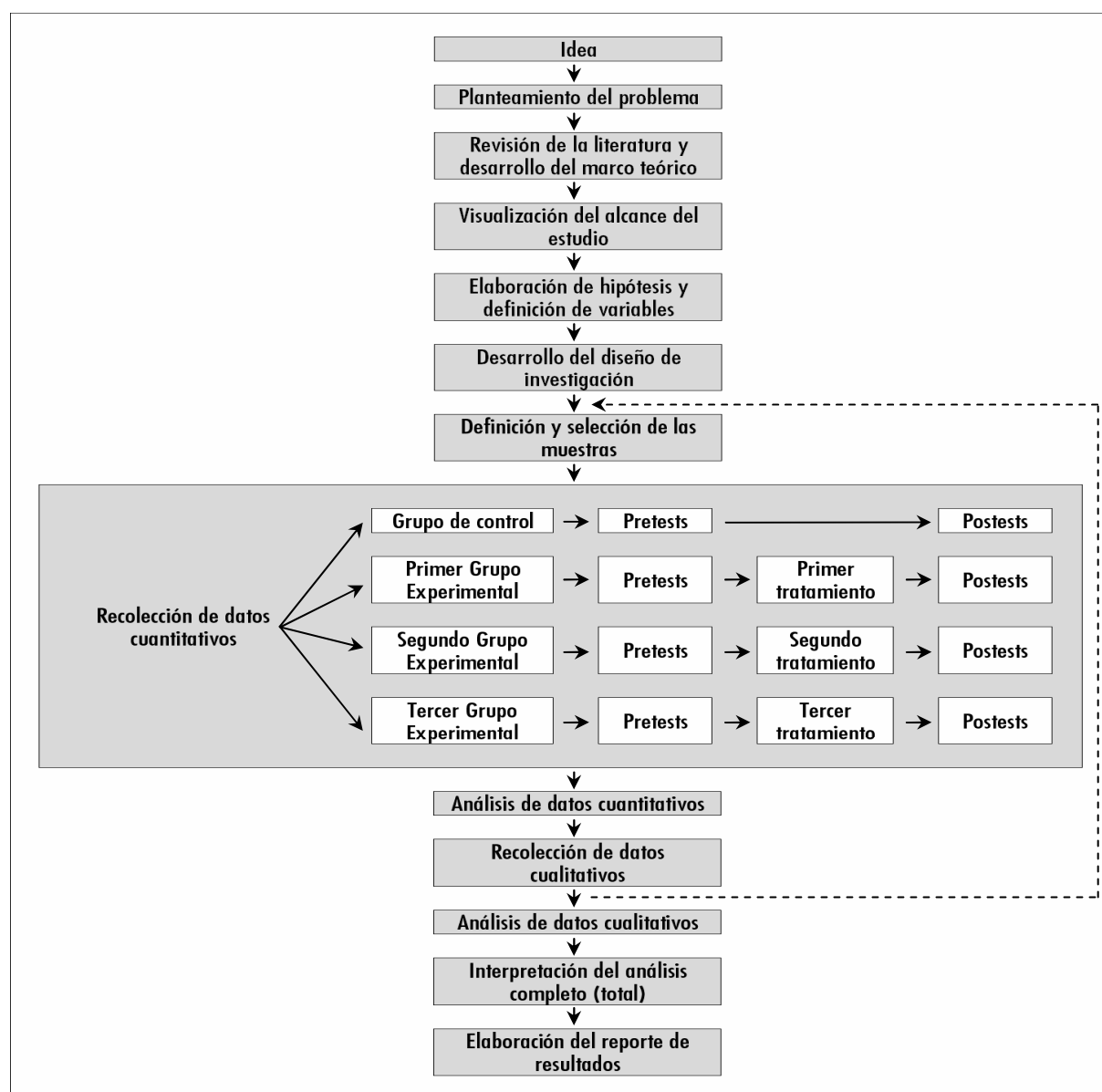


Figura 18. Diseño de la investigación. Elaboración propia. Basado en *Metodología de la investigación* (6ª ed.) por R. Hernández Sampieri, C. Fernández Collado y P. Baptista Lucio, 2014, p. 5, México, D.F.: McGraw-Hill Education. Copyright 2014 por McGraw-Hill / Interamericana Editores, S.A. de C.V.



El propio nombre del *diseño explicativo secuencial* describe con bastante precisión la naturaleza de este modelo:

El *diseño explicativo secuencial* es un diseño en el cual el investigador primero lleva a cabo una investigación cuantitativa, analiza los resultados y luego se basa en ellos para explicarlos más detalladamente mediante una investigación cualitativa. Se lo considera *explicativo* porque los resultados iniciales de los datos cuantitativos se continúan explicando con los datos cualitativos. Se lo considera *secuencial* porque la fase cuantitativa inicial es seguida por la fase cualitativa (Creswell, 2014, p. 15).

En el diseño utilizado en este trabajo, las fases de muestreo, recolección y análisis de datos cuantitativos, y la recolección de datos cualitativos se ejecutaron secuencialmente con cada uno de los cuatro grupos estudiados por separado. Es decir, en lugar de finalizar la recolección y el análisis de datos cuantitativos de *todos* los grupos antes de recolectar los datos cualitativos, estos se empezaron a recoger en cada grupo apenas finalizó la recolección y el análisis de los datos cuantitativos respectivos. La línea punteada en la figura 18 indica que al terminar la recolección de los datos cualitativos de un grupo, es posible recolectar aún los datos cuantitativos y cualitativos de los grupos restantes.

A su vez, en la fase de recolección de datos cuantitativos se utilizó un diseño basado en la aplicación de pretests y postests en tres grupos experimentales y en otro de control, pues “el proceso de comparación, de registro de diferencias o de contrastes es fundamental para la comprobación científica” (Campbell y Stanley, 1973, p. 19). Además, “este es uno de los diseños experimentales más difundidos en la investigación educativa” (p. 93).

En esta investigación, no fue posible definir al azar los grupos experimentales y el grupo de control, ya que son los estudiantes quienes se inscriben voluntariamente en los distintos turnos para cursar la materia observada —Programación II, una asignatura anual que se dicta una vez por semana en la Tecnicatura Superior en Informática Aplicada del Instituto Nacional Superior del Profesorado Técnico, Universidad Tecnológica Nacional, República Argentina—. Consecuentemente, el muestreo de los participantes se llevó a cabo mediante un procedimiento no probabilístico de conveniencia o incidental, por lo cual el diseño debe ser considerado *cuasiexperimental*. A continuación se describen los tres tratamientos aplicados a los grupos experimentales.

### **8.2.1. Primer tratamiento: Desarrollo de un videojuego serio**

La estrategia lúdica que se asoció al primer grupo experimental fue el *Aprendizaje mediante el desarrollo de videojuegos*. En consecuencia, el tratamiento que se aplicó entre los pretests y los postests consistió en que los estudiantes de este grupo se organizaran en ocho equipos y que cada equipo desarrollara —en su mayor parte fuera del horario de clase— un videojuego cuya temática girara en torno a los propios conceptos de la programación orientada a objetos vistos hasta ese momento, es decir, se trataría de juegos serios, ya que su propósito sería principalmente educativo.

La experiencia tuvo lugar en junio de 2016 y duró un mes (el cuarto mes de la cursada de la materia Programación II). En esta materia, los estudiantes aprenden los elementos básicos del lenguaje Java y algunas metodologías de análisis y diseño orientados a objetos. Para realizar ciertas tareas auxiliares como, por ejemplo, generar la documentación de los sistemas o sus diagramas, se usan varias herramientas sencillas (JavaDoc, EasyUML, etc.). No obstante, desarrollar un programa complejo como un videojuego con la edición estándar de Java sería un problema prácticamente imposible de resolver. Por ello, para llevar a cabo esta experiencia se decidió utilizar el framework libGDX, que es específico para el desarrollo de videojuegos.

### **8.2.2. Segundo tratamiento: Uso de videojuegos serios**

Al segundo grupo experimental se asoció otra de las estrategias lúdicas: el *Aprendizaje mediante la utilización de videojuegos*. Consecuentemente, el tratamiento que se aplicó entre los pretests y los postests consistió en que durante la primera media hora de cada una de las cuatro clases que duró la experiencia (en junio de 2017, o sea, en el cuarto mes de la cursada de la materia Programación II), los estudiantes formaran espontáneamente equipos y jugaran con los siete juegos serios desarrollados el año anterior por el primer grupo experimental. Al contrario del primer tratamiento, esta experiencia tuvo lugar solamente durante el horario de clase.

Los videojuegos serios utilizados pertenecen a los más diversos géneros: *The Road of Destiny* (un viaje por una ruta llena de obstáculos, durante el cual se deben adquirir combustible y repuestos para el auto con el dinero ganado al responder

correctamente las preguntas sobre la programación orientada a objetos), *The Java Hacker* (juego en 3D donde un agente secreto ingresa a un edificio y utiliza terminales en las cuales debe responder preguntas para ir desbloqueando las distintas puertas hacia la oficina donde debe finalizar su misión), *Fighting Questions Club* (juego de lucha en el cual se dan o reciben golpes según se respondan bien o mal, respectivamente, ciertas preguntas), *Programación Verdura* (un cocinero debe responder preguntas atrapando verduras que representan conceptos de la programación orientada a objetos), *The Java Mania Game* (un arquero debe lanzar sus flechas en dirección a blancos móviles y acertar los conceptos correctos), *El Tumbalatas* (juego en 3D donde debe lanzarse una pelota hacia una pila de latas para derribarlas, a fin de obtener una pregunta que, en caso de ser respondida correctamente, permite avanzar al siguiente nivel) y *El Escuerzo Erudito* (un sapo debe cruzar una avenida evitando los vehículos, saltar sobre troncos llevados por la corriente de un río y llegar a la otra orilla en el lugar identificado con la respuesta correcta). Algunos de estos juegos se ilustran en la figura 19.

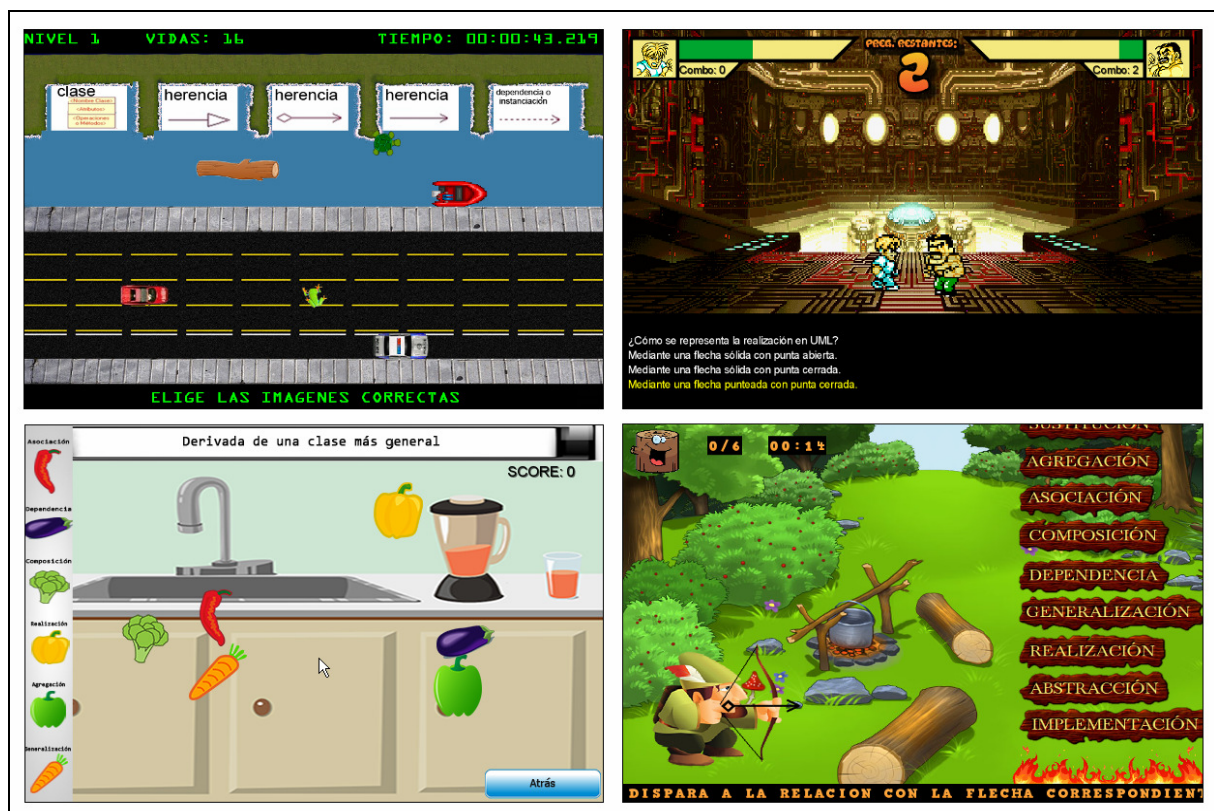






















Figura 19. Algunos de los videojuegos serios desarrollados por el primer grupo experimental y utilizados por el segundo grupo experimental. Elaboración propia

### 8.2.3. Tercer tratamiento: Ludificación del aprendizaje

La estrategia lúdica que se asoció al tercer grupo experimental fue la *Ludificación del aprendizaje (Gamification)*. En consecuencia, el tratamiento que se aplicó entre los pretests y los postests consistió en que los estudiantes de este grupo se organizaran en 12 equipos y participaran en una competición de programación que tuvo lugar durante 4 semanas entre abril y mayo de 2018. La experiencia se proyectó siguiendo el marco de diseño 6D (figura 17) y se decidió encuadrarla en el segundo nivel de implementación de la ludificación para el aprendizaje —*uso de tecnología digital no específica para ludificación*— (tabla 31), con los datos del desempeño de los estudiantes recogidos manualmente y procesados con Microsoft Excel para generar en la Web (en la plataforma *groups.io*) una tabla de clasificación con los puntajes actualizados (figura 20).

Ranking Actual		Ranking Anterior	Variación	Equipo	13/4 Des. 1	20/4 Nombre	20/4 Des. 2	4/5 Des. 3	18/5 Des. 4	Puntaje total	Insignias acumuladas
1	1	0	Effective Shava	95	34	85	95	100	409	 	
2	2	0	inteligentes	90	14	100	80	75	359	  	
3	3	0	Ada Lovelace	60	42	80	100	70	352	   	
4	6	▲2	Castle Galaxy	80	30	95	50	90	345		
5	4	▼1	Los reincidentes	75	50	65	85	50	325		
6	7	▲1	losCasteadores	65	18	75	75	85	318	  	
7	9	▲2	Nuevo código	50	46	70	60	80	306		
	10	▲3	Noname	55	6	60	90	95		  	
9	8	▼1	KOLOSO	100	10	50	70	55	285		
10	11	▲1	Desestructurados	70	22	55	55	65	267		
11	5	▼6	Object null reference	85	26	90	65	0	266		
12	12	0	Los Tabuladores	45	38	45	45	60	233		




 Ascensos sucesivos  
 Cima alcanzada  
 Ascenso veloz

Figura 20. Tabla de clasificación de los 12 equipos que constituyeron el tercer grupo experimental. Elaboración propia

Cada una de las sesiones de la competición ocurridas durante las clases consistió en resolver cuatro ejercicios extraídos del libro *OCA Java SE 8 Programmer I Exam Guide (Exams 1Z0-808)* publicado en 2017 por Kathy Sierra y Bert Bates. Cuanto más rápido se resolvían correctamente estos ejercicios, más puntos se podían ganar. También se llevaron a cabo desafíos fuera del horario de clases y se otorgaron insignias (*Ascenso veloz*, *Cima alcanzada*, etc.) además de los puntos. Se decidió no otorgar un único premio al equipo ganador, sino ofrecer a todos los equipos la posibilidad de elegir el tema del examen parcial según el orden de clasificación alcanzado al finalizar la competición. Así, el equipo ganador podría elegir entre los 12 temas disponibles, el segundo tendría 11 temas a elección, y así sucesivamente, hasta que el último equipo de la tabla debería conformarse con el último tema disponible. Tradicionalmente, el examen parcial consiste en desarrollar un sistema informático para cierto ámbito, y poder elegir un ámbito conocido por el equipo es visto, sin duda, como una ventaja nada despreciable.

### 8.3. Variables

Hernández Sampieri, Fernández Collado y Baptista Lucio (2014) definen el concepto de *variable* como “una propiedad que puede fluctuar y cuya variación es susceptible de medirse u observarse” (p. 105). Asimismo, las clasifican en *variables independientes* y *variables dependientes*:

Al hablar de hipótesis, a las supuestas *causas* se les conoce como *variables independientes* y a los *efectos* como *variables dependientes*. Únicamente es posible hablar de variables independientes y dependientes cuando se formulan hipótesis causales o hipótesis de la diferencia de grupos, siempre y cuando en estas últimas se explique cuál es la causa de la diferencia supuesta en la hipótesis (p. 111).

En esta investigación se han utilizado, por un lado, las cuatro variables independientes *estrategia lúdica*, *grupo etario*, *sexo* y *trabajo con programación* y, por otra parte, 30 variables dependientes medidas en el pretest y en el postest (ocho de ellas referidas a la adquisición de los conocimientos específicos de la programación orientada a objetos, dos a las relaciones interpersonales en el aula y 20 a la inteligencia emocional de los estudiantes). Todas estas variables se definen, explican y categorizan a continuación en las tablas 33 (variables independientes) y 34 (variables dependientes).

Tabla 33

*Variables independientes utilizadas en este trabajo*

Variable	Tipo	Valor
Estrategia lúdica	Catagórica politómica	0: Grupo control; 1: Primer tratamiento; 2: Segundo tratamiento; 3: Tercer tratamiento
Rango etario	Catagórica politómica	1: 19 a 22 años; 2: 23 a 26 años; 3: 27 a 30 años; 4: Más de 30 años
Sexo	Catagórica dicotómica	1: Hombre; 2: Mujer
Trabajo con programación	Catagórica dicotómica	1: Sí; 2: No

Nota: Elaboración propia

Tabla 34

*Variables dependientes utilizadas en este trabajo*

Variable	Dimensión	Tipo	Valor
Capacidad de identificar las relaciones entre clases en UML	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Capacidad de implementar las relaciones entre clases en Java	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Capacidad de ordenar por rango los tipos primitivos en Java	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Capacidad de nombrar las clases de envoltorio en Java	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Capacidad de ordenar los calificadores de acceso que regulan el <i>encapsulamiento</i> en Java	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Capacidad de usar la <i>abstracción</i> para diseñar en UML un diagrama de clases que aplique la <i>herencia</i>	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Capacidad de describir la utilidad de la sobrecarga en Java	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Capacidad de reconocer e interpretar el <i>polimorfismo</i> en Java	Adquisición de conocimientos de POO	Catagórica dicotómica	1: Sí; 2: No
Distancia social media propia	Relaciones interpersonales en el aula	Cuantitativa continua	1 a 5
Distancia social media del grupo	Relaciones interpersonales en el aula	Cuantitativa continua	1 a 5
Faceta Adaptabilidad	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Asertividad	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Autoestima	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Automotivación	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Conciencia Social	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Control de la Impulsividad	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Empatía	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Expresividad Emocional	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Felicidad	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Gestión de la Emoción	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Gestión del Estrés	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Optimismo	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Percepción Emocional	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Regulación Emocional	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Faceta Relaciones	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Factor Autocontrol	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Factor Bienestar	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Factor Emocionalidad	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Factor Sociabilidad	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7
Factor Global de Inteligencia Emocional	Inteligencia emocional de los estudiantes	Cuantitativa continua	1 a 7

Nota: Elaboración propia. (Obs.: POO es la abreviatura de *Programación Orientada a Objetos*)

Debido a que las variables dependientes fueron medidas dos veces para cada participante del estudio (una vez en el pretest y otra en el postest), las mismas fueron codificadas para poder distinguir la medición, como se muestra en la tabla 35.

**Tabla 35**
*Codificación de las variables dependientes utilizadas en este trabajo*

Variable	Código	
	Pretest	Postest
Capacidad de identificar las relaciones entre clases (P001)	PRE-P001	POS-P001
Capacidad de implementar las relaciones entre clases en Java (P002)	PRE-P002	POS-P002
Capacidad de ordenar por rango los tipos primitivos en Java (P003)	PRE-P003	POS-P003
Capacidad de nombrar las clases de envoltorio (P004)	PRE-P004	POS-P004
Capacidad de ordenar los calificadores de acceso que regulan el <i>encapsulamiento</i> en Java (P005)	PRE-P005	POS-P005
Capacidad de usar la <i>abstracción</i> para diseñar en UML un diagrama de clases que aplique la <i>herencia</i> (P006)	PRE-P006	POS-P006
Capacidad de describir la utilidad de la sobrecarga de constructores y métodos en Java (P007)	PRE-P007	POS-P007
Capacidad de reconocer e interpretar el <i>polimorfismo</i> en Java (P008)	PRE-P008	POS-P008
Distancia social media propia (DSP)	PRE-DSP	POS-DSP
Distancia social media del grupo (DSG)	PRE-DSG	POS-DSG
Faceta Adaptabilidad	PRE-Faceta Adaptabilidad	POS-Faceta Adaptabilidad
Faceta Asertividad	PRE-Faceta Asertividad	POS-Faceta Asertividad
Faceta Autoestima	PRE-Faceta Autoestima	POS-Faceta Autoestima
Faceta Automotivación	PRE-Faceta Automotivación	POS-Faceta Automotivación
Faceta Conciencia Social	PRE-Faceta Conciencia Social	POS-Faceta Conciencia Social
Faceta Control de la Impulsividad	PRE-Faceta Control de la Impulsividad	POS-Faceta Control de la Impulsividad
Faceta Empatía	PRE-Faceta Empatía	POS-Faceta Empatía
Faceta Expresividad Emocional	PRE-Faceta Expresividad Emocional	POS-Faceta Expresividad Emocional
Faceta Felicidad	PRE-Faceta Felicidad	POS-Faceta Felicidad
Faceta Gestión de la Emoción	PRE-Faceta Gestión de la Emoción	POS-Faceta Gestión de la Emoción
Faceta Gestión del Estrés	PRE-Faceta Gestión del Estrés	POS-Faceta Gestión del Estrés
Faceta Optimismo	PRE-Faceta Optimismo	POS-Faceta Optimismo
Faceta Percepción Emocional	PRE-Faceta Percepción Emocional	POS-Faceta Percepción Emocional
Faceta Regulación Emocional	PRE-Faceta Regulación Emocional	POS-Faceta Regulación Emocional
Faceta Relaciones	PRE-Faceta Relaciones	POS-Faceta Relaciones
Factor Autocontrol	PRE-Factor Autocontrol	POS-Factor Autocontrol
Factor Bienestar	PRE-Factor Bienestar	POS-Factor Bienestar
Factor Emocionalidad	PRE-Factor Emocionalidad	POS-Factor Emocionalidad
Factor Sociabilidad	PRE-Factor Sociabilidad	POS-Factor Sociabilidad
Factor Global de Inteligencia Emocional	PRE-Factor Global de IE	POS-Factor Global IE

*Nota:* Elaboración propia

Las variables que miden la adquisición de los conocimientos de la programación orientada a objetos (POO1, POO2, POO3, POO4, POO5, POO6, POO7 y POO8) pueden valer 1 ( $S\hat{i}$ ) —cuando la medida del pretest fue 2 ( $No$ ) y la del postest fue 1 ( $S\hat{i}$ )—, 2 ( $No$ ) —cuando la medida del pretest fue 2 ( $No$ ) y la del postest fue 2 ( $No$ )— o pueden ser un caso descartado —cuando la medida del pretest fue 1 ( $S\hat{i}$ ), ya que no se habría podido medir la eficacia del tratamiento si el conocimiento ya se poseía con anterioridad.

#### 8.4. Hipótesis

Considerando que el objetivo de esta investigación es evaluar el impacto que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tiene sobre tres dimensiones específicas —la adquisición de los conocimientos específicos de la disciplina, las relaciones interpersonales en el aula y la inteligencia emocional de los estudiantes—, se plantearon las cinco hipótesis que se presentan a continuación:

- H1) La *estrategia lúdica* y las variables referidas a la *adquisición de los conocimientos específicos de la programación orientada a objetos* están relacionadas, es decir, presentan una asociación significativa.
- H2) Entre los grupos formados en función de la *estrategia lúdica* empleada, las variables referidas a las *relaciones interpersonales en el aula* medidas en el postest difieren significativamente, una vez eliminada la influencia del pretest sobre los resultados del postest.
- H3) Dentro de cada grupo formado en función de la *estrategia lúdica* empleada, las variables referidas a las *relaciones interpersonales en el aula* presentan diferencias significativas entre los valores medidos en el pretest y en el postest.
- H4) Entre los grupos formados en función de la *estrategia lúdica* empleada, las variables referidas a la *inteligencia emocional* de los estudiantes medidas en el postest difieren significativamente, una vez eliminada la influencia del pretest sobre los resultados del postest.
- H5) Dentro de cada grupo formado en función de la *estrategia lúdica* empleada, las variables referidas a la *inteligencia emocional* de los estudiantes presentan diferencias significativas entre los valores medidos en el pretest y en el postest.



## 8.5. Muestras

Debido a que en el Instituto Nacional Superior del Profesorado Técnico (*Universidad Tecnológica Nacional*) la cantidad de estudiantes que anualmente cursan la materia Programación II de la Tecnicatura Superior en Informática Aplicada es bastante reducida (en el turno noche, que es el de mayor concurrencia, el número de inscriptos casi nunca llega a 30 estudiantes por año), no se emplearon para el muestreo procedimientos probabilísticos.

### 8.5.1. Muestras para recolectar datos cuantitativos

El muestreo de los participantes de la fase de recolección de datos cuantitativos se llevó a cabo mediante un procedimiento no probabilístico de conveniencia o incidental. Entre junio de 2016 y mayo de 2018 se obtuvieron pretests y postests de estudiantes ( $n=97$ ) que estaban cursando Programación II en los tres cursos correspondientes a los grupos experimentales y también en la misma materia dictada en otros turnos (grupo de control). En la tabla 36 se resumen las características de los estudiantes que participaron de esta fase. Para facilitar la comparación de los grupos, también se han elaborado los gráficos que se muestran en la figura 21.

**Tabla 36**

*Características de los estudiantes que participaron de la fase de recolección de datos cuantitativos*

		Sexo		Rango etario (años)				Trabajo con programación		Total
		Hombre	Mujer	19 a 22	23 a 26	27 a 30	Más de 30	Si	No	
Estrategia 0	Recuento	23	4	8	9	3	7	7	20	27
	%	85,2%	14,8%	29,7%	33,3%	11,1%	25,9%	25,9%	74,1%	27,8%
1	Recuento	20	5	4	6	4	11	7	18	25
	%	80,0%	20,0%	16,0%	24,0%	16,0%	44,0%	28,0%	72,0%	25,8%
2	Recuento	22	3	2	10	6	7	7	18	25
	%	88,0%	12,0%	8,0%	40,0%	24,0%	28,0%	28,0%	72,0%	25,8%
3	Recuento	16	4	1	6	9	4	7	13	20
	%	80,0%	20,0%	5,0%	30,0%	45,0%	20,0%	35,0%	65,0%	20,6%
Total	Recuento	81	16	15	31	22	29	28	69	97
	%	83,5%	16,5%	15,5%	32,0%	22,6%	29,9%	28,9%	71,1%	100,0%

*Nota:* Elaboración propia

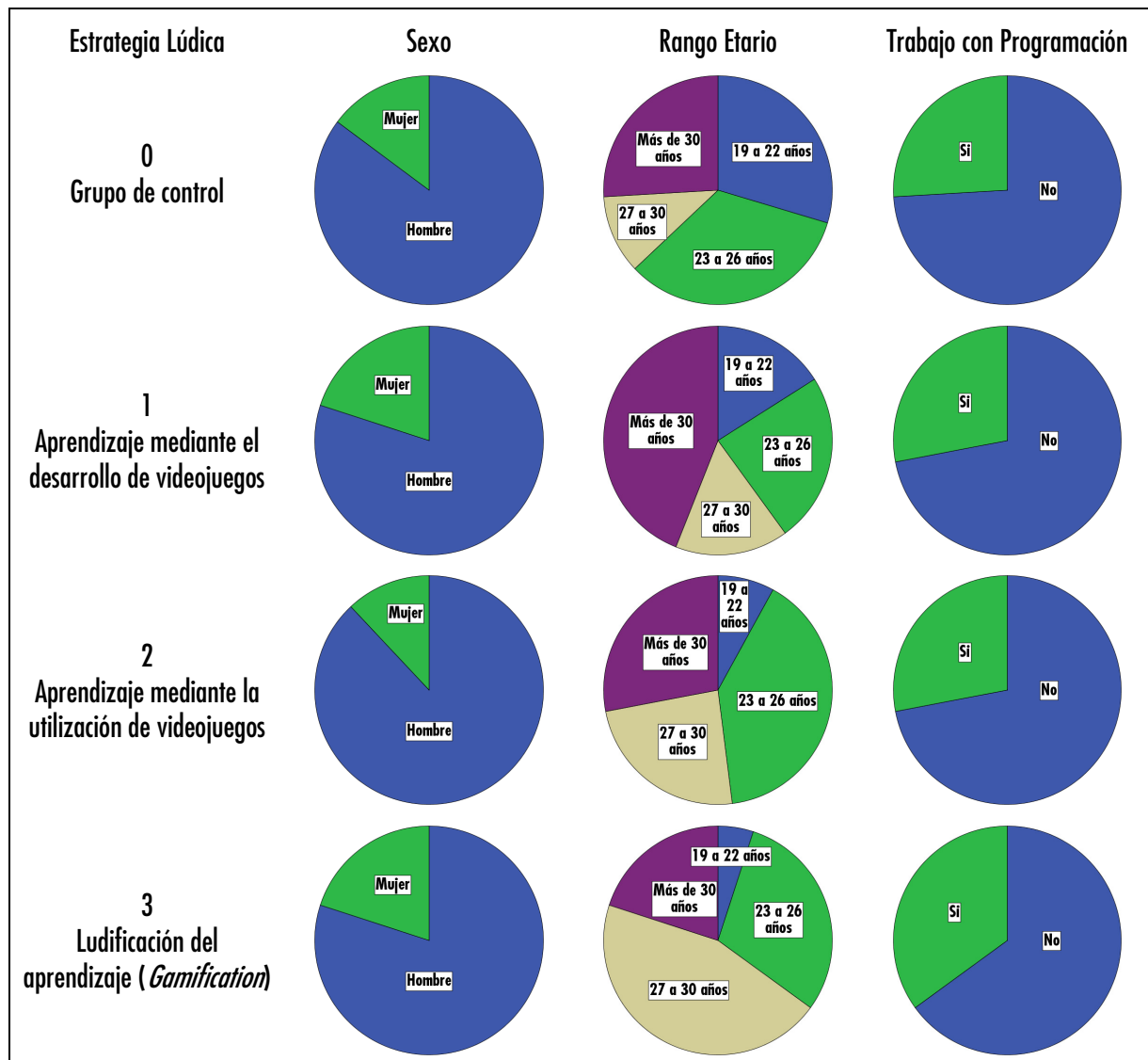


Figura 21. Características de los estudiantes que participaron de la fase de recolección de datos cuantitativos

### 8.5.2. Muestras para recolectar datos cualitativos

En la fase de recolección de datos cualitativos también se utilizó un procedimiento no probabilístico para la selección de los participantes: la muestra fue autoseleccionada, ya que los participantes se propusieron voluntariamente, respondiendo a una invitación. Entre junio de 2016 y mayo de 2018 se entrevistó a estudiantes ( $n=21$ ) que estaban cursando Programación II en los tres cursos correspondientes a los grupos experimentales y también en la misma materia dictada en otros turnos (grupo de control). En la tabla 37 se resumen las características de los estudiantes que participaron de esta fase. Para facilitar la comparación de los grupos, también se han elaborado los gráficos que se muestran en la figura 22.

Tabla 37

Características de los estudiantes que participaron de la fase de recolección de datos cualitativos

Estrategia		Sexo		Rango etario (años)				Trabajo con programación		
		Hombre	Mujer	19 a 22	23 a 26	27 a 30	Más de 30	Si	No	Total
0	Recuento	6	0	1	3	1	1	4	2	6
	%	100,0%	0,0%	16,7%	50,0%	16,7%	16,7%	66,7%	33,3%	28,6%
1	Recuento	5	0	1	1	2	1	4	1	5
	%	100,0%	0,0%	20,0%	20,0%	40,0%	20,0%	80,0%	20,0%	23,8%
2	Recuento	4	1	0	2	2	1	0	5	5
	%	80,0%	20,0%	0,0%	40,0%	40,0%	20,0%	0,0%	100,0%	23,8%
3	Recuento	4	1	0	2	3	0	2	3	5
	%	80,0%	20,0%	0,0%	40,0%	60,0%	0,0%	40,0%	60,0%	23,8%
Total	Recuento	19	2	2	8	8	3	10	11	21
	%	90,5%	9,5%	9,5%	38,1%	38,1%	14,3%	47,6%	52,4%	100,0%

Nota: Elaboración propia

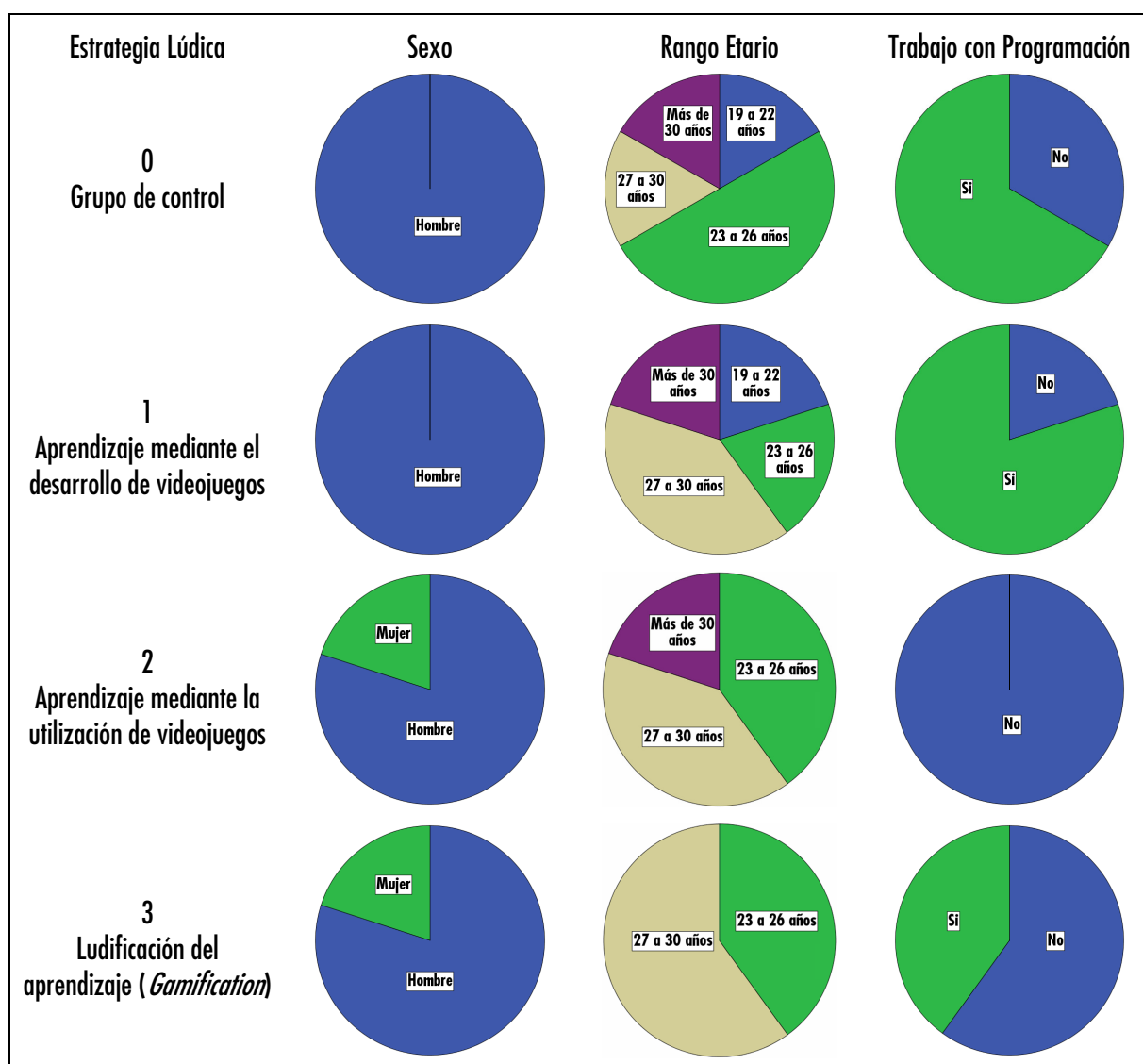


Figura 22. Características de los estudiantes que participaron de la fase de recolección de datos cualitativos

## 8.6. Instrumentos

En esta investigación se utilizaron instrumentos de dos tipos: instrumentos para la recolección de datos e instrumentos para el análisis de datos. A continuación se detallan ambos tipos de instrumentos.

### 8.6.1. Instrumentos para la recolección de datos

En la primera fase (cuantitativa) se recogieron los datos tanto del pretest como del postest. En el pretest se aplicaron los siguientes instrumentos:

- *Escala de la Distancia Social en el Aula*, adaptada (figura 23);
- TEIQue (*Trait Emotional Intelligence Questionnaire*), en español (anexo 1);
- Pretest de conceptos de programación orientada a objetos (figura 24).

En el postest se aplicaron los siguientes instrumentos:

- *Escala de la Distancia Social en el Aula*, adaptada (figura 23);
- TEIQue (*Trait Emotional Intelligence Questionnaire*), en español (anexo 1);
- Postest de conceptos de programación orientada a objetos (figura 25).

Con las respuestas de todas las planillas de la *Escala de la Distancia Social en el Aula* se construyeron sociomatrices para el pretest y el postest —similares a la de la figura 7— para cada uno de los grupos experimentales y para el grupo de control.

A partir de las respuestas de todas las planillas del TEIQue se calcularon para el pretest y el postest las facetas y los factores emocionales utilizando el mismo algoritmo usado por el sistema *online* disponible en el sitio web del *London Psychometric Laboratory* («Psychometric Lab», s. f.) que Petrides dirige en el *University College London*.

Las respuestas de las ocho preguntas del pretest y del postest de conceptos de programación orientada a objetos se evaluaron considerándolas correctas (1: *Sí*) o incorrectas (2: *No*), y esos fueron —respectivamente— los valores de las variables PRE-POO1, PRE-POO2, PRE-POO3, PRE-POO4, PRE-POO5, PRE-POO6, PRE-POO7 y PRE-POO8 (pretest), y POS-POO1, POS-POO2, POS-POO3, POS-POO4, POS-POO5, POS-POO6, POS-POO7 y POS-POO8 (postest), que se utilizaron para calcular POO1, POO2, POO3, POO4, POO5, POO6, POO7 y POO8.

INSTITUTO NACIONAL SUPERIOR DEL PROFESORADO TÉCNICO  
UNIVERSIDAD TECNOLÓGICA NACIONAL

**ESCALA DE LA DISTANCIA SOCIAL EN EL AULA**

Nombre: ..... Fecha: .....

No nos agradan todos nuestros compañeros de la misma manera. Algunos nos simpatizan más que otros. Puede haber algunas personas que no nos agraden en absoluto.

La lista de verificación que se encuentra más abajo te dará una forma de decir cuán próxima es tu relación con otros estudiantes en tu curso. Debajo de cada nombre listado en la grilla, pon una cruz en el espacio correspondiente a la frase que mejor describe tu sentimiento hacia esa persona.

*Cuando llegues a tu propio nombre, pon una raya. Nadie en la sala verá esta hoja, solo tu profesor.*

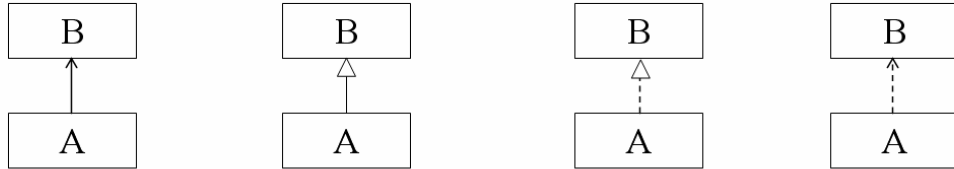
<i>Nombre</i>	1. Somos amigos/os.	2. Siempre trabajamos juntos/os.	3. A veces trabajamos juntos/os.	4. Solo la/lo saludo.	5. No la/lo conozco.

Figura 23. Escala de la Distancia Social en el Aula. Adaptado de *Understanding group behavior of boys and girls* por Cunningham et al., 1951, p. 405. Copyright 1951 por Columbia University

Nombre y Apellido: ..... Edad: .....  
 Si trabaja actualmente, de qué: ..... ¿Programa en el trabajo? .....

**(Pre) Test de Conceptos**

1) Coloque el nombre de las siguientes relaciones:



a) ..... b) ..... c) ..... d) .....

2) ¿Cómo se programan en Java las relaciones anteriores?

- a) Se programa .....
- b) Se programa .....
- c) Se programa .....
- d) Se programa .....

3) Ordene ascendentemente los siguientes tipos primitivos según el tamaño del rango que abarcan:  
 long - double - int - byte - float - short .....

4) ¿Cómo se llaman las *wrapper classes* correspondientes a boolean, char, double, int y byte?  
 .....

5) Numere los siguientes miembros (atributos/métodos), del más accesible [1] al menos accesible [4]:

[ ] miembro <code>private</code>	[ ] miembro <code>protected</code>
[ ] miembro <code>public</code>	[ ] miembro sin calificador de acceso

6) Dibuje en UML el diagrama de clases para el enunciado:

*El agua, la leche, el yogur y el ron son bebidas.  
 La leche y el yogur son lácteos. El ron es inflamable.*

7) ¿Para qué sirve la *sobrecarga* de métodos?  
 .....  
 .....

8) Escriba y explique la salida del siguiente programa:

<pre>public class Main {     public static void main(String[] args){         Clase1 a = new Clase2();         Clase1 b = new Clase1();         Clase2 c = new Clase2();         System.out.println(a.ampliar(2));         System.out.println(a.ampliar(2.0));         System.out.println(b.ampliar(2));         System.out.println(b.ampliar(2.0));         System.out.println(c.ampliar(2));         System.out.println(c.ampliar(2.0));     } }</pre>	<pre>public class Clase1 {     public int ampliar(int x) {         return x * 3;     }     public int ampliar(double x){         return (int) x * 4;     } }</pre>	<pre>public class Clase2     extends Clase1 {     public int ampliar(int x) {         return x * 2;     } }</pre>
---	--	---

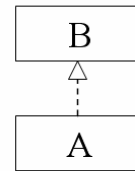
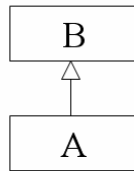
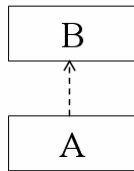
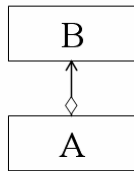
.....  
 .....

Figura 24. Pretest de conceptos de programación orientada a objetos. Elaboración propia

Nombre y Apellido: .....

**(Pos) Test de Conceptos**

1) Coloque el nombre de las siguientes relaciones:



a) ..... b) ..... c) ..... d) .....

2) ¿Cómo se programan en Java las relaciones anteriores?

- a) Se programa .....
- b) Se programa .....
- c) Se programa .....
- d) Se programa .....

3) Ordene ascendentemente las siguientes clases según el tamaño del rango numérico que abarcan:  
Long - Double - Byte - Float - Short

4) ¿Cómo se llaman los tipos envueltos por las *wrapper classes* Boolean, Integer y Character?

5) Numere los siguientes miembros (atributos/métodos), del más accesible [1] al menos accesible [4]:

<input type="checkbox"/> miembro protected	<input type="checkbox"/> miembro sin calificador de acceso
<input type="checkbox"/> miembro private	<input type="checkbox"/> miembro public

6) Dibuje en UML el diagrama de clases para el enunciado:

*El pulpo, el calamar, el fugu y la merluza son animales marinos. El pulpo y el calamar son moluscos. El fugu y la merluza son peces. El fugu es tóxico.*

7) ¿Para qué sirve la *sobrecarga* de constructores?

8) Escriba y explique la salida del siguiente programa:

<pre>public class Main {     public static void main(String[] args){         Clase1 a = new Clase2();         Clase1 b = new Clase1();         Clase2 c = new Clase2();         System.out.println(a.ampliar(2));         System.out.println(a.ampliar(2.0));         System.out.println(b.ampliar(2));         System.out.println(b.ampliar(2.0));         System.out.println(c.ampliar(2));         System.out.println(c.ampliar(2.0));     } }</pre>	<pre>public class Clase1 {     public int ampliar(int x) {         return x * 4;     }     public int ampliar(double x){         return (int) x * 2;     } }</pre>	<pre>public class Clase2     extends Clase1 {     public int ampliar(int x) {         return x * 3;     } }</pre>
---	--	---

.....  
.....

Figura 25. Postest de conceptos de programación orientada a objetos. Elaboración propia

En la segunda fase (la cualitativa) se realizaron entrevistas semiestructuradas. Estas se centran en tres áreas principales, según la percepción que los estudiantes tienen sobre su compromiso con la aplicación de la estrategia de enseñanza-aprendizaje en la cursada, sobre el conocimiento del contenido que adquirieron durante la aplicación de la estrategia de enseñanza-aprendizaje en la cursada y sobre sus emociones hacia la aplicación de estrategias lúdicas de enseñanza-aprendizaje y su motivación. En la figura 26 se muestra la guía con las preguntas utilizadas en las entrevistas.

<p style="text-align: center;"><b>ESTUDIANTES QUE CURSARON PROGRAMACIÓN II SIN ESTRATEGIAS LÚDICAS (TRADI)</b></p> <ol style="list-style-type: none"><li>1. Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?</li><li>2. Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?</li><li>3. ¿Pensás que sería posible cursar esta materia utilizando juegos? ¿Cómo? ¿Tomaría mucho tiempo y esfuerzo?</li></ol> <p style="text-align: center;"><b>ESTUDIANTES QUE DESARROLLARON VIDEOJUEGOS EN PROGRAMACIÓN II (DESVJ)</b></p> <ol style="list-style-type: none"><li>1. Mirando hacia atrás, ¿cuál de los proyectos desarrollados en Programación II te impresionó más? ¿Por qué?</li><li>2. Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?</li><li>3. Volvamos al proyecto del videojuego, ¿te gustó desarrollarlo con tu grupo? ¿Le dedicaron mucho tiempo y esfuerzo? ¿Podrías decirme más sobre cómo administraron el proceso? ¿Qué los motivó?</li></ol> <p style="text-align: center;"><b>ESTUDIANTES QUE USARON VIDEOJUEGOS EN PROGRAMACIÓN II (USOVJ)</b></p> <ol style="list-style-type: none"><li>1. Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?</li><li>2. Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?</li><li>3. Pensando en los videojuegos que jugaste al cursar esta materia, ¿te gustó utilizarlos con tu grupo? ¿Les dedicaron mucho tiempo y esfuerzo?</li></ol> <p style="text-align: center;"><b>ESTUDIANTES QUE CURSARON LA ASIGNATURA PROGRAMACIÓN II LUDIFICADA (LUDIF)</b></p> <ol style="list-style-type: none"><li>1. Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?</li><li>2. Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?</li><li>3. Pensando en la primera parte de la materia que cursaste con una dinámica de juego, ¿te gustó cursarla así? ¿Opinás que te aportó algo?</li></ol>
---

Figura 26. Guía de preguntas para estructurar las entrevistas. Elaboración propia

### 8.6.2. Instrumentos para el análisis de datos

El *software* Microsoft Excel se utilizó para cargar las sociomatrices y calcular los índices sociométricos, así como también las facetas y factores del TEIQue. Para los análisis estadísticos se usó el *software* IBM SPSS Statistics 23 y G\*Power 3.1.9.2. El análisis de los datos cualitativos se llevó a cabo con el *software* ATLAS.ti 6.2.



En este capítulo se presentan los resultados de los análisis cuantitativo y cualitativo de los datos. En el caso del análisis cuantitativo —orientado a conservar o rechazar las hipótesis planteadas—, este comenzó con un análisis exploratorio de las variables dependientes (estadística descriptiva) y finalizó con el contraste de las hipótesis. El objetivo del análisis exploratorio fue conocer y exponer las características generales de los datos. En el caso del análisis cualitativo, este tuvo como objetivo encontrar —en los conceptos expresados por los estudiantes en las entrevistas— indicios que reforzaran los resultados del análisis cuantitativo. A continuación se presentan los diferentes análisis y sus resultados.

### 9.1. Análisis cuantitativo: Estadística descriptiva

Primeramente se llevó a cabo el análisis exploratorio de las variables categóricas dicotómicas —las variables que miden la adquisición de los conocimientos de la programación orientada a objetos— en función de la estrategia lúdica, obteniéndose los estadísticos descriptivos que se muestran a continuación en la tabla 38.

**Tabla 38**

*Estadísticos descriptivos de las variables dependientes (categóricas dicotómicas)*

Variable	Estrategia Lúdica	Frecuencia absoluta			Frecuencia relativa		
		Sí	No	Total	Sí	No	Total
P001	0	8	17	25	32,0%	68,0%	100,0%
	1	9	14	23	39,1%	60,9%	100,0%
	2	16	8	24	66,7%	33,3%	100,0%
	3	4	15	19	21,1%	78,9%	100,0%
P002	0	1	25	26	3,8%	96,2%	100,0%
	1	2	22	24	8,3%	91,7%	100,0%
	2	3	20	23	13,0%	87,0%	100,0%
	3	0	19	19	0,0%	100,0%	100,0%

*Nota:* Elaboración propia

Tabla 38 (continuación)

Estadísticos descriptivos de las variables dependientes (categóricas dicotómicas)

Variable	Estrategia Lúdica	Frecuencia absoluta			Frecuencia relativa		
		Sí	No	Total	Sí	No	Total
P003	0	7	19	26	26,9%	73,1%	100,0%
	1	6	16	22	27,3%	72,7%	100,0%
	2	8	10	18	44,4%	55,6%	100,0%
	3	3	13	16	18,8%	81,3%	100,0%
P004	0	13	11	24	54,2%	45,8%	100,0%
	1	14	9	23	60,9%	39,1%	100,0%
	2	14	9	23	60,9%	39,1%	100,0%
	3	4	10	14	28,6%	71,4%	100,0%
P005	0	2	21	23	8,7%	91,3%	100,0%
	1	5	16	21	23,8%	76,2%	100,0%
	2	2	16	18	11,1%	88,9%	100,0%
	3	2	15	17	11,8%	88,2%	100,0%
P006	0	6	20	26	23,1%	76,9%	100,0%
	1	7	14	21	33,3%	66,7%	100,0%
	2	10	8	18	55,6%	44,4%	100,0%
	3	5	7	12	41,7%	58,3%	100,0%
P007	0	0	17	17	0,0%	100,0%	100,0%
	1	3	16	19	15,8%	84,2%	100,0%
	2	3	11	14	21,4%	78,6%	100,0%
	3	1	12	13	7,7%	92,3%	100,0%
P008	0	13	5	18	72,2%	27,8%	100,0%
	1	13	7	20	65,0%	35,0%	100,0%
	2	5	6	11	45,5%	54,5%	100,0%
	3	5	9	14	35,7%	64,3%	100,0%

Nota: Elaboración propia

Después se realizó el análisis exploratorio de las variables cuantitativas continuas —las variables que miden las relaciones interpersonales en el aula y la inteligencia emocional de los estudiantes— en función de la estrategia lúdica, obteniéndose los estadísticos descriptivos que se muestran a continuación en la tabla 39.

Tabla 39

Estadísticos descriptivos de las variables dependientes (cuantitativas continuas)

Variable	Estrategia	Estadístico						
		Media	Error estándar	Mediana	Mín.	Máx.	Varianza	SD
PRE-DSP	0	4,57550	0,048123	4,61538	3,885	4,962	0,063	0,250053
	1	4,60500	0,056166	4,62500	4,042	5,000	0,079	0,280830
	2	4,43000	0,090386	4,33333	3,625	5,000	0,204	0,451931
	3	4,28158	0,146708	4,36842	3,053	5,000	0,430	0,656097
POS-DSP	0	4,48006	0,040196	4,50000	3,962	4,808	0,044	0,208864
	1	4,36167	0,074018	4,41667	3,542	4,917	0,137	0,370091
	2	4,28167	0,103338	4,25000	3,000	5,000	0,267	0,516689
	3	4,16053	0,147708	4,39474	3,000	4,895	0,436	0,660571

Nota: Elaboración propia. (Obs.: SD = desviación estándar)

Tabla 39 (continuación)

Estadísticos descriptivos de las variables cuantitativas continuas

Variable	Estrategia	Estadístico						
		Media	Error estándar	Mediana	Min.	Máx.	Varianza	SD
PRE-DSG	0	4,57550	0,042346	4,61538	4,269	4,885	0,048	0,220037
	1	4,60500	0,046530	4,62500	4,208	5,000	0,054	0,232650
	2	4,43000	0,090738	4,37500	3,708	5,000	0,206	0,453689
	3	4,28158	0,109491	4,36842	3,526	4,895	0,240	0,489658
POS-DSG	0	4,48006	0,041991	4,53846	4,077	4,808	0,048	0,218190
	1	4,36167	0,052137	4,41667	3,792	4,750	0,068	0,260686
	2	4,28167	0,090212	4,16667	3,542	5,000	0,203	0,451060
	3	4,16053	0,119194	4,34211	3,474	4,842	0,284	0,533052
PRE-Faceta Adaptabilidad	0	4,25926	0,194812	4,33333	2,000	6,444	1,025	1,012270
	1	4,82222	0,140253	4,88889	3,556	6,667	0,492	0,701263
	2	4,62222	0,197828	4,66667	3,000	6,556	0,978	0,989139
	3	4,13889	0,190903	3,94444	3,000	6,333	0,729	0,853746
POS-Faceta Adaptabilidad	0	4,45679	0,183404	4,33333	1,889	6,333	0,908	0,952995
	1	4,90667	0,171375	4,88889	3,444	6,889	0,734	0,856877
	2	4,58222	0,221970	4,44444	2,556	6,778	1,232	1,109851
	3	4,22222	0,223389	4,11111	2,667	5,889	0,998	0,999025
PRE-Faceta Asertividad	0	4,66667	0,179102	4,77778	2,778	6,778	0,866	0,930643
	1	4,76444	0,143059	4,77778	3,333	6,222	0,512	0,715294
	2	4,69778	0,163491	4,66667	3,111	6,222	0,668	0,817454
	3	4,19444	0,212641	4,00000	2,778	5,889	0,904	0,950958
POS-Faceta Asertividad	0	4,63374	0,166568	4,66667	3,111	6,444	0,749	0,865512
	1	4,75556	0,153826	4,66667	3,778	6,556	0,592	0,769132
	2	4,65778	0,148368	4,77778	2,889	5,889	0,550	0,741842
	3	4,46667	0,182147	4,33333	3,222	6,333	0,664	0,814584
PRE-Faceta Autoestima	0	4,73401	0,197398	4,72727	2,000	6,727	1,052	1,025711
	1	4,97818	0,153138	5,00000	3,364	6,455	0,586	0,765690
	2	4,83636	0,142489	5,00000	3,000	5,727	0,508	0,712444
	3	4,51818	0,189216	4,68182	2,455	5,909	0,716	0,846198
POS-Faceta Autoestima	0	4,74747	0,191796	4,63636	2,364	6,636	0,993	0,996604
	1	5,03636	0,156494	5,00000	3,636	6,455	0,612	0,782470
	2	5,01818	0,159199	5,00000	2,818	6,273	0,634	0,795995
	3	4,76364	0,206927	4,81818	2,455	6,182	0,856	0,925404
PRE-Faceta Automotivación	0	4,29630	0,168577	4,60000	2,500	6,000	0,767	0,875953
	1	4,78800	0,122109	4,80000	3,900	6,100	0,373	0,610546
	2	4,55600	0,189656	4,40000	2,400	6,300	0,899	0,948279
	3	4,33500	0,249238	4,30000	2,300	6,300	1,242	1,114628
POS-Faceta Automotivación	0	4,32222	0,216244	4,30000	1,800	6,300	1,263	1,123639
	1	4,82800	0,162534	4,70000	3,000	6,400	0,660	0,812670
	2	4,51200	0,226592	4,30000	2,400	6,400	1,284	1,132961
	3	4,55500	0,238579	4,65000	2,700	6,300	1,138	1,066956
PRE-Faceta Conciencia Social	0	4,59933	0,197299	4,45455	2,364	6,818	1,051	1,025194
	1	4,60000	0,152211	4,63636	3,182	5,909	0,579	0,761053
	2	4,75273	0,158464	5,00000	2,818	6,091	0,628	0,792318
	3	4,16818	0,230765	4,31818	2,091	6,000	1,065	1,032013
POS-Faceta Conciencia Social	0	4,47811	0,165094	4,54545	2,455	6,182	0,736	0,857854
	1	4,85091	0,181658	4,81818	3,182	7,000	0,825	0,908288
	2	4,69818	0,167441	4,72727	2,455	6,273	0,701	0,837203
	3	4,39091	0,232532	4,40909	2,636	6,818	1,081	1,039917

Nota: Elaboración propia. (Obs.: SD = desviación estándar)

Tabla 39 (continuación)

*Estadísticos descriptivos de las variables cuantitativas continuas*

Variable	Estrategia	Estadístico						
		Media	Error estándar	Mediana	Min.	Máx.	Varianza	SD
PRE-Faceta Control de la Impulsividad	0	4,60905	0,184840	4,55556	2,778	6,222	0,922	0,960458
	1	4,52889	0,194289	4,66667	2,778	6,222	0,944	0,971444
	2	4,49778	0,196064	4,55556	2,222	6,667	0,961	0,980321
	3	4,48889	0,182467	4,55556	2,667	5,556	0,666	0,816019
POS-Faceta Control de la Impulsividad	0	4,66255	0,181490	4,66667	3,000	6,444	0,889	0,943052
	1	4,56889	0,203803	4,77778	2,889	7,000	1,038	1,019017
	2	4,25778	0,169933	4,44444	2,333	6,000	0,722	0,849667
	3	4,71111	0,168200	4,83333	3,222	5,889	0,566	0,752211
PRE-Faceta Empatía	0	4,96296	0,209345	5,00000	2,778	7,000	1,183	1,087789
	1	4,82667	0,153971	4,88889	3,556	6,333	0,593	0,769854
	2	5,13333	0,150855	5,00000	3,556	6,333	0,569	0,754275
	3	4,95556	0,216355	4,88889	3,222	6,667	0,936	0,967570
POS-Faceta Empatía	0	4,94239	0,193668	5,00000	3,222	7,000	1,013	1,006329
	1	4,92000	0,213488	5,00000	2,889	7,000	1,139	1,067438
	2	5,24444	0,167160	5,00000	3,778	6,889	0,699	0,835799
	3	5,12778	0,176655	5,22222	3,444	6,222	0,624	0,790025
PRE-Faceta Expresividad Emocional	0	4,25185	0,253655	4,10000	2,200	6,800	1,737	1,318032
	1	4,54400	0,213235	4,60000	2,600	6,800	1,137	1,066177
	2	4,32400	0,197254	4,30000	2,200	6,800	0,973	0,986272
	3	3,40500	0,239899	3,45000	1,700	4,800	1,151	1,072859
POS-Faceta Expresividad Emocional	0	4,25556	0,245259	4,20000	1,900	7,000	1,624	1,274403
	1	4,53200	0,209898	4,30000	3,000	7,000	1,101	1,049492
	2	4,40800	0,204770	4,50000	2,400	6,400	1,048	1,023849
	3	3,84500	0,229011	3,85000	2,400	5,200	1,049	1,024168
PRE-Faceta Felicidad	0	5,66667	0,155617	5,62500	3,875	6,875	0,654	0,808608
	1	5,97000	0,156172	6,12500	4,000	7,000	0,610	0,780858
	2	5,59500	0,168214	5,62500	3,125	6,750	0,707	0,841068
	3	4,66875	0,286091	4,75000	1,500	6,500	1,637	1,279440
POS-Faceta Felicidad	0	5,61111	0,130801	5,50000	4,500	6,875	0,462	0,679661
	1	6,01000	0,184690	6,37500	4,125	7,000	0,853	0,923450
	2	5,64500	0,174505	5,62500	3,625	7,000	0,761	0,872526
	3	4,85625	0,305820	5,06250	2,250	6,875	1,871	1,367669
PRE-Faceta Gestión de la Emoción	0	4,96296	0,206639	5,00000	2,667	6,778	1,153	1,073730
	1	4,45778	0,193763	4,33333	2,889	6,222	0,939	0,968814
	2	4,68889	0,157788	4,88889	3,222	5,667	0,622	0,788941
	3	4,42222	0,191367	4,11111	3,444	6,667	0,732	0,855818
POS-Faceta Gestión de la Emoción	0	4,90535	0,175402	5,00000	3,000	6,778	0,831	0,911415
	1	4,63556	0,192185	4,55556	2,556	6,222	0,923	0,960924
	2	4,84444	0,170572	4,88889	3,556	6,444	0,727	0,852858
	3	4,51667	0,220059	4,44444	3,000	6,111	0,969	0,984133
PRE-Faceta Gestión del Estrés	0	4,59259	0,162243	4,60000	2,900	6,400	0,711	0,843038
	1	4,48000	0,162070	4,30000	2,600	6,000	0,657	0,810350
	2	4,58800	0,167861	4,70000	2,800	6,800	0,704	0,839305
	3	4,15000	0,133870	4,30000	2,900	5,100	0,358	0,598683
POS-Faceta Gestión del Estrés	0	4,47407	0,138736	4,60000	3,200	6,300	0,520	0,720893
	1	4,60400	0,192222	4,60000	3,100	6,900	0,924	0,961110
	2	4,59200	0,176155	4,40000	3,200	6,600	0,776	0,880776
	3	4,56500	0,180828	4,70000	3,200	6,100	0,654	0,808686

Nota: Elaboración propia. (Obs.: SD = desviación estándar)

Tabla 39 (continuación)

Estadísticos descriptivos de las variables cuantitativas continuas

Variable	Estrategia	Estadístico						
		Media	Error estándar	Mediana	Min.	Máx.	Varianza	SD
PRE-Faceta Optimismo	0	5,47222	0,202519	5,62500	3,250	7,000	1,107	1,052317
	1	5,44000	0,207931	5,75000	3,250	6,625	1,081	1,039656
	2	5,21000	0,165944	5,25000	3,500	7,000	0,688	0,829721
	3	4,33125	0,291078	4,31250	2,125	6,500	1,695	1,301742
POS-Faceta Optimismo	0	5,40741	0,181326	5,25000	3,750	7,000	0,888	0,942195
	1	5,61500	0,217097	6,00000	3,250	7,000	1,178	1,085487
	2	5,51000	0,194979	5,50000	3,375	7,000	0,950	0,974893
	3	4,61250	0,349094	4,62500	2,500	7,000	2,437	1,561197
PRE-Faceta Percepción Emocional	0	4,63333	0,162600	4,50000	2,800	6,300	0,714	0,844894
	1	4,65600	0,170790	4,70000	2,800	6,400	0,729	0,853952
	2	4,41200	0,170129	4,30000	3,000	6,100	0,724	0,850647
	3	4,27500	0,219554	4,20000	2,900	6,300	0,964	0,981875
POS-Faceta Percepción Emocional	0	4,66296	0,182117	4,80000	2,800	6,300	0,895	0,946308
	1	4,70800	0,169894	4,60000	3,100	6,300	0,722	0,849470
	2	4,57200	0,152809	4,50000	3,300	6,500	0,584	0,764046
	3	4,45000	0,227168	4,30000	2,800	6,400	1,032	1,015926
PRE-Faceta Regulación Emocional	0	4,62963	0,186091	4,41667	3,250	7,000	0,935	0,966958
	1	4,61667	0,155307	4,66667	3,083	6,167	0,603	0,776537
	2	4,44000	0,176549	4,41667	3,333	6,250	0,779	0,882744
	3	4,27500	0,171647	4,33333	2,333	5,583	0,589	0,767629
POS-Faceta Regulación Emocional	0	4,49691	0,167051	4,50000	3,000	6,750	0,753	0,868022
	1	4,59000	0,173467	4,58333	3,000	7,000	0,752	0,867334
	2	4,60667	0,154299	4,66667	3,000	6,417	0,595	0,771497
	3	4,06250	0,170876	4,12500	2,417	5,167	0,584	0,764181
PRE-Faceta Relaciones	0	5,22634	0,176380	5,33333	3,667	7,000	0,840	0,916495
	1	5,43111	0,126576	5,55556	4,111	6,444	0,401	0,632878
	2	5,32000	0,141613	5,44444	3,778	6,444	0,501	0,708066
	3	4,98889	0,160389	5,05556	3,333	6,000	0,514	0,717280
POS-Faceta Relaciones	0	5,16049	0,125715	5,11111	3,778	6,444	0,427	0,653236
	1	5,37778	0,129259	5,44444	4,111	6,889	0,418	0,646294
	2	5,29778	0,129691	5,22222	4,000	6,667	0,420	0,648455
	3	5,05000	0,174471	5,00000	3,111	6,222	0,609	0,780259
PRE-Factor Autocontrol	0	4,61171	0,157632	4,41935	3,129	6,581	0,671	0,819081
	1	4,54710	0,146960	4,48387	2,839	5,806	0,540	0,734800
	2	4,50452	0,161095	4,54839	3,323	6,548	0,649	0,805477
	3	4,29677	0,141572	4,38710	2,710	5,323	0,401	0,633131
POS-Factor Autocontrol	0	4,53763	0,140367	4,51613	3,387	6,258	0,532	0,729368
	1	4,58839	0,175545	4,58065	3,000	6,968	0,770	0,877726
	2	4,50065	0,135268	4,45161	3,161	6,355	0,457	0,676338
	3	4,41129	0,153458	4,48387	3,000	5,548	0,471	0,686285
PRE-Factor Bienestar	0	5,22908	0,156150	5,18519	3,259	6,815	0,658	0,811378
	1	5,40889	0,149186	5,51852	3,963	6,519	0,556	0,745929
	2	5,17185	0,115349	5,29630	4,074	6,407	0,333	0,576744
	3	4,50741	0,220577	4,55556	2,630	6,074	0,973	0,986451
POS-Factor Bienestar	0	5,19890	0,148913	5,22222	3,519	6,778	0,599	0,773775
	1	5,49630	0,158756	5,55556	4,148	6,778	0,630	0,793780
	2	5,34963	0,140807	5,48148	3,333	6,333	0,496	0,704035
	3	4,74630	0,256969	4,74074	2,481	6,407	1,321	1,149202

Nota: Elaboración propia. (Obs.: SD = desviación estándar)

Tabla 39 (continuación)

*Estadísticos descriptivos de las variables cuantitativas continuas*

Variable	Estrategia	Estadístico						
		Media	Error estándar	Mediana	Min.	Máx.	Varianza	SD
PRE-Factor Emocionalidad	0	4,75146	0,152539	4,78947	2,868	6,289	0,628	0,792616
	1	4,85053	0,123896	4,97368	3,421	5,789	0,384	0,619480
	2	4,77474	0,121880	4,65789	3,789	5,895	0,371	0,609400
	3	4,37632	0,133655	4,42105	3,237	5,395	0,357	0,597725
POS-Factor Emocionalidad	0	4,73977	0,138136	4,73684	3,263	5,895	0,515	0,717774
	1	4,87053	0,149984	4,71053	3,553	6,789	0,562	0,749920
	2	4,86000	0,123616	4,78947	3,789	6,368	0,382	0,618081
	3	4,59342	0,151732	4,72368	3,211	5,974	0,460	0,678566
PRE-Factor Sociabilidad	0	4,73308	0,168162	4,82759	2,862	6,414	0,764	0,873798
	1	4,60690	0,121328	4,58621	3,586	6,069	0,368	0,606642
	2	4,71586	0,128796	4,82759	3,138	5,931	0,415	0,643982
	3	4,25517	0,179951	4,27586	2,862	6,069	0,648	0,804767
POS-Factor Sociabilidad	0	4,65900	0,143862	4,68966	2,862	6,276	0,559	0,747528
	1	4,75448	0,154254	4,65517	3,379	6,552	0,595	0,771271
	2	4,73103	0,132478	4,68966	3,207	5,966	0,439	0,662392
	3	4,45345	0,185936	4,34483	3,103	6,448	0,691	0,831530
PRE-Factor Global de IE	0	4,74486	0,126718	4,65278	2,938	6,243	0,434	0,658446
	1	4,83472	0,101187	4,90278	3,882	5,771	0,256	0,505933
	2	4,75444	0,097082	4,66667	3,896	5,674	0,236	0,485412
	3	4,34167	0,100347	4,36806	3,451	5,285	0,201	0,448767
POS-Factor Global IE	0	4,71939	0,117816	4,80556	3,146	6,132	0,375	0,612188
	1	4,90306	0,137073	4,76389	3,847	6,757	0,470	0,685364
	2	4,80694	0,102825	4,86111	3,868	5,729	0,264	0,514123
	3	4,52882	0,128108	4,45139	3,313	5,896	0,328	0,572916

*Nota:* Elaboración propia. (Obs.: SD = desviación estándar)

## 9.2. Análisis cuantitativo: Contraste de hipótesis

Hay un gran número de pruebas estadísticas que pueden realizarse para el contraste de hipótesis. La elección de una u otra depende de varios factores, entre los cuales se destacan principalmente el tipo y la cantidad de variables que conforman la hipótesis. Existen *pruebas paramétricas* específicas para analizar datos que siguen una distribución de probabilidad basada en un conjunto fijo de parámetros (por ejemplo, la distribución normal) y *pruebas no paramétricas* que no tienen restricciones sobre la distribución de las variables y que pueden funcionar incluso cuando las variables son nominales u ordinales. Sin embargo, si se cumplen los supuestos para el análisis paramétrico, es recomendable usar las pruebas paramétricas en lugar de las no paramétricas, pues —en este caso— las últimas tienen menos potencia, es decir, son menos capaces de detectar diferencias o efectos allí donde los haya (Field, 2009, p. 551).

### 9.2.1. Primera hipótesis (H1)

La primera hipótesis postula que la *estrategia lúdica* y las variables referidas a la *adquisición de los conocimientos específicos de la programación orientada a objetos* están relacionadas, es decir, presentan una asociación significativa.

Cada par formado por la variable independiente —que es categórica politómica— y una de las variables dependientes —que son categóricas dicotómicas— corresponde a una posible hipótesis que se puede contrastar con la prueba de independencia  $\chi^2$ .

Esta prueba se realiza partiendo de una tabla cruzada entre las variables de las hipótesis. Después se analizan el cumplimiento de los supuestos —ninguna celda puede contener un valor esperado menor que 1 y no puede haber más de 20% de casillas con valores esperados menores que 5 (Muijs, 2004, p. 125)— y, si estos no se violan, los estadísticos pueden ser calculados y analizados para conservar o rechazar la hipótesis nula que postula que ambas variables no están relacionadas.

A continuación se presentan los resultados de las pruebas realizadas para analizar si la estrategia lúdica se relaciona con la adquisición de los ocho conocimientos específicos de la programación orientada a objetos estudiados en esta investigación.

#### Capacidad de identificar las relaciones entre clases en UML (POO1)

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO1* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO1* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 40). Ninguna casilla (0,0%) esperó un recuento menor que 5. Por lo tanto, se cumplieron los supuestos para aplicar esta prueba. No se consideró a los estudiantes que en el pretest ya habían demostrado poseer este conocimiento —no lo adquirirían, independientemente de la estrategia lúdica— y, en consecuencia, aparecen en la tabla 41 como casos perdidos.

Tabla 40

Tabla cruzada *Estrategia Lúdica* \* *POO1*

Estrategia Lúdica	0		POO1		Total
			No	Sí	
		Recuento	17	8	25
		Recuento esperado	14,8	10,2	25,0
		% dentro de Estrategia Lúdica	68,0%	32,0%	100,0%
		Residuo estandarizado	0,6	-0,7	
	1	Recuento	14	9	23
		Recuento esperado	13,6	9,4	23,0
		% dentro de Estrategia Lúdica	60,9%	39,1%	100,0%
		Residuo estandarizado	0,1	-0,1	
	2	Recuento	8	16	24
		Recuento esperado	14,2	9,8	24,0
		% dentro de Estrategia Lúdica	33,3%	66,7%	100,0%
		Residuo estandarizado	-1,7	2,0	
	3	Recuento	15	4	19
		Recuento esperado	11,3	7,7	19,0
		% dentro de Estrategia Lúdica	78,9%	21,1%	100,0%
		Residuo estandarizado	1,1	-1,3	
Total		Recuento	54	37	91
		Recuento esperado	54,0	37,0	91,0
		% dentro de Estrategia Lúdica	59,3%	40,7%	100,0%

Nota: Elaboración propia

Tabla 41

Resumen de procesamiento de casos *Estrategia Lúdica* \* *POO1*

	Casos					
	Válidos		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
Estrategia Lúdica * POO1	91	93,8%	6	6,2%	97	100,0%

Nota: Elaboración propia

El resultado de la prueba [ $\chi^2$  (3; N=91) = 10,555; p=0,014; V de Cramér=0,341] permitió rechazar la hipótesis nula  $H_0$  y aceptar la hipótesis alternativa  $H_1$ . Las variables *Estrategia lúdica* y *POO1* están estadísticamente relacionadas. Para interpretar la fuerza de la asociación encontrada se consultó la tabla 42, determinándose que la relación entre ambas variables es una asociación moderada.



Tabla 42

Descriptores del tamaño del efecto según la *V* de Cramér

V de Cramér	Interpretación
$0,00 \leq V < 0,10$	Asociación insignificante
$0,10 \leq V < 0,20$	Asociación débil
$0,20 \leq V < 0,40$	Asociación moderada
$0,40 \leq V < 0,60$	Asociación relativamente fuerte
$0,60 \leq V < 0,80$	Asociación fuerte
$0,80 \leq V \leq 1,00$	Asociación muy fuerte

Nota: Basado en “The Incorporation of Effect Size in Information Technology, Learning, and Performance Research” por Joe W. Kotrlík y Heather A. Williams, 2003, *Information Technology, Learning, and Performance Journal*, 21(1), p. 5. Copyright 2003 por Association of Information Systems

Para continuar el análisis, se observaron los residuos estandarizados como método post-hoc. Según Agresti y Finlay (2009), cuando el valor absoluto del residuo estandarizado en una celda es mayor que 2, esto es una evidencia de que esta celda contribuye a que el resultado de la prueba  $\chi^2$  sea significativo al nivel de  $\alpha=0,05$  (p. 230). En este caso, el único residuo con valor absoluto igual que 2 corresponde a la estrategia lúdica 2 (*Aprendizaje mediante la utilización de videojuegos*). En la figura 27 se puede ver que esta es la única estrategia lúdica mediante la cual la mayoría de los estudiantes adquirieron la capacidad de identificar las relaciones entre clases en UML (POO1).

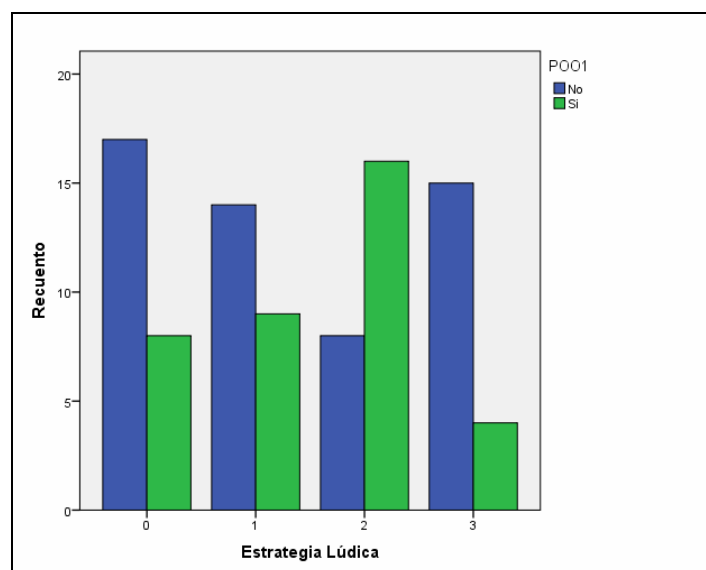


Figura 27. Adquisición de la capacidad de identificar las relaciones entre clases en UML (POO1).

Elaboración propia

**Capacidad de implementar las relaciones entre clases en Java (POO2)**

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO2* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO2* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 43). Cuatro casillas (50,0%) esperaron un recuento menor que 5. Por lo tanto, no se cumplieron los supuestos para aplicar esta prueba.

**Tabla 43**

*Tabla cruzada Estrategia Lúdica \* POO2*

		POO2		Total	
		No	Sí		
Estrategia Lúdica	0	Recuento	25	1	26
		Recuento esperado	24,3	1,7	26,0
		% dentro de Estrategia Lúdica	96,2%	3,8%	100,0%
		Residuo estandarizado	0,1	-0,5	
	1	Recuento	22	2	24
		Recuento esperado	22,4	1,6	24,0
		% dentro de Estrategia Lúdica	91,7%	8,3%	100,0%
		Residuo estandarizado	-0,1	0,3	
	2	Recuento	20	3	23
		Recuento esperado	21,5	1,5	23,0
		% dentro de Estrategia Lúdica	87,0%	13,0%	100,0%
		Residuo estandarizado	-0,3	1,2	
	3	Recuento	19	0	19
		Recuento esperado	17,8	1,2	19,0
		% dentro de Estrategia Lúdica	100,0%	0,0%	100,0%
		Residuo estandarizado	0,3	-1,1	
Total		Recuento	86	6	92
		Recuento esperado	86,0	6,0	92,0
		% dentro de Estrategia Lúdica	93,5%	6,5%	100,0%

*Nota:* Elaboración propia

En la figura 28 se puede ver que, independientemente de la estrategia lúdica empleada, la mayoría de los estudiantes no adquirieron la capacidad de implementar las relaciones entre clases en Java (POO2).

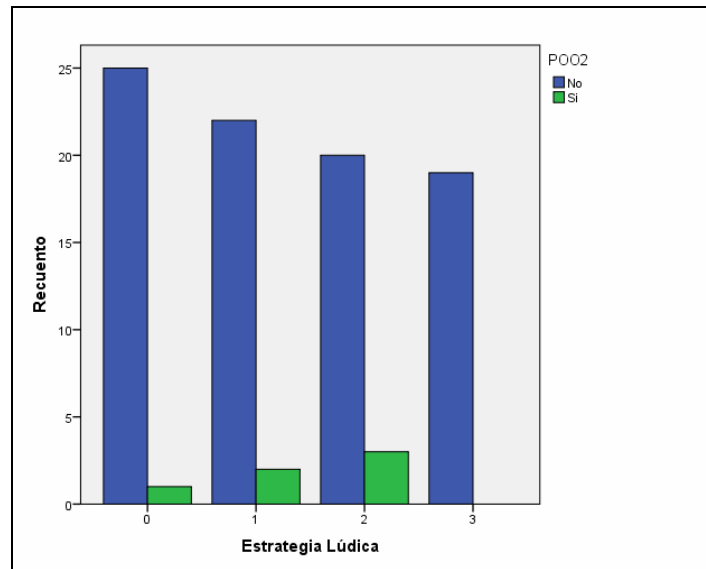


Figura 28. Adquisición de la capacidad de implementar las relaciones entre clases en Java (POO2).

Elaboración propia

### Capacidad de ordenar por rango los tipos primitivos en Java (POO3)

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO3* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO3* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 44). Solo una casilla (12,5%) esperó un recuento menor que 5. Por lo tanto, se cumplieron los supuestos para aplicar esta prueba. No se consideró a los estudiantes que en el pretest ya habían demostrado poseer este conocimiento —no lo adquirirían, independientemente de la estrategia lúdica— y, en consecuencia, aparecen en la tabla 45 como casos perdidos.

Tabla 44

Tabla cruzada Estrategia Lúdica \* POO3

Estrategia Lúdica		POO3		Total
		No	Sí	
0	Recuento	19	7	26
	Recuento esperado	18,4	7,6	26,0
	% dentro de Estrategia Lúdica	73,1%	26,9%	100,0%
	Residuo estandarizado	0,1	-0,2	
1	Recuento	16	6	22
	Recuento esperado	15,6	6,4	22,0
	% dentro de Estrategia Lúdica	72,7%	27,3%	100,0%
	Residuo estandarizado	0,1	-0,2	
2	Recuento	10	8	18
	Recuento esperado	12,7	5,3	18,0
	% dentro de Estrategia Lúdica	55,6%	44,4%	100,0%
	Residuo estandarizado	-0,8	1,2	
3	Recuento	13	3	16
	Recuento esperado	11,3	4,7	16,0
	% dentro de Estrategia Lúdica	81,3%	18,8%	100,0%
	Residuo estandarizado	0,5	-0,8	
Total	Recuento	58	24	82
	Recuento esperado	58,0	24,0	82,0
	% dentro de Estrategia Lúdica	70,7%	29,3%	100,0%

Nota: Elaboración propia

Tabla 45

Resumen de procesamiento de casos Estrategia Lúdica \* POO3

	Casos					
	Válidos		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
Estrategia Lúdica * POO3	82	84,5%	15	15,5%	97	100,0%

Nota: Elaboración propia

El resultado de la prueba [ $\chi^2$  (3; N=82) = 2,969; p=0,408; V de Cramér=0,19] no permitió rechazar la hipótesis nula  $H_0$  y aceptar la hipótesis alternativa  $H_1$ . Las variables *Estrategia lúdica* y *POO3* no están estadísticamente relacionadas.

En la figura 29 se puede ver que, independientemente de la estrategia lúdica empleada, la mayoría de los estudiantes no adquirieron la capacidad de ordenar por rango los tipos primitivos en Java (POO3).

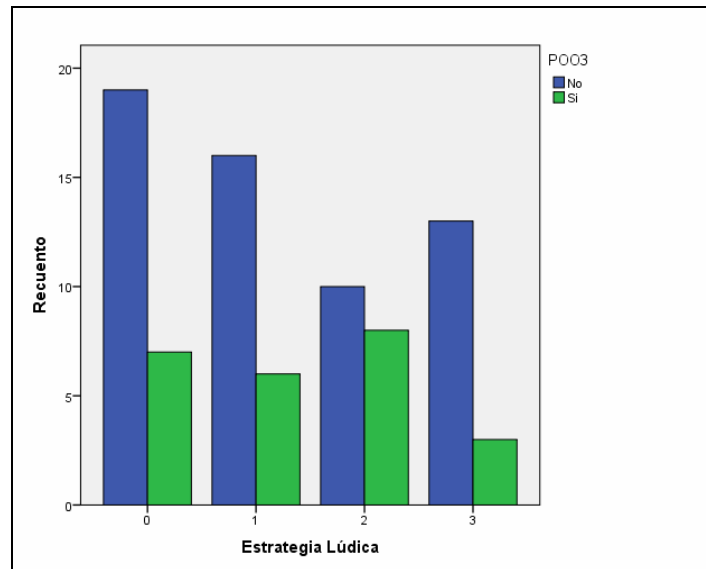


Figura 29. Adquisición de la capacidad de ordenar por rango los tipos primitivos en Java (POO3).

Elaboración propia

### Capacidad de nombrar las clases de envoltorio en Java (POO4)

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO4* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO4* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 46). Ninguna casilla (0,0%) esperó un recuento menor que 5. Por lo tanto, se cumplieron los supuestos para aplicar esta prueba. No se consideró a los estudiantes que en el pretest ya habían demostrado poseer este conocimiento —no lo adquirirían, independientemente de la estrategia lúdica— y, en consecuencia, aparecen en la tabla 47 como casos perdidos.

Tabla 46

Tabla cruzada Estrategia Lúdica \* POO4

Estrategia Lúdica		POO4		Total
		No	Sí	
0	Recuento	11	13	24
	Recuento esperado	11,1	12,9	24,0
	% dentro de Estrategia Lúdica	45,8%	54,2%	100,0%
	Residuo estandarizado	0,0	0,0	
1	Recuento	9	14	23
	Recuento esperado	10,7	12,3	23,0
	% dentro de Estrategia Lúdica	39,1%	60,9%	100,0%
	Residuo estandarizado	-0,5	0,5	
2	Recuento	9	14	23
	Recuento esperado	10,7	12,3	23,0
	% dentro de Estrategia Lúdica	39,1%	60,9%	100,0%
	Residuo estandarizado	-0,5	0,5	
3	Recuento	10	4	14
	Recuento esperado	6,5	7,5	14,0
	% dentro de Estrategia Lúdica	71,4%	28,6%	100,0%
	Residuo estandarizado	1,4	-1,3	
Total	Recuento	39	45	84
	Recuento esperado	39,0	45,0	84,0
	% dentro de Estrategia Lúdica	46,4%	53,6%	100,0%

Nota: Elaboración propia

Tabla 47

Resumen de procesamiento de casos Estrategia Lúdica \* POO4

	Casos					
	Válidos		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
Estrategia Lúdica * POO4	84	86,6%	13	13,4%	97	100,0%

Nota: Elaboración propia

Si bien en la figura 30 se puede ver que —salvo en el caso de la estrategia lúdica 3 (*Ludificación del aprendizaje/Gamification*)—, la mayoría de los estudiantes adquirieron la capacidad de nombrar las clases de envoltorio en Java (POO4), el resultado de la prueba [ $\chi^2$  (3; N=84) = 4,506; p=0,218; V de Cramér=0,232] no permitió rechazar la hipótesis nula  $H_0$  y aceptar la hipótesis alternativa  $H_1$ . Las variables *Estrategia lúdica* y *POO4* no están estadísticamente relacionadas.

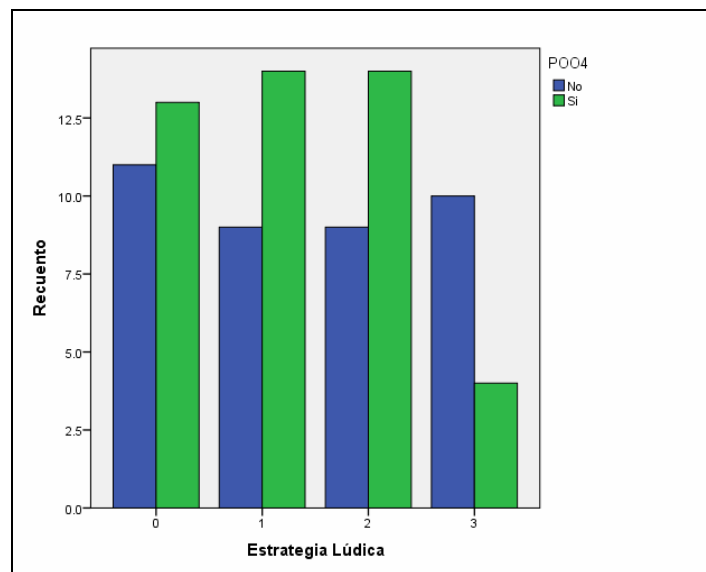


Figura 30. Adquisición de la capacidad de nombrar las clases de envoltorio (POO4).

Elaboración propia

### Capacidad de ordenar los calificadores de acceso que regulan el encapsamiento en Java (POO5)

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO5* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO5* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 48). Cuatro casillas (50,0%) esperaron un recuento menor que 5. Por lo tanto, no se cumplieron los supuestos para aplicar esta prueba.

Tabla 48

Tabla cruzada Estrategia Lúdica \* POO5

Estrategia Lúdica		POO5		Total
		No	Si	
0	Recuento	21	2	23
	Recuento esperado	19,8	3,2	23,0
	% dentro de Estrategia Lúdica	91,3%	8,7%	100,0%
	Residuo estandarizado	0,3	-0,7	
1	Recuento	16	5	21
	Recuento esperado	18,1	2,9	21,0
	% dentro de Estrategia Lúdica	76,2%	23,8%	100,0%
	Residuo estandarizado	-0,5	1,2	
2	Recuento	16	2	18
	Recuento esperado	15,5	2,5	18,0
	% dentro de Estrategia Lúdica	88,9%	11,1%	100,0%
	Residuo estandarizado	0,1	-0,3	
3	Recuento	15	2	17
	Recuento esperado	14,6	2,4	17,0
	% dentro de Estrategia Lúdica	88,2%	11,8%	100,0%
	Residuo estandarizado	0,1	-0,2	
Total	Recuento	68	11	79
	Recuento esperado	68,0	11,0	79,0
	% dentro de Estrategia Lúdica	86,1%	13,9%	100,0%

Nota: Elaboración propia

En la figura 31 se puede ver que, independientemente de la estrategia lúdica empleada, la mayoría de los estudiantes no adquirieron la capacidad de ordenar los calificadores de acceso que regulan el encapsulamiento en Java (POO5).

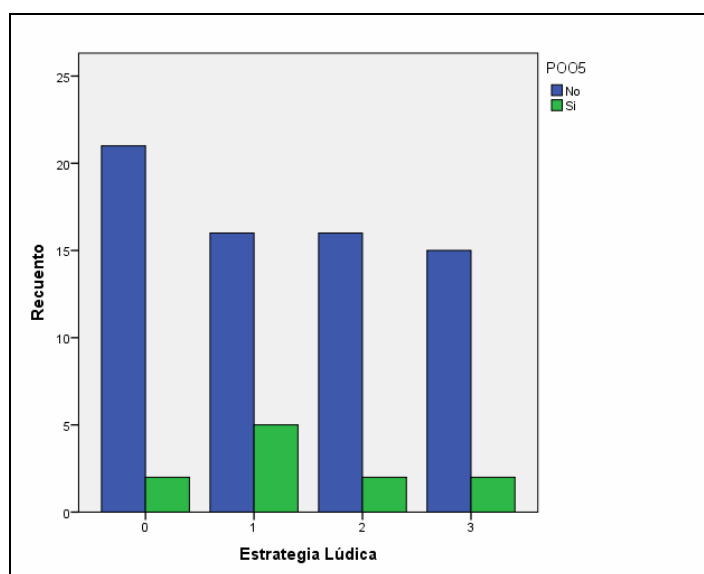


Figura 31. Adquisición de la capacidad de ordenar los calificadores de acceso que regulan el encapsulamiento en Java (POO5). Elaboración propia



**Capacidad de usar la abstracción para diseñar en UML un diagrama de clases que aplique la herencia (POO6)**

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO6* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO6* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 49). Solo una casilla (12,5%) esperó un recuento menor que 5. Por lo tanto, se cumplieron los supuestos para aplicar esta prueba. No se consideró a los estudiantes que en el pretest ya habían demostrado poseer este conocimiento —no lo adquirirían, independientemente de la estrategia lúdica— y, en consecuencia, aparecen en la tabla 50 como casos perdidos.

**Tabla 49**

*Tabla cruzada Estrategia Lúdica \* POO6*

		POO6		Total	
		No	Sí		
Estrategia Lúdica	0	Recuento	20	6	26
		Recuento esperado	16,5	9,5	26,0
		% dentro de Estrategia Lúdica	76,9%	23,1%	100,0%
		Residuo estandarizado	0,8	-1,1	
	1	Recuento	14	7	21
		Recuento esperado	13,4	7,6	21,0
		% dentro de Estrategia Lúdica	66,7%	33,3%	100,0%
		Residuo estandarizado	0,2	-0,2	
	2	Recuento	8	10	18
		Recuento esperado	11,5	6,5	18,0
		% dentro de Estrategia Lúdica	44,4%	55,6%	100,0%
		Residuo estandarizado	-1,0	1,4	
	3	Recuento	7	5	12
		Recuento esperado	7,6	4,4	12,0
		% dentro de Estrategia Lúdica	58,3%	41,7%	100,0%
		Residuo estandarizado	-0,2	0,3	
Total		Recuento	49	28	77
		Recuento esperado	49,0	28,0	77,0
		% dentro de Estrategia Lúdica	63,6%	36,4%	100,0%

*Nota:* Elaboración propia

Tabla 50

Resumen de procesamiento de casos Estrategia Lúdica \* POO6

	Casos					
	Válidos		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
Estrategia Lúdica * POO6	77	79,4%	20	20,6%	97	100,0%

Nota: Elaboración propia

Aunque en la figura 32 se puede ver que —salvo en el caso de la estrategia lúdica 2 (*Aprendizaje mediante la utilización de videojuegos*)—, la mayoría de los estudiantes no adquirieron la capacidad de usar la abstracción para diseñar en UML un diagrama de clases que aplique la herencia (POO6), el resultado de la prueba [ $\chi^2$  (3; N=77) = 5,078; p=0,17; V de Cramér=0,257] no permitió rechazar la hipótesis nula  $H_0$  y aceptar la hipótesis alternativa  $H_1$ . Las variables *Estrategia lúdica* y *POO6* no están estadísticamente relacionadas.

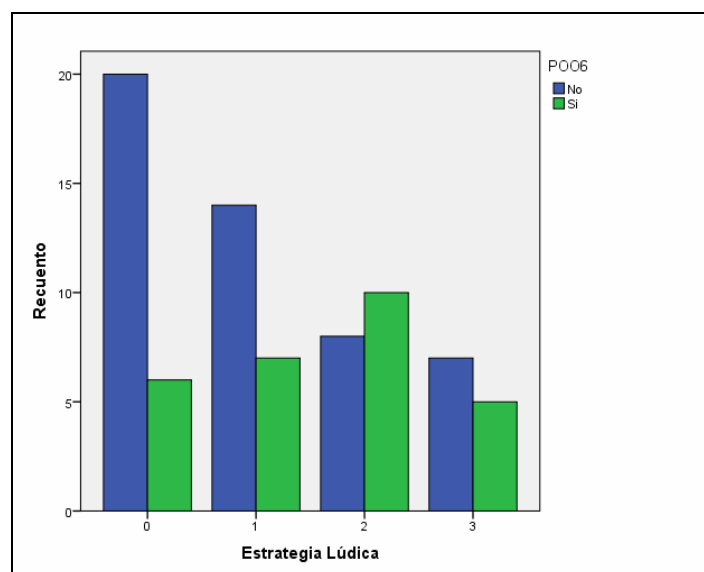


Figura 32. Adquisición de la capacidad de usar la abstracción para diseñar en UML un diagrama de clases que aplique la herencia (POO6). Elaboración propia

### Capacidad de describir la utilidad de la sobrecarga de constructores y métodos en Java (POO7)

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO7* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO7* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 51). Cuatro casillas (50,0%) esperaron un recuento menor que 5. Por lo tanto, no se cumplieron los supuestos para aplicar esta prueba.

**Tabla 51**

*Tabla cruzada Estrategia Lúdica \* POO7*

		POO7		Total	
		No	Sí		
Estrategia Lúdica	0	Recuento	17	0	17
		Recuento esperado	15,1	1,9	17,0
		% dentro de Estrategia Lúdica	100,0%	0,0%	100,0%
		Residuo estandarizado	0,5	-1,4	
	1	Recuento	16	3	19
		Recuento esperado	16,9	2,1	19,0
		% dentro de Estrategia Lúdica	84,2%	15,8%	100,0%
		Residuo estandarizado	-0,2	0,6	
	2	Recuento	11	3	14
		Recuento esperado	12,4	1,6	14,0
		% dentro de Estrategia Lúdica	78,6%	21,4%	100,0%
		Residuo estandarizado	-0,4	1,2	
	3	Recuento	12	1	13
		Recuento esperado	11,6	1,4	13,0
		% dentro de Estrategia Lúdica	92,3%	7,7%	100,0%
		Residuo estandarizado	0,1	-0,4	
Total		Recuento	56	7	63
		Recuento esperado	56,0	7,0	63,0
		% dentro de Estrategia Lúdica	88,9%	11,1%	100,0%

*Nota:* Elaboración propia

En la figura 33 se puede ver que, independientemente de la estrategia lúdica empleada, la mayoría de los estudiantes no adquirieron la capacidad de describir la utilidad de la sobrecarga de constructores y métodos en Java (POO7).

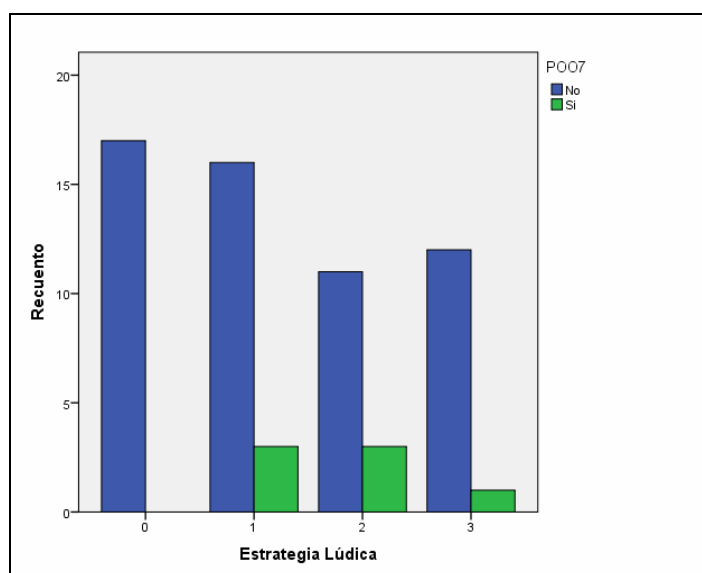


Figura 33. Adquisición de la capacidad de describir la utilidad de la sobrecarga de constructores y métodos en Java (POO7). Elaboración propia

### Capacidad de reconocer e interpretar el polimorfismo en Java (POO8)

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba  $\chi^2$ :

- $H_0$ : Las variables *Estrategia lúdica* y *POO8* no están relacionadas.
- $H_1$ : Las variables *Estrategia lúdica* y *POO8* están relacionadas.

Después se generó la tabla cruzada entre las variables de la hipótesis (tabla 52). Solo una casilla (12,5%) esperó un recuento menor que 5. Por lo tanto, se cumplieron los supuestos para aplicar esta prueba. No se consideró a los estudiantes que en el pretest ya habían demostrado poseer este conocimiento —no lo adquirirían, independientemente de la estrategia lúdica— y, en consecuencia, aparecen en la tabla 53 como casos perdidos.

**Tabla 52**

*Tabla cruzada Estrategia Lúdica \* POO8*

		POO8		Total	
		No	Sí		
Estrategia Lúdica	0	Recuento	5	13	18
		Recuento esperado	7,7	10,3	18,0
		% dentro de Estrategia Lúdica	27,8%	72,2%	100,0%
		Residuo estandarizado	-1,0	0,8	
	1	Recuento	7	13	20
		Recuento esperado	8,6	11,4	20,0
		% dentro de Estrategia Lúdica	35,0%	65,0%	100,0%
		Residuo estandarizado	-0,5	0,5	
	2	Recuento	6	5	11
		Recuento esperado	4,7	6,3	11,0
		% dentro de Estrategia Lúdica	54,5%	45,5%	100,0%
		Residuo estandarizado	0,6	-0,5	
	3	Recuento	9	5	14
		Recuento esperado	6,0	8,0	14,0
		% dentro de Estrategia Lúdica	64,3%	35,7%	100,0%
		Residuo estandarizado	1,2	-1,1	
Total		Recuento	27	36	63
		Recuento esperado	27,0	36,0	63,0
		% dentro de Estrategia Lúdica	42,9%	57,1%	100,0%

*Nota:* Elaboración propia

**Tabla 53**

*Resumen de procesamiento de casos Estrategia Lúdica \* POO8*

	Casos					
	Válidos		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
Estrategia Lúdica * POO8	63	64,9%	34	35,1%	97	100,0%

*Nota:* Elaboración propia

Si bien en la figura 34 puede verse que, en la mitad de las estrategias lúdicas, la mayoría de los estudiantes adquirieron la capacidad de reconocer e interpretar el polimorfismo en Java (POO8), el resultado de la prueba [ $\chi^2$  (3; N=63) = 5,414; p=0,153; V de Cramér=0,293] no permitió rechazar la hipótesis nula  $H_0$  y aceptar la hipótesis alternativa  $H_1$ . Las variables *Estrategia lúdica* y *POO8* no están estadísticamente relacionadas.

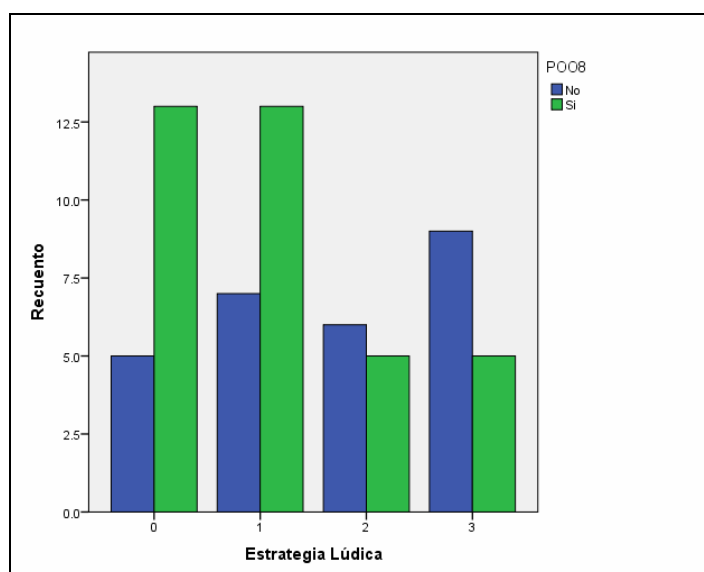


Figura 34. Adquisición de la capacidad de reconocer e interpretar el polimorfismo en Java (POO8).

Elaboración propia

### 9.2.2. Segunda hipótesis (H2)

La segunda hipótesis postula que, entre los grupos formados en función de la *estrategia lúdica* empleada (variable independiente), las variables referidas a las *relaciones interpersonales en el aula* medidas en el postest (variables dependientes) difieren significativamente, una vez eliminada la influencia del pretest sobre los resultados del postest.

Cada par formado por la variable independiente —que es categórica politómica— y una de las variables dependientes —que son cuantitativas continuas— corresponde a una posible hipótesis que puede contrastarse con la prueba ANCOVA (análisis de covarianza) de un factor (llamada así porque hay una única variable independiente).

Es bastante usual utilizar ANCOVA para comparar los postests de los distintos grupos (de control y experimentales), tomando los pretests como covariable. Según Dugard y Todman (1995), al utilizar ANCOVA se le da al pretest su status correcto de auténtica covariable, ya que después de haber hecho el esfuerzo de recolectar los datos del pretest, no tendría sentido hacer un uso inapropiado de ellos (p. 183). En relación con las diferencias existentes en los pretests, Balluerka Lasa y Vergara Iraeta (2002) señalan lo siguiente:

ANCOVA posibilita eliminar el sesgo sistemático derivado de las diferencias existentes *a priori* entre los grupos de tratamiento, las cuales son consecuencia de la ausencia de aleatorización en la asignación de los sujetos a los tratamientos, propia de los diseños cuasiexperimentales (p. 209).

De acuerdo con Maxwell y Delaney (2004), comparado con el ANOVA (análisis de varianza) de las ganancias (valor de postest menos valor de pretest), este método es casi siempre preferible porque proporciona más potencia y precisión (p. 446).

Para poder llevar a cabo la prueba ANCOVA, deben cumplirse los supuestos que se muestran en la tabla 54.

**Tabla 54**

*Supuestos para la prueba ANCOVA*

Supuesto	Explicación
Medida de la VD	La variable dependiente requiere un nivel de medición al menos por intervalos (debe ser continua)
Independencia	Cada observación debe ser independiente de las demás (cada sujeto debe pertenecer a un único grupo)
Homocedasticidad	Los grupos deben tener una varianza homogénea (o sea, una dispersión similar en sus distribuciones)
Normalidad de los errores	Los errores deben seguir una distribución normal en cada nivel de la variable independiente
Niveles de tratamiento fijos	Los niveles de la variable independiente deben ser fijados por el investigador (modelo de efectos fijos)
Linealidad entre la VD y la CV	La regresión de Y (la variable dependiente) sobre X (la covariable) debe ser lineal en cada nivel de la VI
Independencia entre la VI y la CV	La covariable debe ser independiente de la variable independiente (el tratamiento no debe ejercer ningún efecto sobre la covariable)
Homogeneidad de las pendientes de regresión	La pendiente de la línea de regresión entre la variable dependiente y la covariable debe ser la misma para cada nivel de la variable independiente
Fiabilidad en la medida de la CV	Los valores de la covariable deben ser obtenidos sin errores de medición

*Nota:* Basado en *Statistical Concepts: A Second Course (4ª ed.)* por R. G. Lomax y D. L. Hahs-Vaughn, 2013, p. 151, New York: Routledge. Copyright 2012 by Taylor & Francis Group, LLC. Obs.: VD: Variable dependiente; VI: Variable independiente; CV: Covariable

Si estos supuestos no se violan, los estadísticos pueden ser calculados y analizados para conservar o rechazar la hipótesis nula que postula que no existen diferencias estadísticamente significativas entre los grupos.

En aquellos casos donde los supuestos no se cumplen, una alternativa es utilizar alguna de las versiones no paramétricas de ANCOVA disponibles, como las propuestas por Quade (1967), Puri y Sen (1969), Conover e Iman (1982) y Bathke y Brunner (2003). Según Olejnik y Algina (1985), las versiones no paramétricas de ANCOVA son menos restrictivas que la prueba ANCOVA paramétrica, debido a que no asumen los supuestos de normalidad, linealidad y homocedasticidad dentro de los grupos o entre ellos (p. 52).

La versión no paramétrica de ANCOVA utilizada en esta investigación es la de Conover e Iman (1982) —aunque estos autores reconocen que la idea ya había sido publicada originalmente en 1971 por McSweeney y Porter—. Básicamente, esta técnica consiste en llevar a cabo la prueba ANCOVA paramétrica con los valores de la variable dependiente y la covariable convertidos en rangos.

De los nueve supuestos requeridos para utilizar la versión paramétrica de ANCOVA, en este trabajo hay cinco que se cumplieron a través del diseño metodológico: la variable dependiente de esta hipótesis es cuantitativa continua, cada participante no podía formar parte de más de un grupo, los niveles de la variable independiente fueron fijados por el autor, los pretests se tomaron mediante instrumentos estándar (de validez y confiabilidad reconocidas, con lo cual se aseguró la fiabilidad en la medición de la covariable) y ocurrieron antes de cada uno de los tratamientos (con lo cual los tratamientos no pudieron tener ningún efecto sobre las covariables).

El cumplimiento de cada uno de los otros cuatro supuestos debió evaluarse mediante diversas técnicas estadísticas que se mencionarán oportunamente.

A continuación se exponen los resultados de las pruebas realizadas para analizar si entre los grupos formados en función de la *estrategia lúdica* empleada existen diferencias significativas en las dos variables referidas a las *relaciones interpersonales en el aula*.



### Distancia social media propia

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-DSP*, una vez eliminada la influencia que la variable *PRE-DSP* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-DSP*, una vez eliminada la influencia que la variable *PRE-DSP* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-DSP* en cada grupo seguían o no una distribución normal. Según Delgado de la Torre (2004, p. 140), la prueba de Shapiro-Wilk es para muestras pequeñas (de tamaño  $n \leq 30$ ), como es el caso de los grupos experimentales y de control de esta investigación. Como ocurre en la mayoría de las pruebas de normalidad, en la prueba de Shapiro-Wilk, la hipótesis nula es que la muestra proviene de una variable normal. Si  $p < 0,05$ , esta hipótesis de normalidad se puede rechazar con un nivel de confianza del 95% (tabla 55).

**Tabla 55**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-DSP)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-DSP	0	0,923	27	0,045	No
	1	0,949	25	0,239	Sí
	2	0,945	25	0,189	Sí
	3	0,849	20	0,005	No

*Nota:* Elaboración propia

La violación del supuesto de normalidad obligó a llevar a cabo la versión no paramétrica de la prueba ANCOVA de un factor. Para convertir las variables *PRE-DSP* y *POS-DSP* en rangos se crearon dos nuevas variables (*Rango de PRE-DSP* y *Rango de POS-DSP*), cuyos estadísticos descriptivos se muestran en la tabla 56.

Tabla 56

Estadísticos descriptivos de las variables Rango de PRE-DSP y Rango de POS-DSP

Variable	Estrategia Lúdica	Estadístico		
		Media	Desviación estándar	N
Rango de PRE-DSP	0	50,94444	18,987513	27
	1	54,70000	23,260750	25
	2	46,08000	33,273075	25
	3	42,90000	36,376540	20
Rango de POS-DSP	0	53,55556	16,964518	27
	1	49,20000	26,979931	25
	2	46,74000	32,798031	25
	3	45,42500	35,753276	20

Nota: Elaboración propia

Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 57.

Tabla 57

Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-DSP; Covariable: Rango de PRE-DSP)

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	58275,018 <sup>a</sup>	4	14568,754	75,609	0,000	0,767	302,437	1,000
Intersección	232897,000	1	232897,000	1208,696	0,041	0,929	1208,696	1,000
Rango de PRE-DSP	57418,356	1	57418,356	297,991	0,000	0,764	297,991	1,000
Estrategia Lúdica	856,662	3	285,554	1,482	0,225	0,046	4,446	0,380
Error	17726,982	92	192,685					
Total	308899,000	97						
Total corregido	76002,000	96						

a. R al cuadrado = 0,767 (R al cuadrado ajustada = 0,757)

b. Se ha calculado utilizando alpha = 0,05

Nota: Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *Rango de PRE-DSP* sobre la variable dependiente *Rango de POS-DSP* [ $F_{\text{Rango de PRE-DSP}}(1; 92) = 297,991$ ;  $p < 0,001$ ;  $\eta^2 = 0,764$ ; potencia observada = 1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *Rango de POS-DSP* [ $F_{\text{Estrategia Lúdica}}(3; 92) = 1,482$ ;  $p = 0,225$ ;  $\eta^2 = 0,046$ ; potencia observada = 0,380].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la *distancia social media propia*, una vez eliminada la influencia del pretest.

### Distancia social media del grupo

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- H<sub>0</sub>: Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-DSG*, una vez eliminada la influencia que la variable *PRE-DSG* pudiera ejercer sobre esta.
- H<sub>1</sub>: Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-DSG*, una vez eliminada la influencia que la variable *PRE-DSG* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-DSG* en cada grupo seguían o no una distribución normal (tabla 58).

**Tabla 58**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-DSG)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-DSG	0	0,933	27	0,083	Sí
	1	0,939	25	0,137	Sí
	2	0,931	25	0,093	Sí
	3	0,815	20	0,001	No

*Nota:* Elaboración propia

La violación del supuesto de normalidad obligó a llevar a cabo la versión no paramétrica de la prueba ANCOVA de un factor. Para convertir las variables *PRE-DSG* y *POS-DSG* en rangos se crearon dos nuevas variables (*Rango de PRE-DSG* y *Rango de POS-DSG*), cuyos estadísticos descriptivos se muestran en la tabla 59.

Tabla 59

Estadísticos descriptivos de las variables Rango de PRE-DSG y Rango de POS-DSG

Variable	Estrategia Lúdica	Estadístico		
		Media	Desviación estándar	N
Rango de PRE-DSG	0	52,74074	19,914901	27
	1	55,26000	22,732191	25
	2	47,46000	34,951133	25
	3	38,05000	32,543453	20
Rango de POS-DSG	0	57,53704	20,997219	27
	1	47,64000	21,089847	25
	2	46,66000	32,718407	25
	3	42,10000	36,311373	20

Nota: Elaboración propia

Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 60.

Tabla 60

Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-DSG; Covariable: Rango de PRE-DSG)

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	64537,110 <sup>a</sup>	4	16134,278	129,667	0,000	0,849	518,670	1,000
Intersección	232897,000	1	232897,000	1871,739	0,000	0,953	1871,739	1,000
Rango de PRE-DSG	62360,511	1	62360,511	501,177	0,000	0,845	501,177	1,000
Estrategia Lúdica	2176,599	3	725,533	5,831	0,001	0,160	17,493	0,945
Error	11447,390	92	124,428					
Total	308881,500	97						
Total corregido	75984,500	96						

a. R al cuadrado = 0,849 (R al cuadrado ajustada = 0,843)

b. Se ha calculado utilizando alpha = 0,05

Nota: Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *Rango de PRE-DSG* sobre la variable dependiente *Rango de POS-DSG* [ $F_{\text{Rango de PRE-DSG}}(1; 92) = 501,177; p < 0,001; \eta^2 = 0,845; \text{potencia observada} = 1,000$ ]. Además, hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente

Rango de POS-DSG [ $F_{\text{Estrategia Lúdica}} (3; 92) = 5,831; p=0,001; \eta^2=0,160; \text{potencia observada}=0,945$ ].

El tamaño del efecto sugirió que el 16% de la varianza en la *distancia social media del grupo* se puede atribuir a la *estrategia lúdica* empleada, una vez eliminadas las diferencias que pudieran existir *a priori* entre los grupos. Con G\*Power 3.1.9.2 se convirtió  $\eta^2$  en un índice *f* de Cohen=0,436, determinándose mediante la tabla 61 que se trató de un efecto de tamaño grande.

**Tabla 61**

*Descriptores del tamaño del efecto según el índice f de Cohen*

f de Cohen	Interpretación
0,10	Efecto de tamaño pequeño
0,25	Efecto de tamaño medio
0,40	Efecto de tamaño grande

*Nota:* Basado en “The Incorporation of Effect Size in Information Technology, Learning, and Performance Research” por Joe W. Kotrlik y Heather A. Williams, 2003, *Information Technology, Learning, and Performance Journal*, 21(1), p. 5. Copyright 2003 por Association of Information Systems

Por lo tanto, se rechazó la hipótesis nula  $H_0$  y se aceptó la hipótesis alternativa  $H_1$  que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, existen diferencias significativas en el postest de la *distancia social media del grupo*, una vez eliminada la influencia del pretest.

A continuación se calcularon las medias marginales de la variable dependiente *Rango de POS-DSG* estimadas con base en la covariable *Rango de PRE-DSG* (tabla 62).

**Tabla 62**

*Medias marginales de Rango de POS-DSG estimadas con base en la covariable Rango de PRE-DSG*

Estrategia Lúdica	Media	Error estándar	Intervalo de confianza al 95%	
			Límite inferior	Límite superior
0	54,085 <sup>a</sup>	2,152	49,811	58,360
1	41,864 <sup>a</sup>	2,246	37,403	46,325
2	48,081 <sup>a</sup>	2,232	43,648	52,514
3	52,204 <sup>a</sup>	2,535	47,168	57,239

a. Las covariables que aparecen en el modelo se evalúan en los valores siguientes: Rango de PRE-DSG = 49,00000.

*Nota:* Elaboración propia

Por último, se llevó a cabo un análisis post-hoc para evaluar las diferencias de las medias anteriores, entre las parejas formadas en función de la variable independiente *Estrategia Lúdica*. Para controlar el riesgo de que aumente el error de tipo I, se utilizó la corrección de Bonferroni. La tabla 63 muestra los resultados de este análisis.

**Tabla 63**

*Comparación de las medias marginales de Rango de POS-DSG entre las parejas formadas según Estrategia Lúdica*

(I) Estrategia Lúdica	(J) Estrategia Lúdica	Diferencia de medias (I-J)	Error estándar	Sig.	95% de intervalo de confianza para diferencia	
					Límite inferior	Límite superior
0	1	12,222*	3,098	0,001	3,868	20,575
	2	6,004	3,104	0,337	-2,365	14,374
	3	1,881	3,347	1,000	-7,144	10,907
1	0	-12,222*	3,098	0,001	-20,575	-3,868
	2	-6,217	3,172	0,318	-14,770	2,335
	3	-10,340*	3,422	0,020	-19,567	-1,113
2	0	-6,004	3,104	0,337	-14,374	2,365
	1	6,217	3,172	0,318	-2,335	14,770
	3	-4,123	3,369	1,000	-13,208	4,962
3	0	-1,881	3,347	1,000	-10,907	7,144
	1	10,340*	3,422	0,020	1,113	19,567
	2	4,123	3,369	1,000	-4,962	13,208

\*. La diferencia de medias es significativa en el nivel 0,05.

*Nota:* Elaboración propia

Se observaron diferencias estadísticamente significativas entre las medias marginales estimadas del grupo de control ( $p=0,001$ ) y las del grupo en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*), así como también entre las de este último y las del grupo en el cual se empleó la estrategia lúdica 3 (*Ludificación del aprendizaje/Gamification*) ( $p=0,020$ ). En la figura 35 puede observarse que, tal como se había mostrado en la tabla 61, el grupo en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*) es el que tiene el menor valor en la *distancia social media del grupo* (lo cual es positivo para este grupo, desde un punto de vista psicopedagógico).

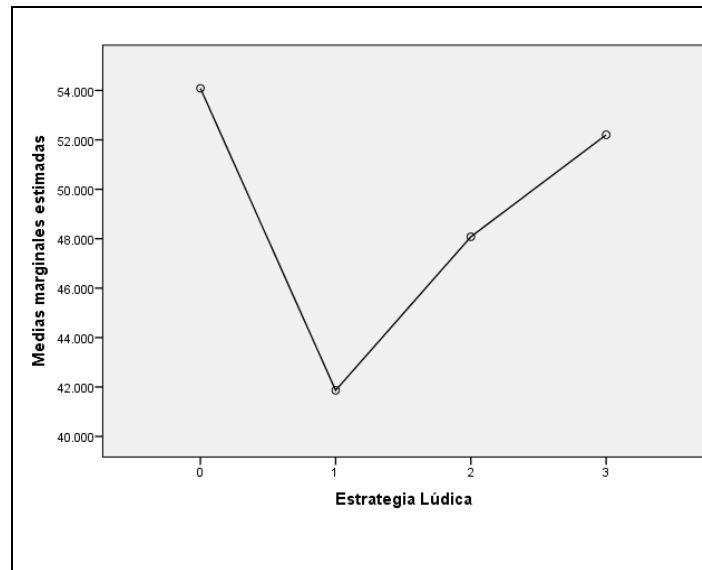


Figura 35. Medias marginales estimadas de *Rango de POS-DSG*. Elaboración propia

### 9.2.3. Tercera hipótesis (H3)

La tercera hipótesis postula que, dentro de cada grupo formado en función de la *estrategia lúdica* empleada, las variables referidas a las *relaciones interpersonales en el aula* presentan diferencias significativas entre los valores medidos en el pretest y en el postest.

Cada par formado por una de estas variables medida en el pretest y la misma variable medida en el postest —en ambos casos se trata de valores cuantitativos continuos— corresponde a una posible hipótesis que puede contrastarse con la prueba *t* para muestras relacionadas (paramétrica) o la prueba de rangos con signo de Wilcoxon para muestras relacionadas (no paramétrica). Son pruebas *para muestras relacionadas* porque —dentro de cada grupo— los estudiantes que participaron en el pretest son los mismos que participaron en el postest.

Si las *diferencias* entre los valores medidos en el pretest y en el postest siguen una distribución normal —sin importar si los propios valores lo hacen—, puede utilizarse la prueba *t* para conservar o rechazar la hipótesis nula que postula que la *media* de las diferencias entre los valores del pretest y del postest es igual a 0. De lo contrario, la prueba de rangos con signo de Wilcoxon se puede usar para conservar o rechazar la hipótesis nula que postula que la *mediana* de las diferencias entre el pretest y el postest es igual a 0.

En consecuencia, para empezar se usó la prueba de Shapiro-Wilk con el objetivo de determinar si las *diferencias* entre los valores de las dos variables referidas a las *relaciones interpersonales en el aula* —DSP y DSG— medidos en el pretest y en el postest seguían o no una distribución normal (tabla 64).

**Tabla 64**

*Prueba de Shapiro-Wilk para las diferencias entre el pretest y el postest (relaciones interpersonales en el aula)*

Estrategia Lúdica	Variable	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
0	DIF-DSP	0,802	27	0,000	No
	DIF-DSG	0,915	27	0,030	No
1	DIF-DSP	0,950	25	0,253	Sí
	DIF-DSG	0,923	25	0,059	Sí
2	DIF-DSP	0,808	25	0,000	No
	DIF-DSG	0,930	25	0,085	Sí
3	DIF-DSP	0,918	20	0,091	Sí
	DIF-DSG	0,949	20	0,348	Sí

*Nota:* Elaboración propia

A continuación se llevaron a cabo las pruebas que se reseñan en la tabla 65, en función de que se haya cumplido o no, en el análisis anterior, el supuesto de normalidad.

**Tabla 65**

*Contraste de las hipótesis referidas a las diferencias entre el pretest y el postest (relaciones interpersonales en el aula)*

Estrategia Lúdica	H <sub>0</sub> se refiere a*	Prueba	Resultado	Efecto	Decisión
0	DSP	Wilcoxon	$z=-3,210$ ; $p=0,001$ ; $r=-0,437$	Medio a grande	Se rechazó H <sub>0</sub>
	DSG	Wilcoxon	$z=-3,864$ ; $p<0,001$ ; $r=-0,526$	Grande	Se rechazó H <sub>0</sub>
1	DSP	t	$t(24) = 4,599$ ; $p<0,001$ ; $d=-0,920$	Grande	Se rechazó H <sub>0</sub>
	DSG	t	$t(24) = 8,005$ ; $p<0,001$ ; $d=-1,601$	Grande	Se rechazó H <sub>0</sub>
2	DSP	Wilcoxon	$z=-3,448$ ; $p=0,001$ ; $r=-0,488$	Medio a grande	Se rechazó H <sub>0</sub>
	DSG	t	$t(24) = 5,050$ ; $p<0,001$ ; $d=-1,010$	Grande	Se rechazó H <sub>0</sub>
3	DSP	t	$t(19) = 1,413$ ; $p=0,174$ ; $d=-0,316$	Pequeño a medio	Se conservó H <sub>0</sub>
	DSG	t	$t(19) = 3,708$ ; $p=0,001$ ; $d=-0,829$	Grande	Se rechazó H <sub>0</sub>

\*. H<sub>0</sub> postula que la media (prueba t) o la mediana (Wilcoxon) de las diferencias entre los valores del pretest y del postest es igual a 0.

*Nota:* Elaboración propia



El tamaño del efecto se mostró, en función de la prueba usada, a través de los índices  $r$  (en las pruebas de rangos con signo de Wilcoxon) y  $d$  de Cohen (en las pruebas  $t$  para muestras relacionadas). El índice  $r$  se obtuvo como el cociente entre  $z$  y la raíz cuadrada del número de observaciones (el cual, en este caso, era el doble de la cantidad de participantes). El índice  $d$  de Cohen se obtuvo como el cociente entre la media de las diferencias y la desviación estándar de las diferencias. Para interpretar el significado de los tamaños del efecto, se consultó en cada caso la tabla 66 que se muestra a continuación.

**Tabla 66**

*Descriptores del tamaño del efecto según los índices  $r$  y  $d$  de Cohen*

$r$	$d$ de Cohen	Interpretación
0,10	0,20	Efecto de tamaño pequeño
0,30	0,50	Efecto de tamaño medio
0,50	0,80	Efecto de tamaño grande

*Nota:* Basado en *SPSS survival manual: a step by step guide to data analysis using SPSS (4ª ed.)* por J. Pallant, 2011, p. 210 y p. 232, Crows Nest: Allen & Unwin. Copyright 2011 por Julie Pallant

Por último se comentan las diferencias significativas encontradas entre los valores medidos en el pretest y en el postest. Las medias (M) y medianas (Me) que se mencionan en los comentarios pueden consultarse en la tabla 39.

En el grupo de control, la prueba de rangos con signo de Wilcoxon reveló una reducción estadísticamente significativa [ $z=-3,210$ ;  $p=0,001$ ] en la *distancia social media propia*, desde el pretest (Me=4,61538) hasta el postest (Me=4,50000), con un efecto de tamaño medio a grande ( $r=-0,437$ ), así como también una reducción estadísticamente significativa [ $z=-3,864$ ;  $p<0,001$ ] en la *distancia social media del grupo*, desde el pretest (Me=4,61538) hasta el postest (Me=4,53846), con un efecto de tamaño grande ( $r=-0,526$ ).

En el grupo en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*), la prueba  $t$  para muestras relacionadas reveló una reducción estadísticamente significativa [ $t(24) = 4,599$ ;  $p<0,001$ ] en la *distancia social media propia*, desde el pretest (M=4,60500) hasta el postest (M=4,36167), con un efecto de tamaño grande ( $d=-0,920$ ), y además una reducción estadísticamente significativa [ $t(24) = 8,005$ ;  $p<0,001$ ] en la *distancia social media del grupo*, desde el pretest (M= M=4,60500) hasta el postest (M=4,36167), también con un efecto de tamaño grande ( $d=-1,601$ ).

En el grupo en el cual se empleó la estrategia lúdica 2 (*Aprendizaje mediante la utilización de videojuegos*), la prueba de rangos con signo de Wilcoxon reveló una reducción estadísticamente significativa [ $z=-3,448$ ;  $p=0,001$ ] en la *distancia social media propia*, desde el pretest (Me=4,33333) hasta el postest (Me=4,25000), con un efecto de tamaño medio a grande ( $r=-0,488$ ). En el mismo grupo, la prueba  $t$  para muestras relacionadas también reveló una reducción estadísticamente significativa [ $t(24) = 5,050$ ;  $p<0,001$ ] en la *distancia social media del grupo*, desde el pretest (M=4,43000) hasta el postest (M=4,28167), con un efecto de tamaño grande ( $d=-1,010$ ).

En el grupo en el cual se empleó la estrategia lúdica 3 (*Ludificación del aprendizaje/ Gamification*), solo la prueba  $t$  para muestras relacionadas reveló una reducción estadísticamente significativa [ $t(19) = 3,708$ ;  $p=0,001$ ] en la *distancia social media del grupo*, desde el pretest (M=4,28158) hasta el postest (M=4,16053), con un efecto de tamaño grande ( $d=-0,829$ ).

#### 9.2.4. Cuarta hipótesis (H4)

La cuarta hipótesis postula que, entre los grupos formados según la *estrategia lúdica* empleada (variable independiente), las variables referidas a la *inteligencia emocional* de los estudiantes medidas en el postest (variables dependientes) difieren significativamente, una vez eliminada la influencia del pretest sobre los resultados del postest.

Al igual que en la segunda hipótesis, aquí también cada par formado por la variable independiente —que es categórica politómica— y una de las variables dependientes —que son cuantitativas continuas— corresponde a una posible hipótesis que puede contrastarse con la prueba ANCOVA de un factor.

A continuación se exponen los resultados de las pruebas realizadas para analizar si entre los grupos formados en función de la *estrategia lúdica* empleada existen diferencias significativas en las 20 variables referidas a la *inteligencia emocional* de los estudiantes.

#### Faceta Adaptabilidad

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Adaptabilidad*, una vez eliminada la influencia que la variable *PRE-Faceta Adaptabilidad* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Adaptabilidad*, una vez eliminada la influencia que la variable *PRE-Faceta Adaptabilidad* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Adaptabilidad* en cada grupo seguían o no una distribución normal (tabla 67), constatándose el cumplimiento de este supuesto.

**Tabla 67**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Adaptabilidad)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Adaptabilidad	0	0,976	27	0,757	Sí
	1	0,965	25	0,521	Sí
	2	0,979	25	0,857	Sí
	3	0,949	20	0,358	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene, cuya hipótesis nula es que la varianza de error de la variable dependiente es igual entre grupos, para determinar si se cumplía el supuesto de homocedasticidad (homogeneidad de las varianzas). Si  $p < 0,05$ , la hipótesis de homocedasticidad se puede rechazar con un nivel de confianza del 95%. El resultado [ $F(3; 93)=0,746$ ;  $p=0,527$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Adaptabilidad* (variable dependiente) sobre *PRE-Faceta Adaptabilidad* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Adaptabilidad* contra *PRE-Faceta Adaptabilidad*: uno general (figura 36) y los otros cuatro separados según la variable independiente (figura 37). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

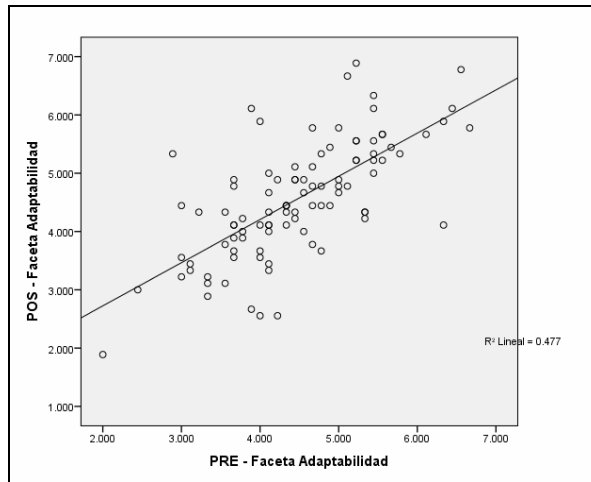


Figura 36. Diagrama de dispersión general de *POS-Faceta Adaptabilidad* contra *PRE-Faceta Adaptabilidad*. Elaboración propia

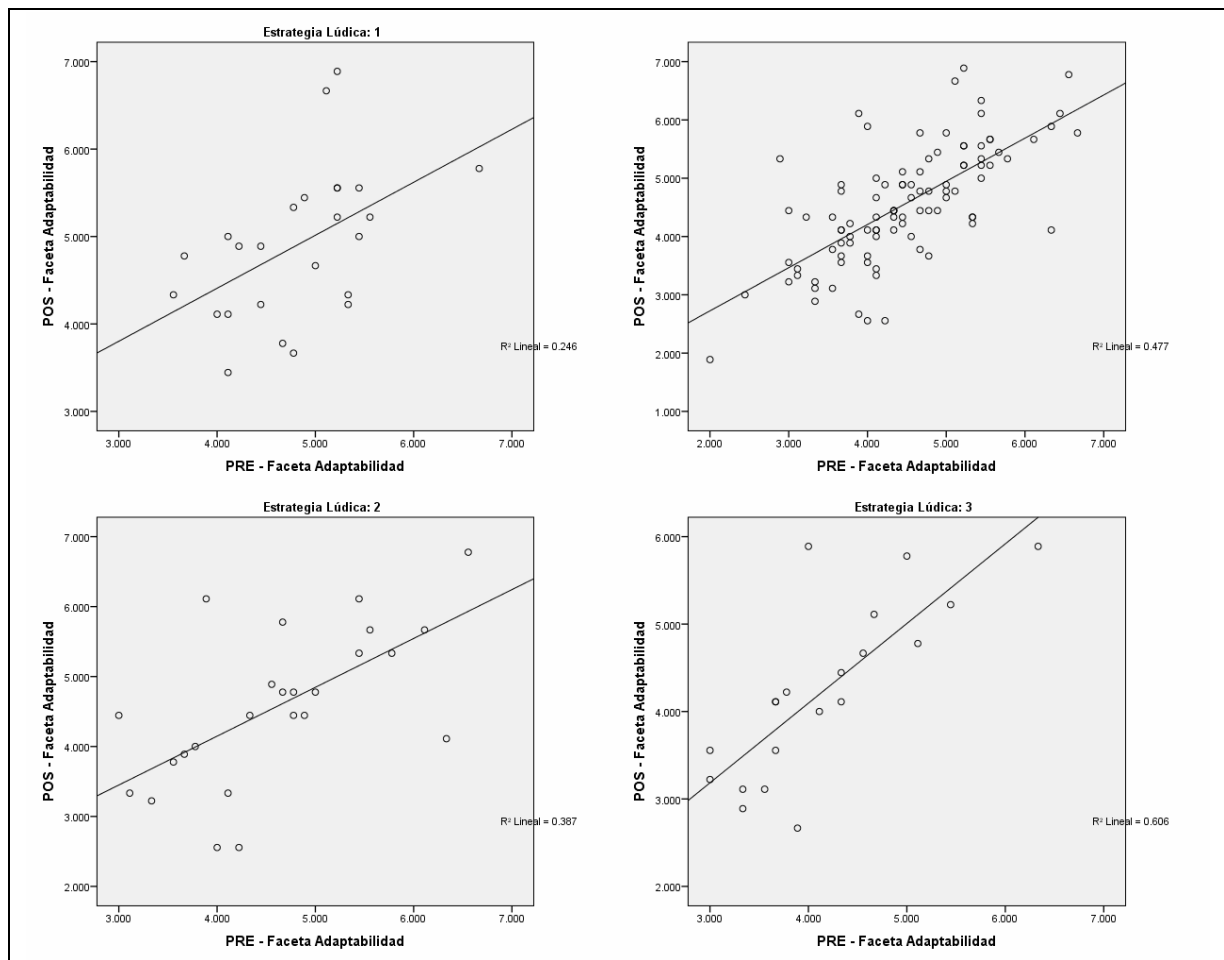


Figura 37. Diagramas de dispersión de *POS-Faceta Adaptabilidad* contra *PRE-Faceta Adaptabilidad* separados según *Estrategia Lúdica*. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Adaptabilidad* contra

*PRE-Faceta Adaptabilidad* separados según *Estrategia Lúdica* (figura 37) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Adaptabilidad* por *Estrategia Lúdica* [ $F(3; 89) = 0,403$ ;  $p=0,751$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 68.

**Tabla 68**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Adaptabilidad;*

*Covariable: PRE-Faceta Adaptabilidad)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	46,114 <sup>a</sup>	4	11,529	21,544	0,000	0,484	86,175	1,000
Intersección	2014,062	1	2014,062	3763,738	0,000	0,976	3763,738	1,000
PRE	45,475	1	45,475	84,981	0,000	0,480	84,981	1,000
Estrategia Lúdica	0,639	3	0,213	0,398	0,755	0,013	1,194	0,126
Error	49,231	92	0,535					
Total	2109,407	97						
Total corregido	95,346	96						

a. R al cuadrado = 0,484 (R al cuadrado ajustada = 0,461)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Adaptabilidad* sobre la variable dependiente *POS-Faceta Adaptabilidad* [ $F_{PRE-Faceta Adaptabilidad} (1; 92) = 84,981$ ;  $p<0,001$ ;  $\eta^2=0,480$ ; potencia observada=1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Adaptabilidad* [ $F_{Estrategia Lúdica} (3; 92) = 0,398$ ;  $p=0,755$ ;  $\eta^2=0,013$ ; potencia observada=0,126].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Adaptabilidad*, una vez eliminada la influencia del pretest.

## Faceta Asertividad

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Asertividad*, una vez eliminada la influencia que la variable *PRE-Faceta Asertividad* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Asertividad*, una vez eliminada la influencia que la variable *PRE-Faceta Asertividad* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Asertividad* en cada grupo seguían o no una distribución normal (tabla 69), constatándose el cumplimiento de este supuesto.

**Tabla 69**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Asertividad)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Asertividad	0	0,982	27	0,913	Sí
	1	0,932	25	0,097	Sí
	2	0,961	25	0,426	Sí
	3	0,957	20	0,495	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 0,824$ ;  $p=0,484$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Asertividad* (variable dependiente) sobre *PRE-Faceta Asertividad* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Asertividad* contra *PRE-Faceta Asertividad*: uno general (figura 38) y los otros cuatro separados según la variable independiente

(figura 39). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

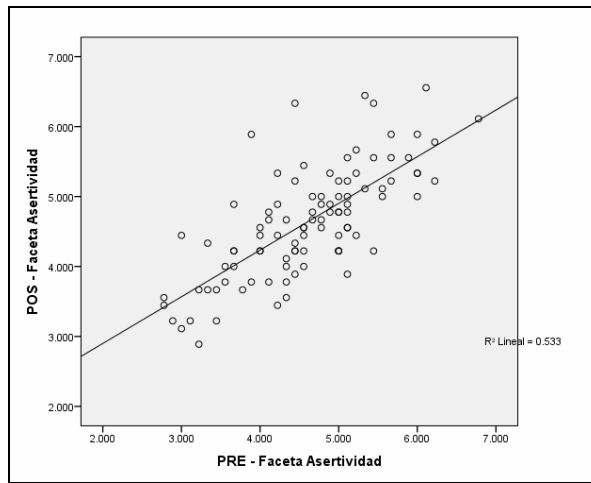


Figura 38. Diagrama de dispersión general de POS-Faceta Asertividad contra PRE-Faceta Asertividad. Elaboración propia

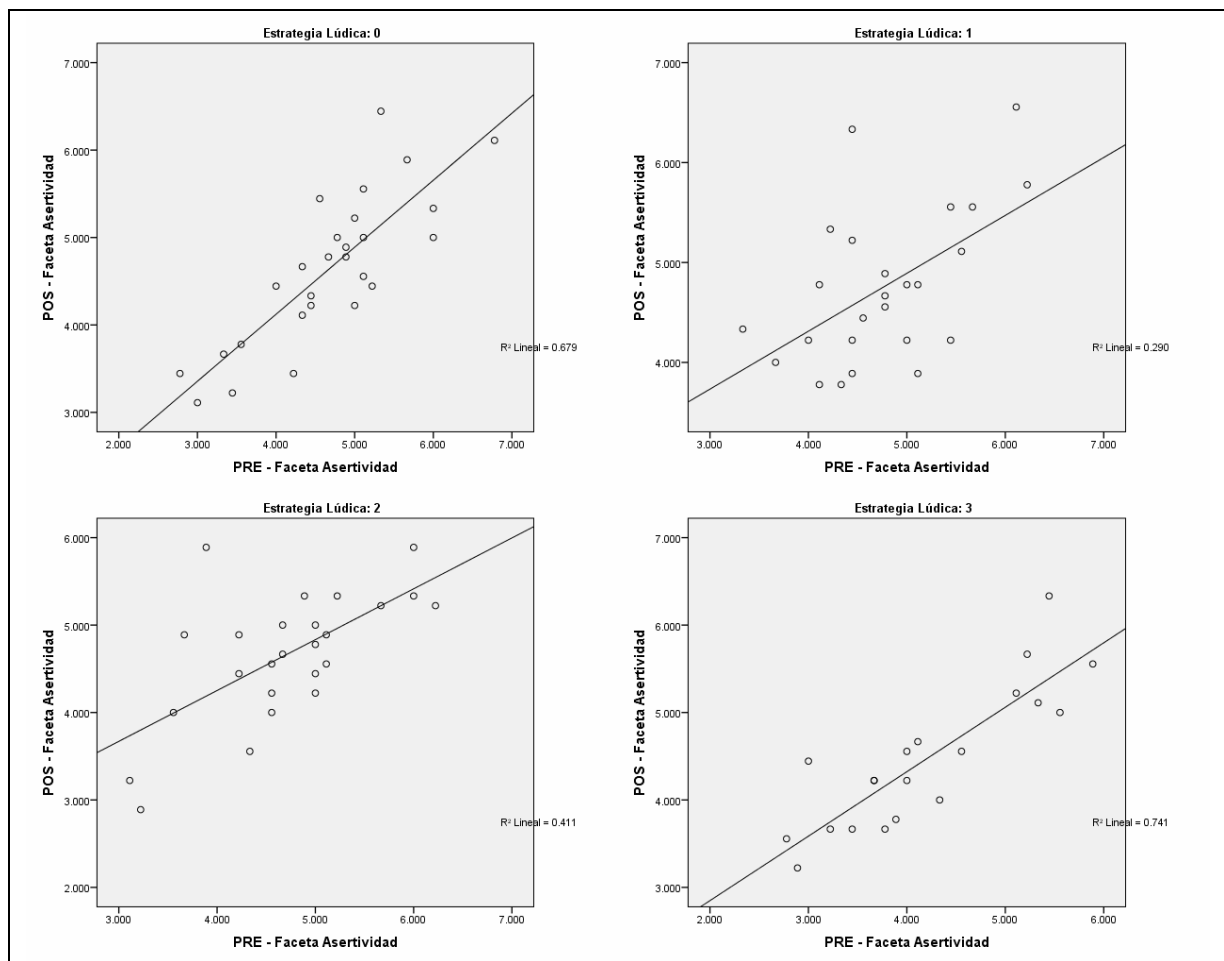


Figura 39. Diagramas de dispersión de POS-Faceta Asertividad contra PRE-Faceta Asertividad separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Asertividad* contra *PRE-Faceta Asertividad* separados según *Estrategia Lúdica* (figura 39) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Asertividad* por *Estrategia Lúdica* [ $F(3; 89) = 0,546$ ;  $p=0,652$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 70.

**Tabla 70**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Asertividad; Covariable: PRE-Faceta Asertividad)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	32,555 <sup>a</sup>	4	8,139	26,858	0,000	0,539	107,434	1,000
Intersección	2085,568	1	2085,568	6882,598	0,000	0,987	6882,598	1,000
PRE	32,227	1	32,227	106,351	0,000	0,536	106,351	1,000
Estrategia Lúdica	0,328	3	0,109	0,361	0,781	0,012	1,082	0,118
Error	27,878	92	0,303					
Total	2146,000	97						
Total corregido	60,432	96						

a. R al cuadrado = 0,539 (R al cuadrado ajustada = 0,519)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Asertividad* sobre la variable dependiente *POS-Faceta Asertividad* [ $F_{PRE-Faceta Asertividad} (1; 92) = 106,351$ ;  $p<0,001$ ;  $\eta^2=0,536$ ; potencia observada=1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Asertividad* [ $F_{Estrategia Lúdica} (3; 92) = 0,361$ ;  $p=0,781$ ;  $\eta^2=0,012$ ; potencia observada=0,118].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la *faceta Asertividad*, una vez eliminada la influencia del pretest.



### Faceta Autoestima

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Autoestima*, una vez eliminada la influencia que la variable *PRE-Faceta Autoestima* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Autoestima*, una vez eliminada la influencia que la variable *PRE-Faceta Autoestima* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Autoestima* en cada grupo seguían o no una distribución normal (tabla 71), constatándose el cumplimiento de este supuesto.

**Tabla 71**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Autoestima)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Autoestima	0	0,970	27	0,591	Sí
	1	0,958	25	0,380	Sí
	2	0,953	25	0,299	Sí
	3	0,963	20	0,610	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,474; p=0,227$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Autoestima* (variable dependiente) sobre *PRE-Faceta Autoestima* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Autoestima* contra *PRE-Faceta Autoestima*: uno general (figura 40) y los otros cuatro separados según la variable independiente

(figura 41). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

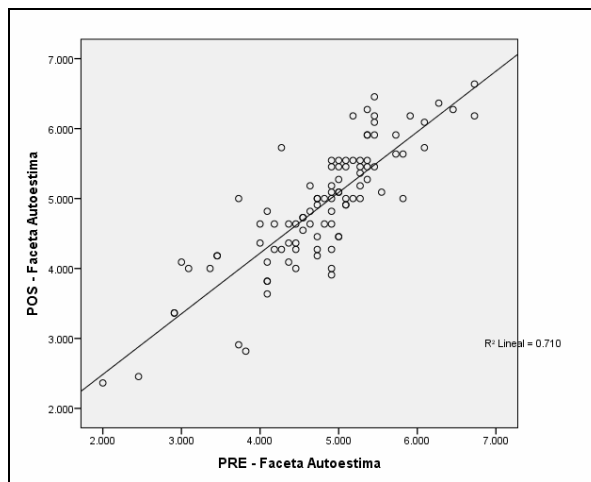


Figura 40. Diagrama de dispersión general de *POS-Faceta Autoestima* contra *PRE-Faceta Autoestima*. Elaboración propia

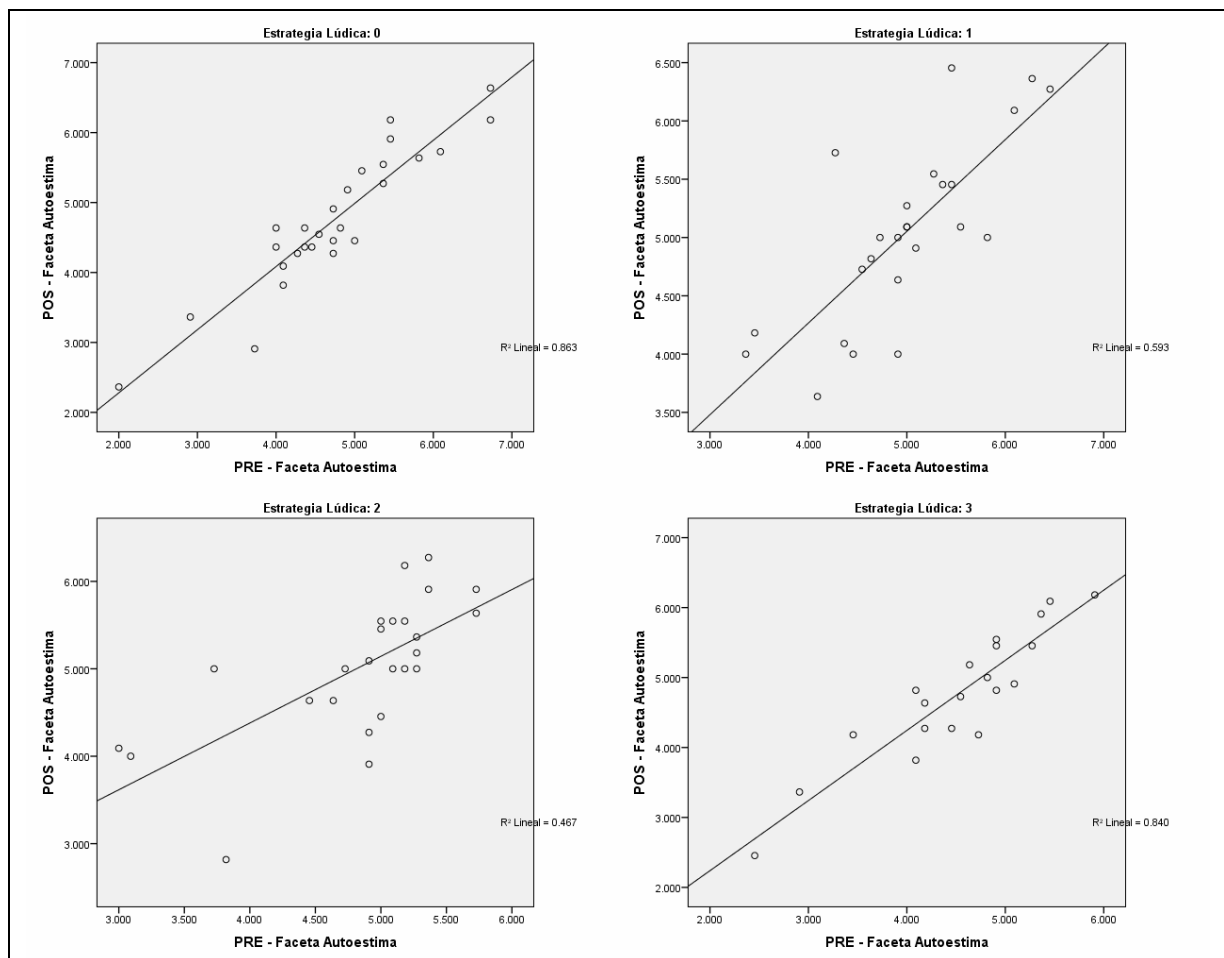


Figura 41. Diagramas de dispersión de *POS-Faceta Autoestima* contra *PRE-Faceta Autoestima* separados según *Estrategia Lúdica*. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Autoestima* contra *PRE-Faceta Autoestima* separados según *Estrategia Lúdica* (figura 41) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Autoestima* por *Estrategia Lúdica* [ $F(3; 89) = 0,739$ ;  $p=0,532$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 72.

**Tabla 72**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Autoestima;*

*Covariable: PRE-Faceta Autoestima)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	53,073 <sup>a</sup>	4	13,268	58,872	0,000	0,719	235,489	1,000
Intersección	2324,251	1	2324,251	10312,905	0,000	0,991	10312,905	1,000
PRE	52,413	1	52,413	232,563	0,000	0,717	232,563	1,000
Estrategia Lúdica	0,660	3	0,220	0,976	0,408	0,031	2,927	0,258
Error	20,734	92	0,225					
Total	2398,058	97						
Total corregido	73,807	96						

a. R al cuadrado = 0,719 (R al cuadrado ajustada = 0,707)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Autoestima* sobre la variable dependiente *POS-Faceta Autoestima* [ $F_{PRE-Faceta Autoestima} (1; 92) = 232,563$ ;  $p<0,001$ ;  $\eta^2=0,717$ ; potencia observada=1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Autoestima* [ $F_{Estrategia Lúdica} (3; 92) = 0,976$ ;  $p=0,408$ ;  $\eta^2=0,031$ ; potencia observada=0,258].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Autoestima*, una vez eliminada la influencia del pretest.

## Faceta Automotivación

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Automotivación*, una vez eliminada la influencia que la variable *PRE-Faceta Automotivación* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Automotivación*, una vez eliminada la influencia que la variable *PRE-Faceta Automotivación* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Automotivación* en cada grupo seguían o no una distribución normal (tabla 73), constatándose el cumplimiento de este supuesto.

**Tabla 73**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Automotivación)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Automotivación	0	0,970	27	0,613	Sí
	1	0,977	25	0,822	Sí
	2	0,963	25	0,488	Sí
	3	0,958	20	0,513	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,170$ ;  $p=0,325$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Automotivación* (variable dependiente) sobre *PRE-Faceta Automotivación* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Automotivación* contra *PRE-Faceta Automotivación*: uno general (figura 42) y los otros cuatro separados según la variable

independiente (figura 43). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

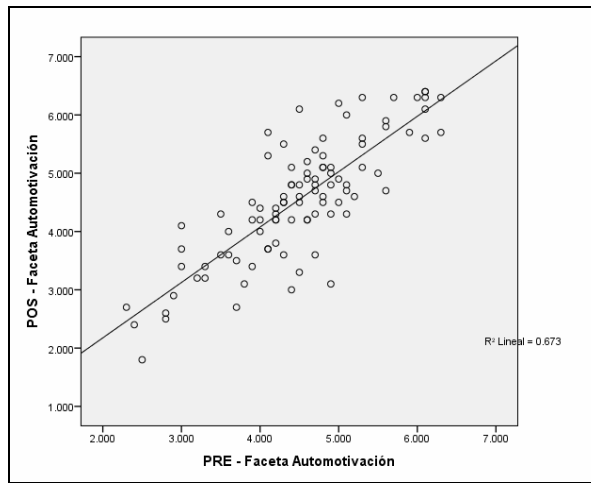


Figura 42. Diagrama de dispersión general de POS-Faceta Automotivación contra PRE-Faceta Automotivación. Elaboración propia

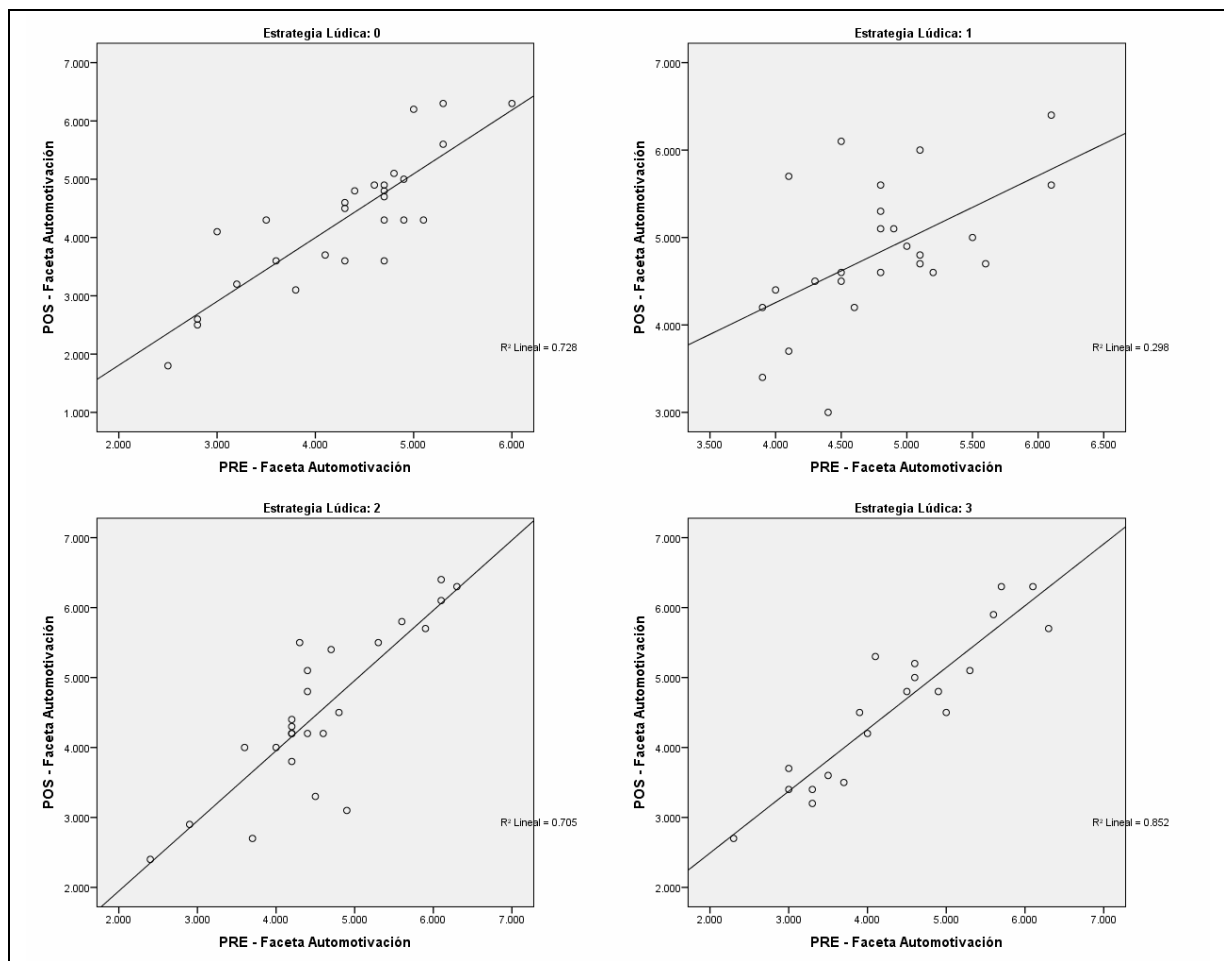


Figura 43. Diagramas de dispersión de POS-Faceta Automotivación contra PRE-Faceta Automotivación separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Automotivación* contra *PRE-Faceta Automotivación* separados según *Estrategia Lúdica* (figura 43) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Automotivación* por *Estrategia Lúdica* [ $F(3; 89) = 0,940$ ;  $p=0,425$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 74.

**Tabla 74**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Automotivación; Covariable: PRE-Faceta Automotivación)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	71,119 <sup>a</sup>	4	17,780	49,027	0,000	0,681	196,107	1,000
Intersección	2007,688	1	2007,688	5536,134	0,000	0,984	5536,134	1,000
PRE	70,352	1	70,352	193,995	0,000	0,678	193,995	1,000
Estrategia Lúdica	0,766	3	0,255	0,704	0,552	0,022	2,112	0,194
Error	33,364	92	0,363					
Total	2112,170	97						
Total corregido	104,482	96						

a. R al cuadrado = 0,681 (R al cuadrado ajustada = 0,667)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Automotivación* sobre la variable dependiente *POS-Faceta Automotivación* [ $F_{PRE-Faceta Automotivación} (1; 92) = 193,995$ ;  $p<0,001$ ;  $\eta^2=0,678$ ; potencia observada=1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Automotivación* [ $F_{Estrategia Lúdica} (3; 92) = 0,704$ ;  $p=0,552$ ;  $\eta^2=0,022$ ; potencia observada=0,194].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Automotivación*, una vez eliminada la influencia del pretest.

## Faceta Conciencia Social

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Conciencia Social*, una vez eliminada la influencia que la variable *PRE-Faceta Conciencia Social* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Conciencia Social*, una vez eliminada la influencia que la variable *PRE-Faceta Conciencia Social* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Conciencia Social* en cada grupo seguían o no una distribución normal (tabla 75), constatándose el cumplimiento de este supuesto.

**Tabla 75**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Conciencia Social)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Conciencia Social	0	0,973	27	0,690	Sí
	1	0,979	25	0,871	Sí
	2	0,967	25	0,571	Sí
	3	0,971	20	0,783	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,677; p=0,177$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Conciencia Social* (variable dependiente) sobre *PRE-Faceta Conciencia Social* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Conciencia Social* contra *PRE-Faceta Conciencia Social*: uno general (figura 44) y los otros cuatro separados según la variable

independiente (figura 45). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

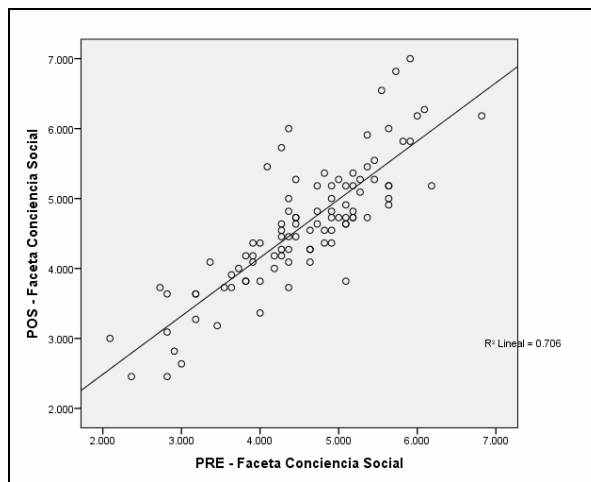


Figura 44. Diagrama de dispersión general de POS-Faceta Conciencia Social contra PRE-Faceta Conciencia Social. Elaboración propia

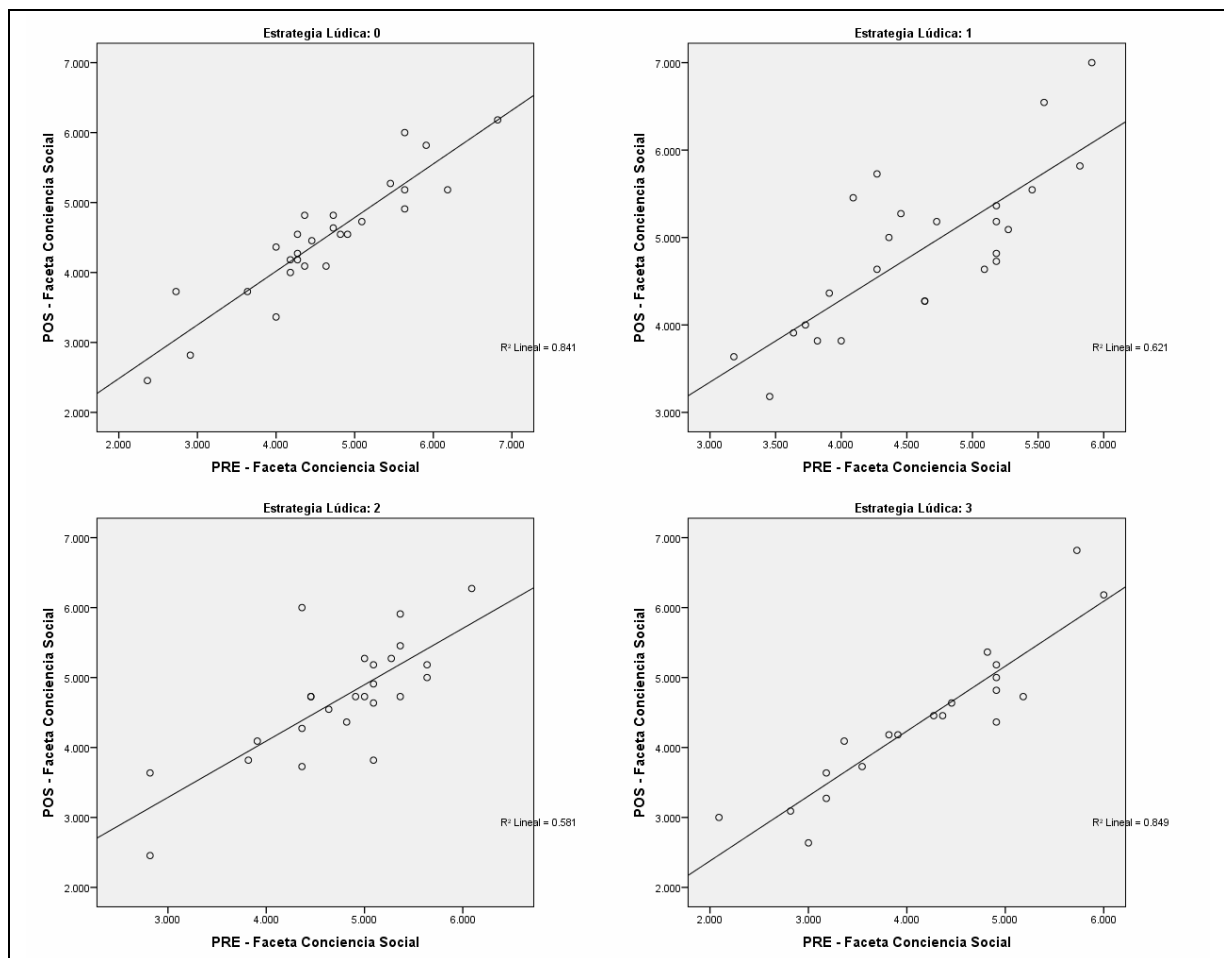


Figura 45. Diagramas de dispersión de POS-Faceta Conciencia Social contra PRE-Faceta Conciencia Social separados según Estrategia Lúdica. Elaboración propia



Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Conciencia Social* contra *PRE-Faceta Conciencia Social* separados según *Estrategia Lúdica* (figura 45) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Conciencia Social* por *Estrategia Lúdica* [ $F(3; 89) = 0,651; p=0,585$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 76.

**Tabla 76**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Conciencia Social; Covariable: PRE-Faceta Conciencia Social)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	58,205 <sup>a</sup>	4	14,551	63,234	0,000	0,733	252,936	1,000
Intersección	2064,078	1	2064,078	8969,582	0,000	0,990	8969,582	1,000
PRE	56,028	1	56,028	243,474	0,000	0,726	243,474	1,000
Estrategia Lúdica	2,177	3	0,726	3,154	0,029	0,093	9,462	0,715
Error	21,171	92	0,230					
Total	2143,455	97						
Total corregido	79,377	96						

a. R al cuadrado = 0,733 (R al cuadrado ajustada = 0,722)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Conciencia Social* sobre la variable dependiente *POS-Faceta Conciencia Social* [ $F_{PRE-Faceta Conciencia Social} (1; 92) = 243,474; p<0,001; \eta^2=0,726; potencia observada=1,000$ ]. Además, hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Conciencia Social* [ $F_{Estrategia Lúdica} (3; 92) = 3,154; p=0,029; \eta^2=0,093; potencia observada=0,715$ ].

El tamaño del efecto sugirió que el 9,3% de la varianza en la faceta *Conciencia Social* se puede atribuir a la *estrategia lúdica* empleada, una vez eliminadas las diferencias que pudieran existir *a priori* entre los grupos. Con G\*Power 3.1.9.2 se convirtió  $\eta^2$  en un índice  $f$  de Cohen=0,320, determinándose mediante la tabla 61 que se trató de un efecto de tamaño medio a grande.

Por lo tanto, se rechazó la hipótesis nula  $H_0$  y se aceptó la hipótesis alternativa  $H_1$  que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, existen diferencias significativas en el postest de la faceta *Conciencia Social*, una vez eliminada la influencia del pretest.

A continuación se calcularon las medias marginales de la variable dependiente *POS-Faceta Conciencia Social* estimadas con base en la covariable *PRE-Faceta Conciencia Social* (tabla 77).

**Tabla 77**

*Medias marginales de POS-Faceta Conciencia Social estimadas con base en la covariable PRE-Faceta Conciencia Social*

Estrategia Lúdica	Media	Error estándar	Intervalo de confianza al 95%	
			Límite inferior	Límite superior
0	4,436 <sup>a</sup>	0,092	4,253	4,620
1	4,809 <sup>a</sup>	0,096	4,618	4,999
2	4,526 <sup>a</sup>	0,097	4,334	4,718
3	4,715 <sup>a</sup>	0,109	4,498	4,932

a. Las covariables que aparecen en el modelo se evalúan en los valores siguientes: PRE-Faceta Conciencia Social = 4,55014.

*Nota:* Elaboración propia

Por último, se llevó a cabo un análisis post-hoc para evaluar las diferencias de las medias anteriores, entre las parejas formadas en función de la variable independiente *Estrategia Lúdica*. Para controlar el riesgo de que aumente el error de tipo I, se utilizó la corrección de Bonferroni. La tabla 78 muestra los resultados de este análisis.

Tabla 78

Comparación de las medias marginales de POS-Faceta Conciencia Social entre las parejas formadas según Estrategia Lúdica

(I) Estrategia Lúdica	(J) Estrategia Lúdica	Diferencia de medias (I-J)	Error estándar	Sig.	95% de intervalo de confianza para diferencia	
					Límite inferior	Límite superior
0	1	-0,372*	0,133	0,038	-0,731	-0,013
	2	-0,090	0,133	1,000	-0,450	0,270
	3	-0,279	0,143	0,331	-0,666	0,108
1	0	0,372*	0,133	0,038	0,013	0,731
	2	0,282	0,136	0,244	-0,084	0,649
	3	0,093	0,146	1,000	-0,300	0,487
2	0	0,090	0,133	1,000	-0,270	0,450
	1	-0,282	0,136	0,244	-0,649	0,084
	3	-0,189	0,147	1,000	-0,586	0,209
3	0	0,279	0,143	0,331	-0,108	0,666
	1	-0,093	0,146	1,000	-0,487	0,300
	2	0,189	0,147	1,000	-0,209	0,586

\*. La diferencia de medias es significativa en el nivel 0,05.

Nota: Elaboración propia

Se observaron diferencias estadísticamente significativas ( $p=0,038$ ) entre las medias marginales estimadas del grupo de control y las del grupo en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*). En la figura 46 puede observarse que, tal como se había mostrado en la tabla 77, el grupo en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*) es el que tiene el mayor valor en la faceta *Conciencia Social* (lo cual es positivo para este grupo, desde un punto de vista psicopedagógico).

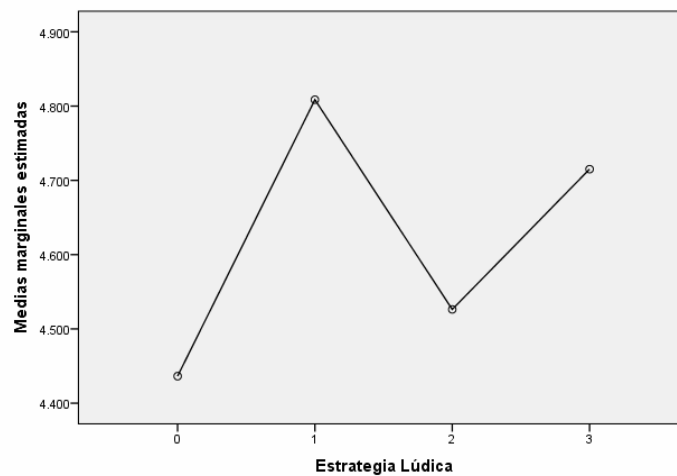


Figura 46. Medias marginales estimadas de POS-Faceta Conciencia Social. Elaboración propia

## Faceta Control de la Impulsividad

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Control de la Impulsividad*, una vez eliminada la influencia que la variable *PRE-Faceta Control de la Impulsividad* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Control de la Impulsividad*, una vez eliminada la influencia que la variable *PRE-Faceta Control de la Impulsividad* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Control de la Impulsividad* en cada grupo seguían o no una distribución normal (tabla 79), constatándose el cumplimiento de este supuesto.

**Tabla 79**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Control de la Impulsividad)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Control de la Impulsividad	0	0,972	27	0,651	Sí
	1	0,971	25	0,667	Sí
	2	0,972	25	0,690	Sí
	3	0,933	20	0,176	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 2,924; p=0,038$ ] mostró que no se cumplía este supuesto, lo que obligó a llevar a cabo la versión no paramétrica de la prueba ANCOVA de un factor. Para convertir las variables *PRE-Faceta Control de la Impulsividad* y *POS-Faceta Control de la Impulsividad* en rangos se crearon dos nuevas variables (*Rango de PRE-Faceta Control de la Impulsividad* y *Rango de POS-Faceta Control de la Impulsividad*), cuyos estadísticos descriptivos se muestran en la tabla 80. Los valores obtenidos para ANCOVA se presentan en la tabla 81.

**Tabla 80**

*Estadísticos de Rango de PRE-Faceta Control de la Impulsividad y Rango de POS-Faceta Control de la Impulsividad*

Variable	Estrategia Lúdica	Estadístico		
		Media	Desviación estándar	N
Rango de PRE-Faceta Control de la Impulsividad	0	51,92593	29,686083	27
	1	48,72000	30,678467	25
	2	47,38000	27,423484	25
	3	47,42500	25,140801	20
Rango de POS-Faceta Control de la Impulsividad	0	51,51852	29,792707	27
	1	49,18000	29,864234	25
	2	40,86000	25,405675	25
	3	55,55000	26,352319	20

*Nota:* Elaboración propia

**Tabla 81**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Control de la Impulsividad;*

*Covariable: Rango de PRE-Faceta Control de la Impulsividad)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	37142,304 <sup>a</sup>	4	9285,576	22,067	0,000	0,490	88,269	1,000
Intersección	232897,000	1	232897,000	553,483	0,000	0,857	553,483	1,000
Rango de PRE	34730,547	1	34730,547	82,538	0,000	0,473	82,538	1,000
Estrategia Lúdica	2411,757	3	803,919	1,911	0,133	0,059	5,732	0,479
Error	38712,196	92	420,785					
Total	308751,500	97						
Total corregido	75854,500	96						

a. R al cuadrado = 0,490 (R al cuadrado ajustada = 0,467)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto significativo de la covariable *Rango de PRE-Faceta Control de la Impulsividad* sobre *Rango de POS-Faceta Control de la Impulsividad* ( $F_{\text{Rango de PRE-Faceta Control de la Impulsividad}} (1; 92) = 82,538; p < 0,001; \eta^2 = 0,473; \text{potencia observada} = 1,000$ ). Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *Rango de POS-Faceta Control de la Impulsividad* [ $F_{\text{Estrategia Lúdica}} (3; 92) = 1,911; p = 0,133; \eta^2 = 0,059; \text{potencia observada} = 0,479$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados según la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Control de la Impulsividad*, una vez eliminada la influencia del pretest.

## Faceta Empatía

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Empatía*, una vez eliminada la influencia que la variable *PRE-Faceta Empatía* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Empatía*, una vez eliminada la influencia que la variable *PRE-Faceta Empatía* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Empatía* en cada grupo seguían o no una distribución normal (tabla 82), constatándose el cumplimiento de este supuesto.

**Tabla 82**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Empatía)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Empatía	0	0,971	27	0,618	Sí
	1	0,980	25	0,890	Sí
	2	0,966	25	0,550	Sí
	3	0,931	20	0,165	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,878; p=0,139$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Empatía* (variable dependiente) sobre *PRE-Faceta Empatía* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Empatía* contra *PRE-Faceta Empatía*: uno general (figura 47) y los otros cuatro separados según la variable independiente (figura 48).

Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

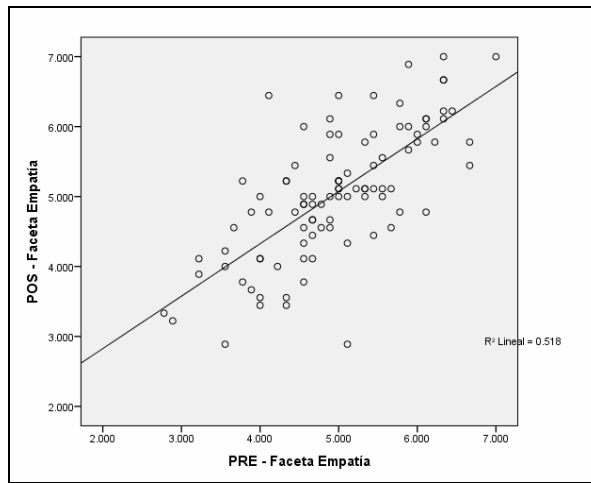


Figura 47. Diagrama de dispersión general de *POS-Faceta Empatía* contra *PRE-Faceta Empatía*.

Elaboración propia

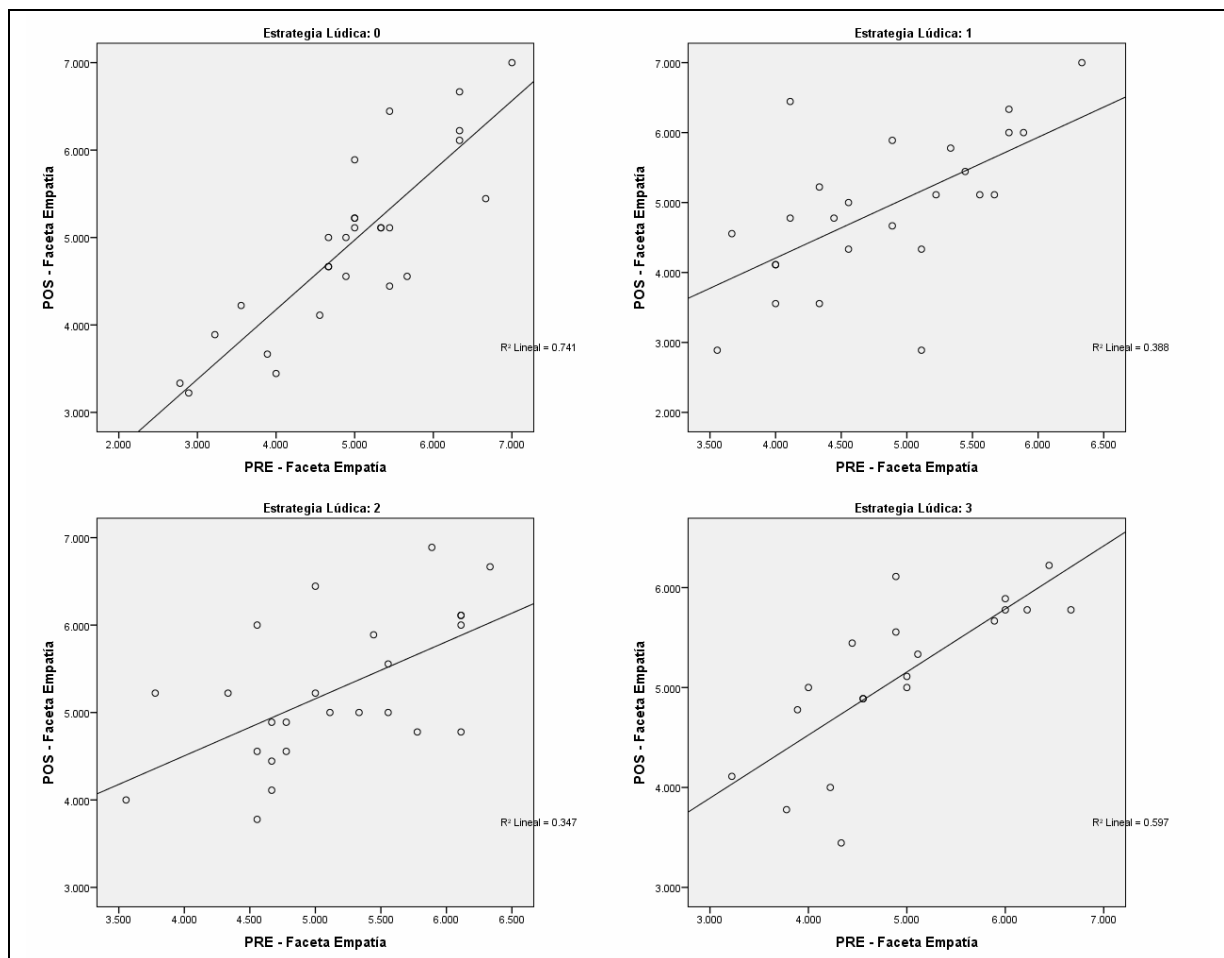


Figura 48. Diagramas de dispersión de *POS-Faceta Empatía* contra *PRE-Faceta Empatía*

separados según *Estrategia Lúdica*. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Empatía* contra *PRE-Faceta Empatía* separados según *Estrategia Lúdica* (figura 48) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Empatía* por *Estrategia Lúdica* [ $F(3; 89) = 0,474$ ;  $p=0,701$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 83.

**Tabla 83**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Empatía; Covariable: PRE-Faceta Empatía)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	44,189 <sup>a</sup>	4	11,047	25,464	0,000	0,525	101,858	1,000
Intersección	2476,380	1	2476,380	5708,197	0,000	0,984	5708,197	1,000
PRE	43,605	1	43,605	100,513	0,000	0,522	100,513	1,000
Estrategia Lúdica	0,583	3	0,194	0,448	0,719	0,014	1,345	0,137
Error	39,912	92	0,434					
Total	2560,481	97						
Total corregido	84,101	96						

a. R al cuadrado = 0,525 (R al cuadrado ajustada = 0,505)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Empatía* sobre la variable dependiente *POS-Faceta Empatía* [ $F_{PRE-Faceta Empatía} (1; 92) = 100,513$ ;  $p<0,001$ ;  $\eta^2=0,522$ ; potencia observada=1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Empatía* [ $F_{Estrategia Lúdica} (3; 92) = 0,448$ ;  $p=0,719$ ;  $\eta^2=0,014$ ; potencia observada=0,137].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Empatía*, una vez eliminada la influencia del pretest.



### Faceta Expresividad Emocional

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Expresividad Emocional*, una vez eliminada la influencia que la variable *PRE-Faceta Expresividad Emocional* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Expresividad Emocional*, una vez eliminada la influencia que la variable *PRE-Faceta Expresividad Emocional* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Expresividad Emocional* en cada grupo seguían o no una distribución normal (tabla 84).

**Tabla 84**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Expresividad Emocional)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Expresividad Emocional	0	0,981	27	0,892	Sí
	1	0,958	25	0,375	Sí
	2	0,985	25	0,959	Sí
	3	0,899	20	0,040	No

*Nota:* Elaboración propia

La violación del supuesto de normalidad obligó a llevar a cabo la versión no paramétrica de la prueba ANCOVA de un factor. Para convertir las variables *PRE-Faceta Expresividad Emocional* y *POS-Faceta Expresividad Emocional* en rangos se crearon dos nuevas variables (*Rango de PRE-Faceta Expresividad Emocional* y *Rango de POS-Faceta Expresividad Emocional*), cuyos estadísticos descriptivos se muestran en la tabla 85.

Tabla 85

Estadísticos de Rango de PRE-Faceta Expresividad Emocional y Rango de POS-Faceta Expresividad Emocional

Variable	Estrategia Lúdica	Estadístico		
		Media	Desviación estándar	N
Rango de PRE-Faceta Expresividad Emocional	0	49,72222	30,235719	27
	1	58,30000	27,353778	25
	2	52,56000	25,010131	25
	3	31,95000	23,953848	20
Rango de POS-Faceta Expresividad Emocional	0	48,51852	29,717090	27
	1	54,02000	26,554770	25
	2	52,08000	27,793015	25
	3	39,52500	27,971073	20

Nota: Elaboración propia

Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 86.

Tabla 86

Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Expresividad Emocional;

Covariable: Rango de PRE-Faceta Expresividad Emocional)

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	46110,407 <sup>a</sup>	4	11527,602	35,532	0,000	0,607	142,130	1,000
Intersección	232897,000	1	232897,000	717,876	0,000	0,886	717,876	1,000
Rango de PRE	45632,234	1	45632,234	140,656	0,000	0,605	140,656	1,000
Estrategia Lúdica	478,173	3	159,391	0,491	0,689	0,016	1,474	0,146
Error	29847,093	92	324,425					
Total	308854,500	97						
Total corregido	75957,500	96						

a. R al cuadrado = 0,607 (R al cuadrado ajustada = 0,590)

b. Se ha calculado utilizando alpha = 0,05

Nota: Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto significativo de la covariable *Rango de PRE-Faceta Expresividad Emocional* sobre la variable dependiente *Rango de POS-Faceta Expresividad Emocional* ( $F_{\text{Rango de PRE-Faceta Expresividad Emocional}}(1; 92) = 140,656$ ;  $p < 0,001$ ;  $\eta^2 = 0,605$ ; potencia observada=1,000). Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *Rango de POS-Faceta Expresividad Emocional* [ $F_{\text{Estrategia Lúdica}}(3; 92) = 0,491$ ;  $p = 0,689$ ;  $\eta^2 = 0,016$ ; potencia observada=0,146].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados según la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Expresividad Emocional*, una vez eliminada la influencia del pretest.

### Faceta Felicidad

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- H<sub>0</sub>: Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Felicidad*, una vez eliminada la influencia que la variable *PRE-Faceta Felicidad* pudiera ejercer sobre esta.
- H<sub>1</sub>: Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Felicidad*, una vez eliminada la influencia que la variable *PRE-Faceta Felicidad* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Felicidad* en cada grupo seguían o no una distribución normal (tabla 87).

**Tabla 87**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Felicidad)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad (p ≥ 0,05)
Residuo para POS-Faceta Felicidad	0	0,958	27	0,335	Sí
	1	0,891	25	0,012	No
	2	0,966	25	0,541	Sí
	3	0,942	20	0,257	Sí

*Nota:* Elaboración propia

La violación del supuesto de normalidad obligó a llevar a cabo la versión no paramétrica de la prueba ANCOVA de un factor. Para convertir las variables *PRE-Faceta Felicidad* y *POS-Faceta Felicidad* en rangos se crearon dos nuevas variables (*Rango de PRE-Faceta Felicidad* y *Rango de POS-Faceta Felicidad*), cuyos estadísticos descriptivos se muestran en la tabla 88.

Tabla 88

Estadísticos de Rango de PRE-Faceta Felicidad y Rango de POS-Faceta Felicidad

Variable	Estrategia Lúdica	Estadístico		
		Media	Desviación estándar	N
Rango de PRE-Faceta Felicidad	0	50,98148	26,978439	27
	1	62,02000	26,638584	25
	2	50,02000	24,933863	25
	3	28,77500	25,550505	20
Rango de POS-Faceta Felicidad	0	47,31481	22,968267	27
	1	61,82000	29,286971	25
	2	50,28000	26,673989	25
	3	33,65000	28,574971	20

Nota: Elaboración propia

Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 89.

Tabla 89

Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Felicidad;

Covariable: Rango de PRE-Faceta Felicidad)

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	46337,711 <sup>a</sup>	4	11584,428	36,137	0,000	0,611	144,546	1,000
Intersección	232897,000	1	232897,000	726,500	0,000	0,888	726,500	1,000
Rango de PRE	45861,365	1	45861,365	143,060	0,000	0,609	143,060	1,000
Estrategia Lúdica	476,347	3	158,782	0,495	0,686	0,016	1,486	0,147
Error	29492,789	92	320,574					
Total	308727,500	97						
Total corregido	75830,500	96						

a. R al cuadrado = 0,611 (R al cuadrado ajustada = 0,594)

b. Se ha calculado utilizando alpha = 0,05

Nota: Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *Rango de PRE-Faceta Felicidad* sobre la variable *Rango de POS-Faceta Felicidad* [ $F_{\text{Rango de PRE-Faceta Felicidad}}(1; 92) = 143,060$ ;  $p < 0,001$ ;  $\eta^2 = 0,609$ ; potencia observada = 1,000). Sin embargo, no hubo evidencia de un efecto significativo de la variable independiente *Estrategia Lúdica* sobre *Rango de POS-Faceta Felicidad* [ $F_{\text{Estrategia Lúdica}}(3; 92) = 0,495$ ;  $p = 0,686$ ;  $\eta^2 = 0,016$ ; potencia observada = 0,147].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Felicidad*, una vez eliminada la influencia del pretest.

### Faceta Gestión de la Emoción

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Gestión de la Emoción*, una vez eliminada la influencia que la variable *PRE-Faceta Gestión de la Emoción* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Gestión de la Emoción*, una vez eliminada la influencia que la variable *PRE-Faceta Gestión de la Emoción* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Gestión de la Emoción* en cada grupo seguían o no una distribución normal (tabla 90), constatándose el cumplimiento de este supuesto.

**Tabla 90**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Gestión de la Emoción)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Gestión de la Emoción	0	0,967	27	0,522	Sí
	1	0,975	25	0,763	Sí
	2	0,950	25	0,255	Sí
	3	0,938	20	0,217	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,518; p=0,215$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Gestión de la Emoción* (variable dependiente) sobre *PRE-Faceta Gestión de la Emoción* (covariable) era lineal, se realizaron cinco *scatterplots* de *POS-Faceta Gestión de la Emoción* contra *PRE-Faceta Gestión de la Emoción*: uno general (figura 49) y los otros cuatro separados según la variable

independiente (figura 50). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

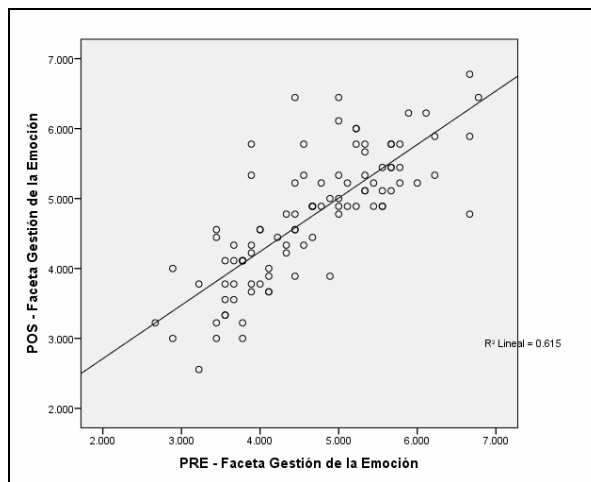


Figura 49. Diagrama de dispersión general de POS-Faceta Gestión de la Emoción contra PRE-Faceta Gestión de la Emoción. Elaboración propia

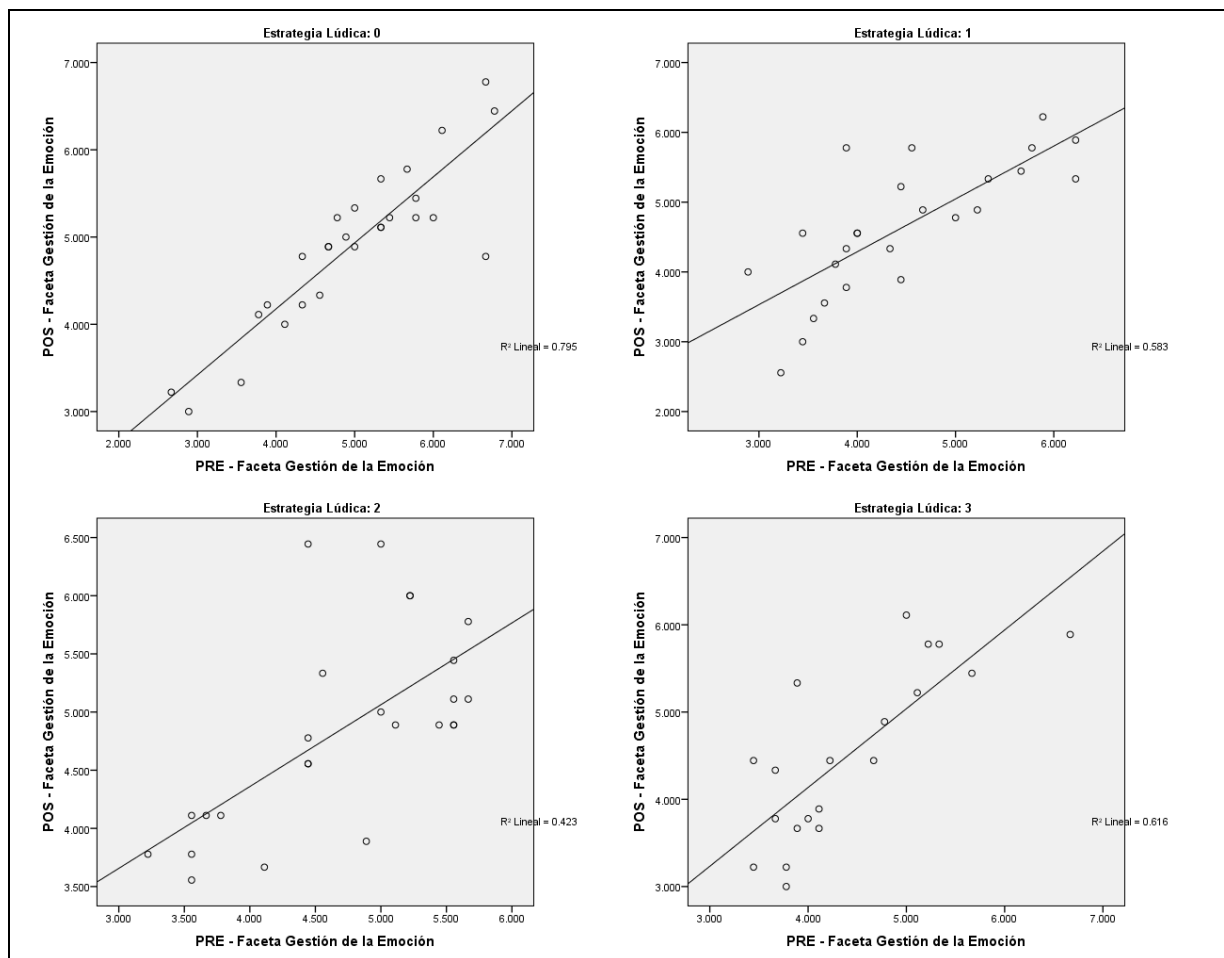


Figura 50. Diagramas de dispersión de POS-Faceta Gestión de la Emoción contra PRE-Faceta Gestión de la Emoción separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Gestión de la Emoción* contra *PRE-Faceta Gestión de la Emoción* separados según *Estrategia Lúdica* (figura 50) mostraron líneas de regresión similares. Por otra parte, se observó una interacción no significativa de *PRE-Faceta Gestión de la Emoción* por *Estrategia Lúdica* [ $F(3; 89) = 0,308$ ;  $p=0,819$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 91.

**Tabla 91**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Gestión de la Emoción; Covariable: PRE-Faceta Gestión de la Emoción)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	50,776 <sup>a</sup>	4	12,694	37,524	0,000	0,620	150,098	1,000
Intersección	2179,336	1	2179,336	6442,270	0,000	0,986	6442,270	1,000
PRE	50,391	1	50,391	148,959	0,000	0,618	148,959	1,000
Estrategia Lúdica	0,385	3	0,128	0,380	0,768	0,012	1,139	0,122
Error	31,122	92	0,338					
Total	2261,235	97						
Total corregido	81,898	96						

a. R al cuadrado = 0,620 (R al cuadrado ajustada = 0,603)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Gestión de la Emoción* sobre la variable dependiente *POS-Faceta Gestión de la Emoción* ( $F_{PRE-Faceta Gestión de la Emoción} (1; 92) = 148,959$ ;  $p<0,001$ ;  $\eta^2=0,618$ ; potencia observada=1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Gestión de la Emoción* [ $F_{Estrategia Lúdica} (3; 92) = 0,380$ ;  $p=0,768$ ;  $\eta^2=0,012$ ; potencia observada=0,122].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Gestión de la Emoción*, una vez eliminada la influencia del pretest.

## Faceta Gestión del Estrés

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Gestión del Estrés*, una vez eliminada la influencia que la variable *PRE-Faceta Gestión del Estrés* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Gestión del Estrés*, una vez eliminada la influencia que la variable *PRE-Faceta Gestión del Estrés* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Gestión del Estrés* en cada grupo seguían o no una distribución normal (tabla 92), constatándose el cumplimiento de este supuesto.

**Tabla 92**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Gestión del Estrés)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Gestión del Estrés	0	0,956	27	0,300	Sí
	1	0,972	25	0,685	Sí
	2	0,946	25	0,204	Sí
	3	0,971	20	0,786	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,664; p=0,180$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Gestión del Estrés* (variable dependiente) sobre *PRE-Faceta Gestión del Estrés* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Gestión del Estrés* contra *PRE-Faceta Gestión del Estrés*: uno general (figura 51) y los otros cuatro separados según la



variable independiente (figura 52). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

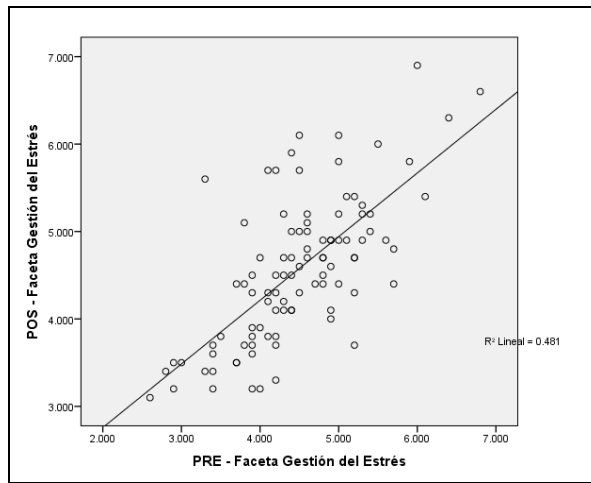


Figura 51. Diagrama de dispersión general de *POS-Faceta Gestión del Estrés* contra *PRE-Faceta Gestión del Estrés*. Elaboración propia

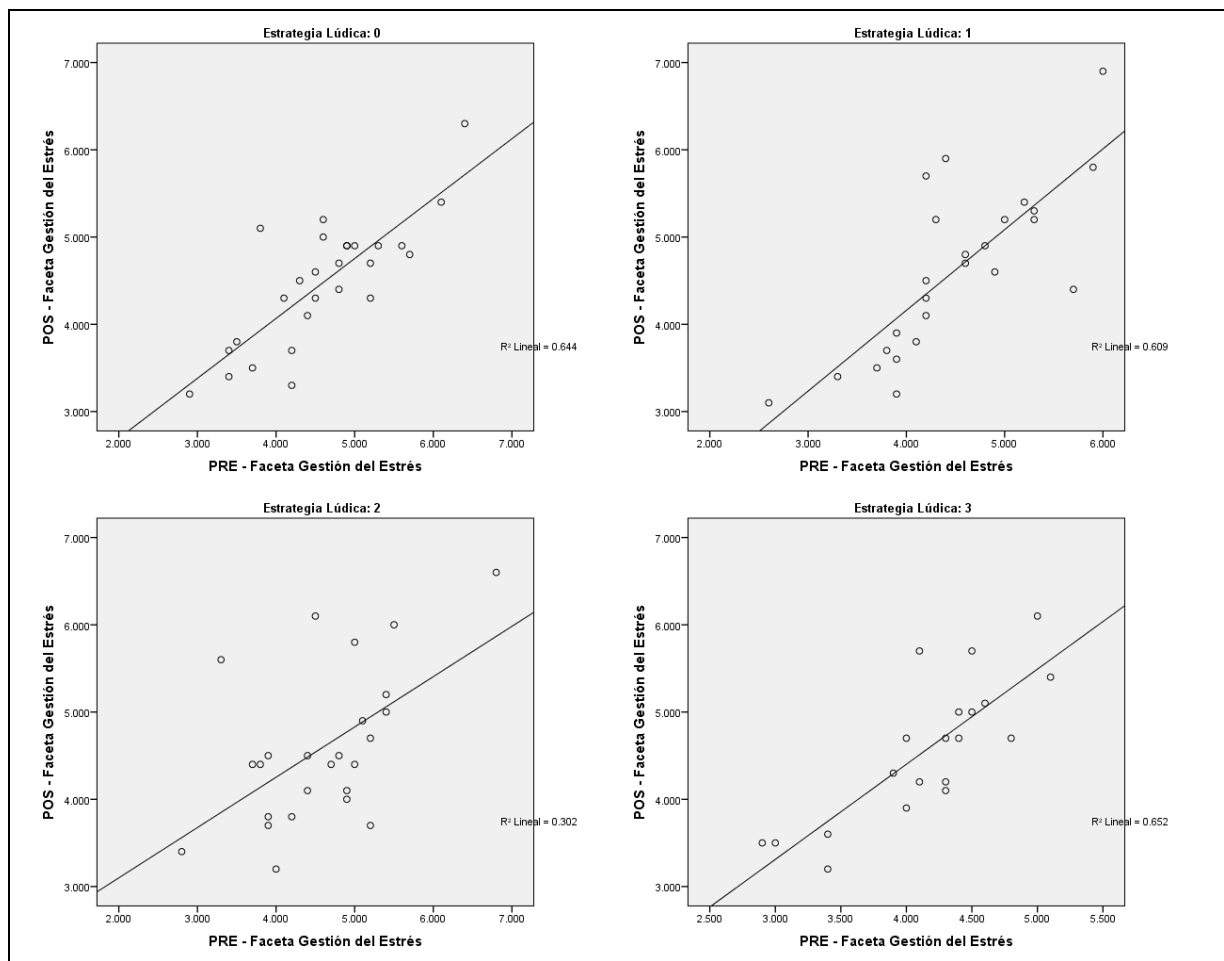


Figura 52. Diagramas de dispersión de *POS-Faceta Gestión del Estrés* contra *PRE-Faceta Gestión del Estrés* separados según *Estrategia Lúdica*. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Gestión del Estrés* contra *PRE-Faceta Gestión del Estrés* separados según *Estrategia Lúdica* (figura 52) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Gestión del Estrés* por *Estrategia Lúdica* [ $F(3; 89) = 1,771; p=0,158$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 93.

**Tabla 93**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Gestión del Estrés; Covariable: PRE-Faceta Gestión del Estrés)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	34,349 <sup>a</sup>	4	8,587	24,197	0,000	0,513	96,788	1,000
Intersección	2014,062	1	2014,062	5675,233	0,000	0,984	5675,233	1,000
PRE	32,196	1	32,196	90,721	0,000	0,497	90,721	1,000
Estrategia Lúdica	2,153	3	0,718	2,022	0,116	0,062	6,067	0,504
Error	32,650	92	0,355					
Total	2081,060	97						
Total corregido	66,998	96						

a. R al cuadrado = 0,513 (R al cuadrado ajustada = 0,491)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Gestión del Estrés* sobre la variable dependiente *POS-Faceta Gestión del Estrés* [ $F_{PRE-Faceta Gestión del Estrés} (1; 92) = 90,721; p<0,001; \eta^2=0,497; potencia observada=1,000$ ]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Gestión del Estrés* [ $F_{Estrategia Lúdica} (3; 92) = 2,022; p=0,116; \eta^2=0,062; potencia observada=0,504$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Gestión del Estrés*, una vez eliminada la influencia del pretest.

## Faceta Optimismo

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Optimismo*, una vez eliminada la influencia que la variable *PRE-Faceta Optimismo* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Optimismo*, una vez eliminada la influencia que la variable *PRE-Faceta Optimismo* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Optimismo* en cada grupo seguían o no una distribución normal (tabla 94).

**Tabla 94**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Optimismo)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Optimismo	0	0,959	27	0,348	Sí
	1	0,898	25	0,016	No
	2	0,953	25	0,286	Sí
	3	0,912	20	0,069	Sí

*Nota:* Elaboración propia

La violación del supuesto de normalidad obligó a llevar a cabo la versión no paramétrica de la prueba ANCOVA de un factor. Para convertir las variables *PRE-Faceta Optimismo* y *POS-Faceta Optimismo* en rangos se crearon dos nuevas variables (*Rango de PRE-Faceta Optimismo* y *Rango de POS-Faceta Optimismo*), cuyos estadísticos descriptivos se muestran en la tabla 95.

Tabla 95

Estadísticos de Rango de PRE-Faceta Optimismo y Rango de POS-Faceta Optimismo

Variable	Estrategia Lúdica	Estadístico		
		Media	Desviación estándar	N
Rango de PRE-Faceta Optimismo	0	56,07407	28,232109	27
	1	56,50000	28,494883	25
	2	47,90000	23,487585	25
	3	31,45000	26,673908	20
Rango de POS-Faceta Optimismo	0	48,92593	25,667138	27
	1	55,96000	27,110084	25
	2	52,54000	24,986213	25
	3	35,97500	33,435907	20

Nota: Elaboración propia

Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 96.

Tabla 96

Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Faceta Optimismo;

Covariable: Rango de PRE-Faceta Optimismo)

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	52588,539 <sup>a</sup>	4	13147,135	51,864	0,000	0,693	207,455	1,000
Intersección	232897,000	1	232897,000	918,747	0,000	0,909	918,747	1,000
Rango de PRE	51072,039	1	51072,039	201,472	0,000	0,687	201,472	1,000
Estrategia Lúdica	1516,500	3	505,500	1,994	0,120	0,061	5,982	0,498
Error	23321,461	92	253,494					
Total	308807,000	97						
Total corregido	75910,000	96						

a. R al cuadrado = 0,693 (R al cuadrado ajustada = 0,679)

b. Se ha calculado utilizando alpha = 0,05

Nota: Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto significativo de la covariable *Rango de PRE-Faceta Optimismo* sobre la variable *Rango de POS-Faceta Optimismo* [ $F_{\text{Rango de PRE-Faceta Optimismo}}(1; 92) = 201,472; p < 0,001; \eta^2 = 0,687; \text{potencia observada} = 1,000$ ). Sin embargo, no hubo evidencia de un efecto significativo de la variable independiente *Estrategia Lúdica* sobre *Rango de POS-Faceta Optimismo* [ $F_{\text{Estrategia Lúdica}}(3; 92) = 1,994; p = 0,120; \eta^2 = 0,061; \text{potencia observada} = 0,498$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Optimismo*, una vez eliminada la influencia del pretest.

## Faceta Percepción Emocional

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Percepción Emocional*, una vez eliminada la influencia que la variable *PRE-Faceta Percepción Emocional* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Percepción Emocional*, una vez eliminada la influencia que la variable *PRE-Faceta Percepción Emocional* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Percepción Emocional* en cada grupo seguían o no una distribución normal (tabla 97), constatándose el cumplimiento de este supuesto.

**Tabla 97**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Percepción Emocional)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Percepción Emocional	0	0,967	27	0,521	Sí
	1	0,975	25	0,766	Sí
	2	0,961	25	0,429	Sí
	3	0,939	20	0,233	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 0,565$ ;  $p=0,640$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Percepción Emocional* (variable dependiente) sobre *PRE-Faceta Percepción Emocional* (covariable) era lineal, se realizaron cinco *scatterplots* de *POS-Faceta Percepción Emocional* contra *PRE-Faceta Percepción Emocional*: uno general (figura 53) y los otros cuatro separados según la variable

independiente (figura 54). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

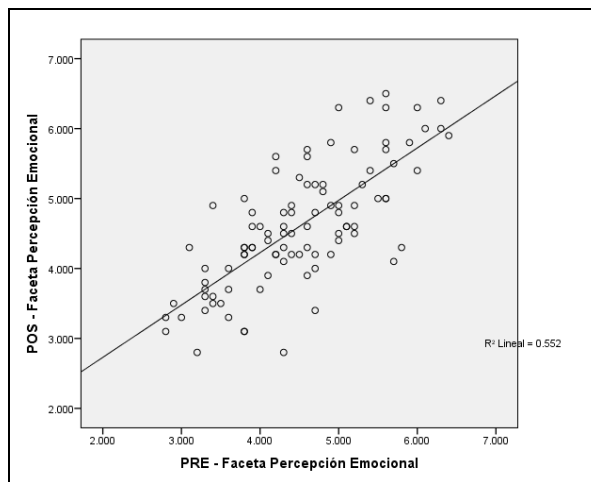


Figura 53. Diagrama de dispersión general de POS-Faceta Percepción Emocional contra PRE-Faceta Percepción Emocional. Elaboración propia

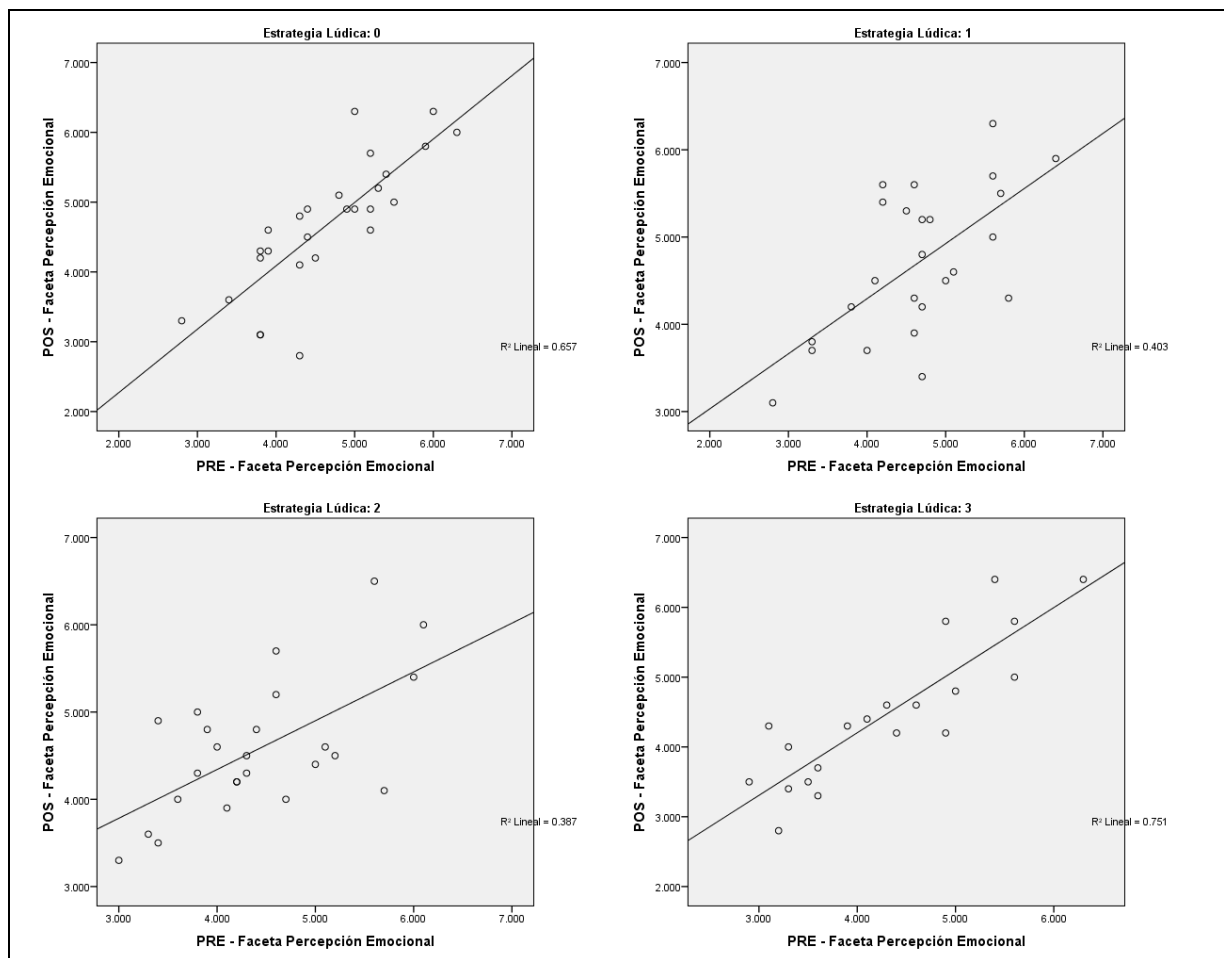


Figura 54. Diagramas de dispersión de POS-Faceta Percepción Emocional contra PRE-Faceta Percepción Emocional separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Percepción Emocional* contra *PRE-Faceta Percepción Emocional* separados según *Estrategia Lúdica* (figura 54) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Percepción Emocional* por *Estrategia Lúdica* [ $F(3; 89) = 1,621; p=0,190$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 98.

**Tabla 98**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Percepción Emocional; Covariable: PRE-Faceta Percepción Emocional)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	41,560 <sup>a</sup>	4	10,390	28,513	0,000	0,554	114,052	1,000
Intersección	2058,965	1	2058,965	5650,323	0,000	0,984	5650,323	1,000
PRE	41,478	1	41,478	113,825	0,000	0,553	113,825	1,000
Estrategia Lúdica	0,083	3	0,028	0,076	0,973	0,002	0,227	0,063
Error	33,525	92	0,364					
Total	2134,050	97						
Total corregido	75,085	96						

a. R al cuadrado = 0,554 (R al cuadrado ajustada = 0,534)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Percepción Emocional* sobre la variable dependiente *POS-Faceta Percepción Emocional* [ $F_{PRE-Faceta Percepción Emocional} (1; 92) = 113,825; p<0,001; \eta^2=0,553; potencia observada=1,000$ ]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Percepción Emocional* [ $F_{Estrategia Lúdica} (3; 92) = 0,076; p=0,973; \eta^2=0,002; potencia observada=0,063$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Percepción Emocional*, una vez eliminada la influencia del pretest.

## Faceta Regulación Emocional

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Regulación Emocional*, una vez eliminada la influencia que la variable *PRE-Faceta Regulación Emocional* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Regulación Emocional*, una vez eliminada la influencia que la variable *PRE-Faceta Regulación Emocional* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Regulación Emocional* en cada grupo seguían o no una distribución normal (tabla 99), constatándose el cumplimiento de este supuesto.

**Tabla 99**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Regulación Emocional)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Regulación Emocional	0	0,970	27	0,603	Sí
	1	0,963	25	0,469	Sí
	2	0,973	25	0,720	Sí
	3	0,959	20	0,517	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 0,044$ ;  $p=0,988$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Regulación Emocional* (variable dependiente) sobre *PRE-Faceta Regulación Emocional* (covariable) era lineal, se realizaron cinco *scatterplots* de *POS-Faceta Regulación Emocional* contra *PRE-Faceta Regulación Emocional*: uno general (figura 55) y los otros cuatro separados según la variable



independiente (figura 56). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

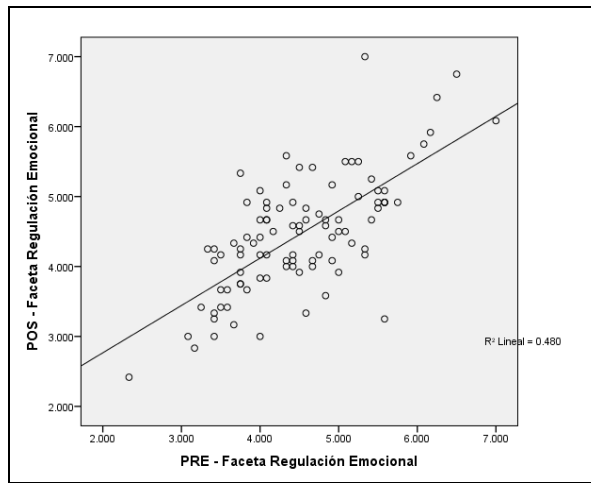


Figura 55. Diagrama de dispersión general de *POS-Faceta Regulación Emocional* contra *PRE-Faceta Regulación Emocional*. Elaboración propia

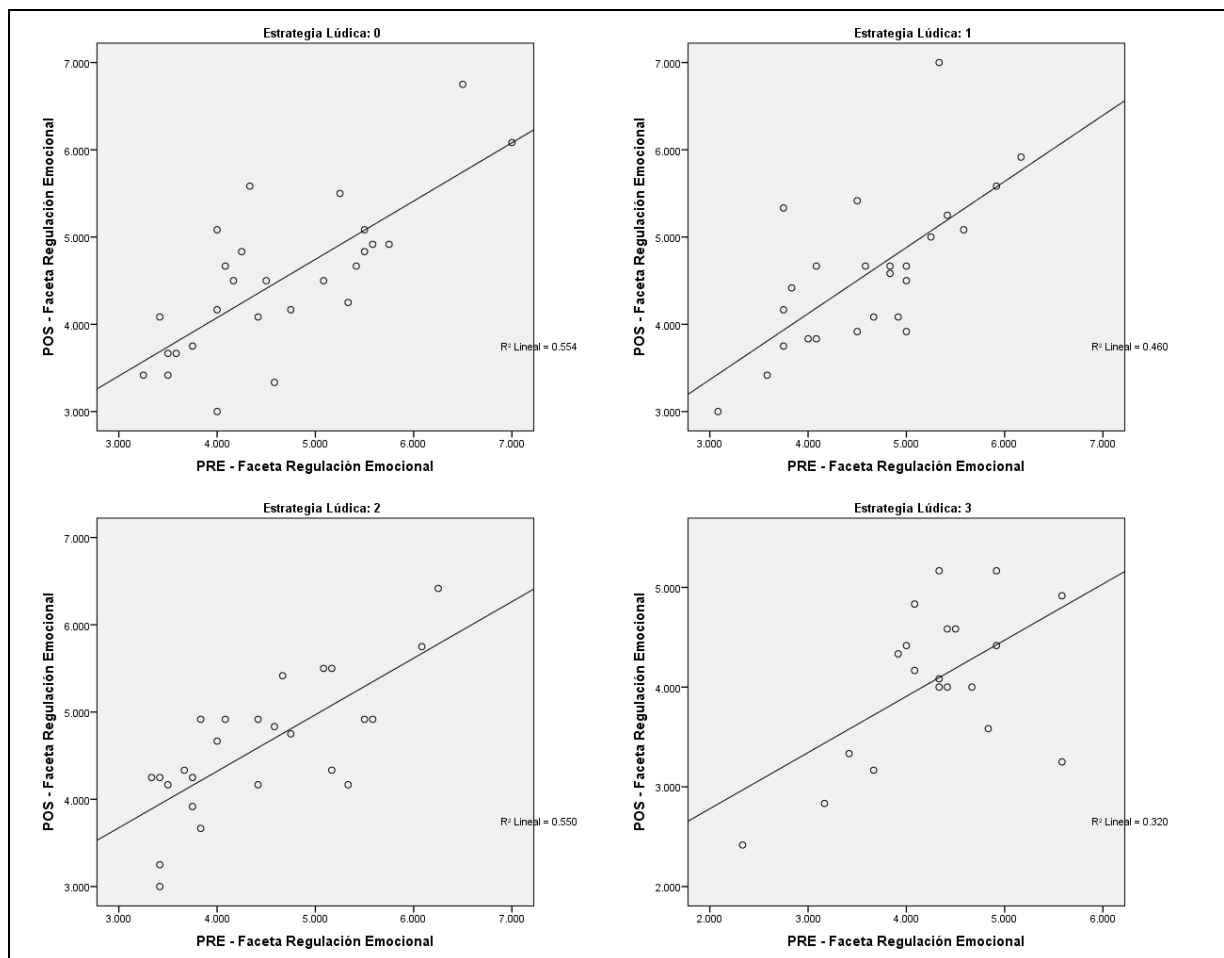


Figura 56. Diagramas de dispersión de *POS-Faceta Regulación Emocional* contra *PRE-Faceta Regulación Emocional* separados según *Estrategia Lúdica*. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Regulación Emocional* contra *PRE-Faceta Regulación Emocional* separados según *Estrategia Lúdica* (figura 56) mostraron líneas de regresión similares. Por otra parte, se observó una interacción no significativa de *PRE-Faceta Regulación Emocional* por *Estrategia Lúdica* [ $F(3; 89) = 0,223$ ;  $p=0,880$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 100.

**Tabla 100**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Regulación Emocional; Covariable: PRE-Faceta Regulación Emocional)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	34,478 <sup>a</sup>	4	8,620	24,248	0,000	0,513	96,993	1,000
Intersección	1929,158	1	1929,158	5426,989	0,000	0,983	5426,989	1,000
PRE	32,255	1	32,255	90,737	0,000	0,497	90,737	1,000
Estrategia Lúdica	2,224	3	0,741	2,085	0,108	0,064	6,256	0,518
Error	32,704	92	0,355					
Total	1996,340	97						
Total corregido	67,182	96						

a. R al cuadrado = 0,513 (R al cuadrado ajustada = 0,492)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Regulación Emocional* sobre la variable dependiente *POS-Faceta Regulación Emocional* ( $F_{PRE-Faceta Regulación Emocional} (1; 92) = 90,737$ ;  $p < 0,001$ ;  $\eta^2 = 0,497$ ; potencia observada = 1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Regulación Emocional* [ $F_{Estrategia Lúdica} (3; 92) = 2,085$ ;  $p = 0,108$ ;  $\eta^2 = 0,064$ ; potencia observada = 0,518].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Regulación Emocional*, una vez eliminada la influencia del pretest.

## Faceta Relaciones

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Faceta Relaciones*, una vez eliminada la influencia que la variable *PRE-Faceta Relaciones* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Faceta Relaciones*, una vez eliminada la influencia que la variable *PRE-Faceta Relaciones* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Faceta Relaciones* en cada grupo seguían o no una distribución normal (tabla 101), constatándose el cumplimiento de este supuesto.

**Tabla 101**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Faceta Relaciones)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Faceta Relaciones	0	0,991	27	0,997	Sí
	1	0,971	25	0,678	Sí
	2	0,978	25	0,849	Sí
	3	0,952	20	0,396	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,136; p=0,339$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Faceta Relaciones* (variable dependiente) sobre *PRE-Faceta Relaciones* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Faceta Relaciones* contra *PRE-Faceta Relaciones*: uno general (figura 57) y los otros cuatro separados según la variable independiente

(figura 58). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

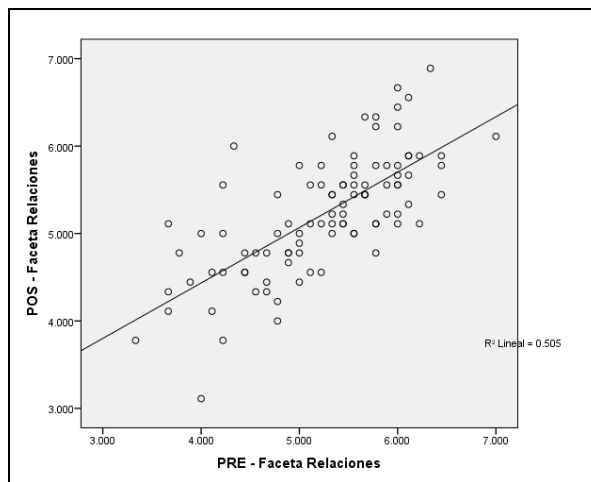


Figura 57. Diagrama de dispersión general de POS-Faceta Relaciones contra PRE-Faceta Relaciones. Elaboración propia

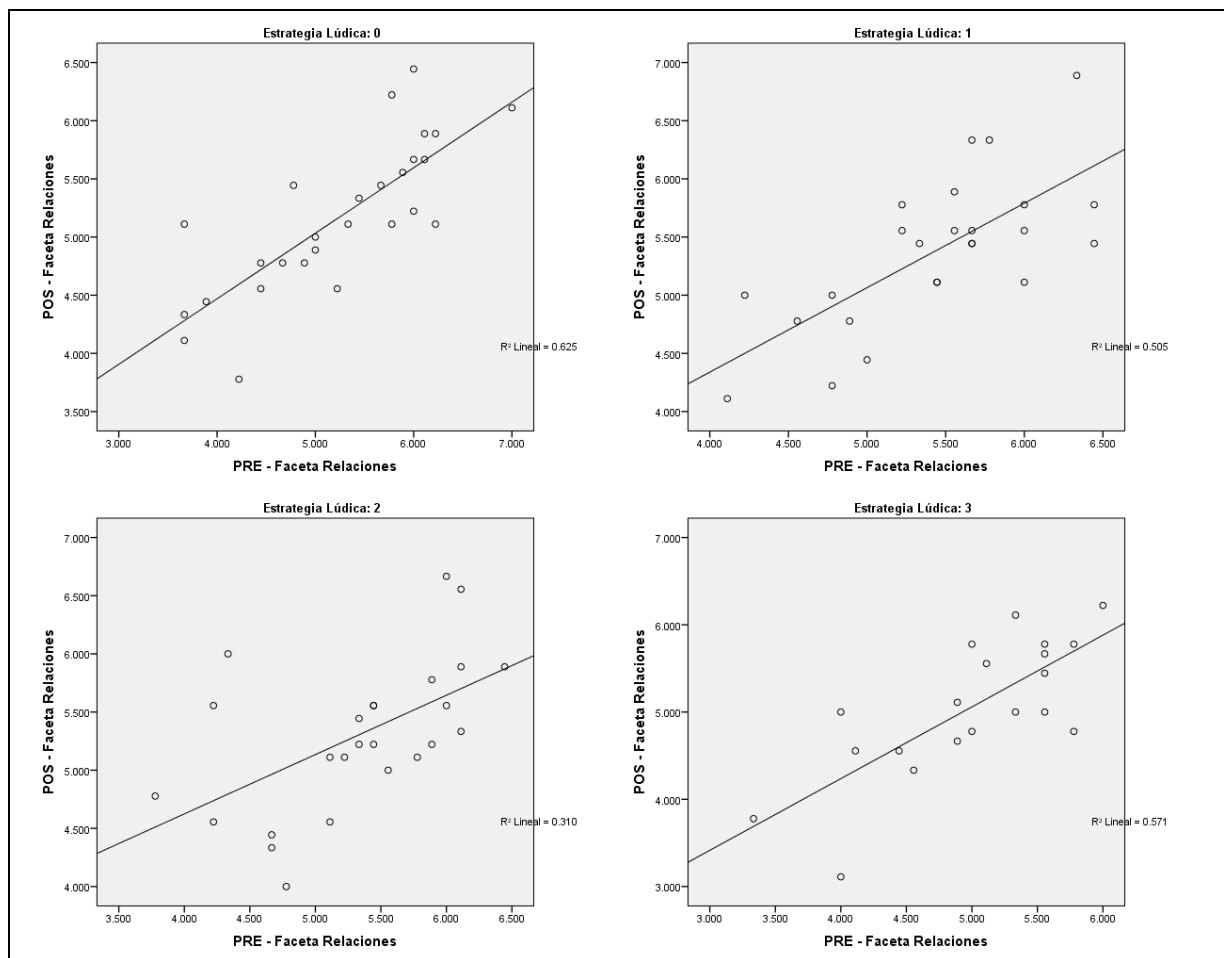


Figura 58. Diagramas de dispersión de POS-Faceta Relaciones contra PRE-Faceta Relaciones separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Faceta Relaciones* contra *PRE-Faceta Relaciones* separados según *Estrategia Lúdica* (figura 58) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Faceta Relaciones* por *Estrategia Lúdica* [ $F(3; 89) = 1,014; p=0,391$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 102.

**Tabla 102**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Faceta Relaciones;*

*Covariable: PRE-Faceta Relaciones)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	22,451 <sup>a</sup>	4	5,613	23,724	0,000	0,508	94,896	1,000
Intersección	2652,313	1	2652,313	11210,590	0,000	0,992	11210,590	1,000
PRE	22,326	1	22,326	94,364	0,000	0,506	94,364	1,000
Estrategia Lúdica	0,126	3	0,042	0,177	0,912	0,006	0,532	0,082
Error	21,766	92	0,237					
Total	2696,531	97						
Total corregido	44,218	96						

a. R al cuadrado = 0,508 (R al cuadrado ajustada = 0,486)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Faceta Relaciones* sobre la variable dependiente *POS-Faceta Relaciones* [ $F_{PRE-Faceta Relaciones} (1; 92) = 94,364; p<0,001; \eta^2=0,506; potencia observada=1,000$ ]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Faceta Relaciones* [ $F_{Estrategia Lúdica} (3; 92) = 0,177; p=0,912; \eta^2=0,006; potencia observada=0,082$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest de la faceta *Relaciones*, una vez eliminada la influencia del pretest.

## Factor Autocontrol

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Factor Autocontrol*, una vez eliminada la influencia que la variable *PRE-Factor Autocontrol* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Factor Autocontrol*, una vez eliminada la influencia que la variable *PRE-Factor Autocontrol* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Factor Autocontrol* en cada grupo seguían o no una distribución normal (tabla 103), constatándose el cumplimiento de este supuesto.

**Tabla 103**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Autocontrol)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Factor Autocontrol	0	0,956	27	0,293	Sí
	1	0,970	25	0,657	Sí
	2	0,965	25	0,531	Sí
	3	0,965	20	0,658	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 0,941; p=0,424$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Factor Autocontrol* (variable dependiente) sobre *PRE-Factor Autocontrol* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Factor Autocontrol* contra *PRE-Factor Autocontrol*: uno general (figura 59) y los otros cuatro separados según la variable independiente

(figura 60). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

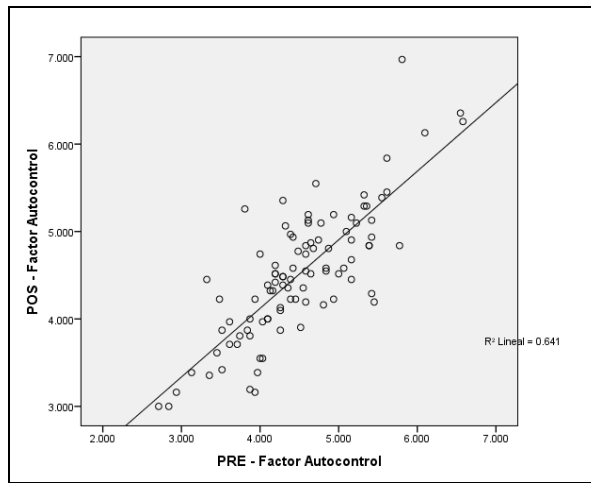


Figura 59. Diagrama de dispersión general de *POS-Factor Autocontrol* contra *PRE-Factor Autocontrol*. Elaboración propia

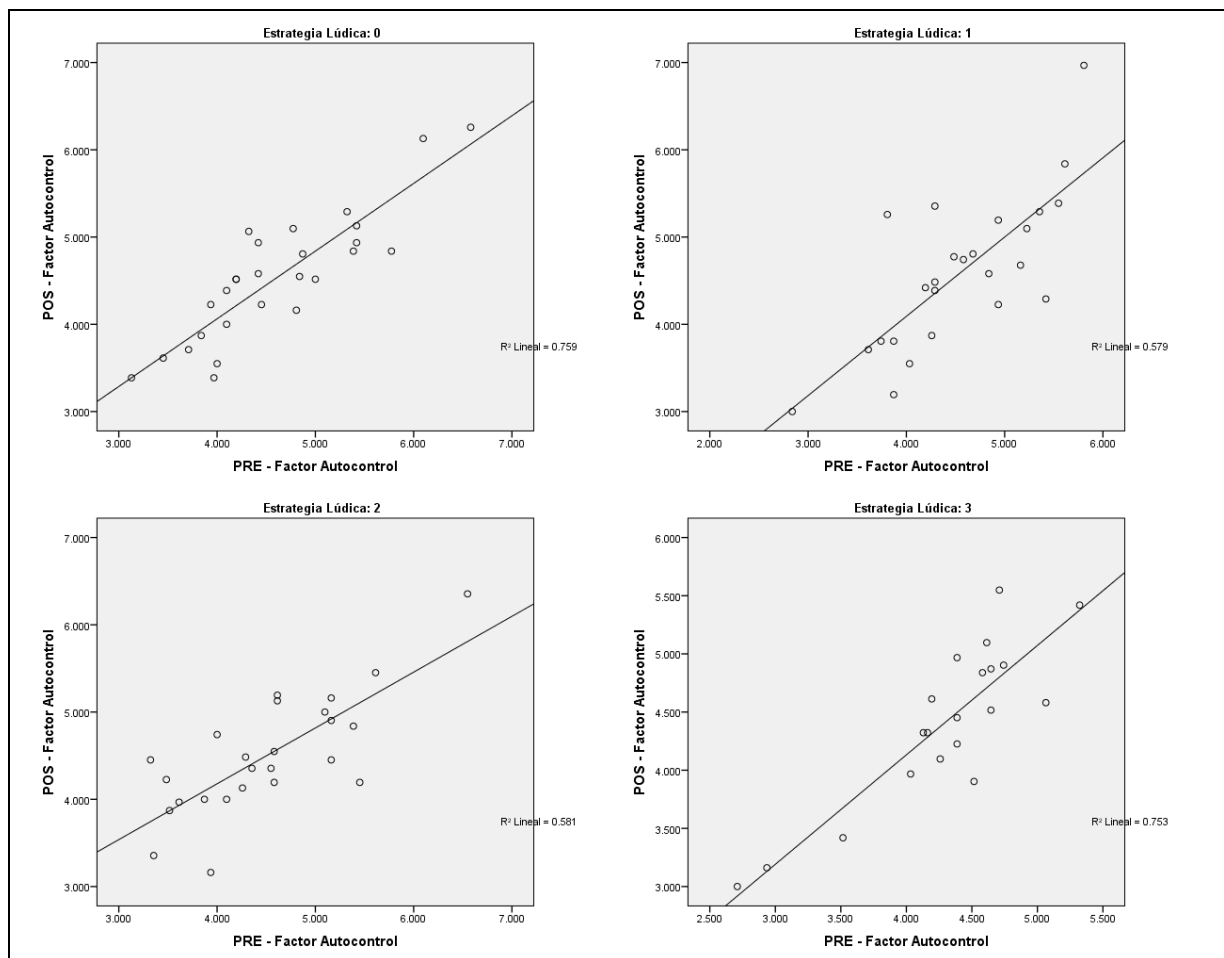


Figura 60. Diagramas de dispersión de *POS-Factor Autocontrol* contra *PRE-Factor Autocontrol* separados según *Estrategia Lúdica*. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Factor Autocontrol* contra *PRE-Factor Autocontrol* separados según *Estrategia Lúdica* (figura 60) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Factor Autocontrol* por *Estrategia Lúdica* [ $F(3; 89) = 1,174; p=0,324$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 104.

**Tabla 104**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Autocontrol; Covariable: PRE-Factor Autocontrol)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	33,967 <sup>a</sup>	4	8,492	41,891	0,000	0,646	167,564	1,000
Intersección	1977,482	1	1977,482	9755,090	0,000	0,991	9755,090	1,000
PRE	33,747	1	33,747	166,479	0,000	0,644	166,479	1,000
Estrategia Lúdica	0,220	3	0,073	0,362	0,781	0,012	1,085	0,119
Error	18,650	92	0,203					
Total	2030,099	97						
Total corregido	52,617	96						

a. R al cuadrado = 0,646 (R al cuadrado ajustada = 0,630)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Factor Autocontrol* sobre la variable dependiente *POS-Factor Autocontrol* [ $F_{PRE-Factor Autocontrol} (1; 92) = 166,479; p<0,001; \eta^2=0,644; potencia observada=1,000$ ]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Factor Autocontrol* [ $F_{Estrategia Lúdica} (3; 92) = 0,362; p=0,781; \eta^2=0,012; potencia observada=0,119$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest del factor *Autocontrol*, una vez eliminada la influencia del pretest.



**Factor Bienestar**

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Factor Bienestar*, una vez eliminada la influencia que la variable *PRE-Factor Bienestar* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Factor Bienestar*, una vez eliminada la influencia que la variable *PRE-Factor Bienestar* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Factor Bienestar* en cada grupo seguían o no una distribución normal (tabla 105), constatándose el cumplimiento de este supuesto.

**Tabla 105**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Bienestar)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Factor Bienestar	0	0,984	27	0,946	Sí
	1	0,947	25	0,211	Sí
	2	0,929	25	0,083	Sí
	3	0,964	20	0,626	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 2,490; p=0,065$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Factor Bienestar* (variable dependiente) sobre *PRE-Factor Bienestar* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Factor Bienestar* contra *PRE-Factor Bienestar*: uno general (figura 61) y los otros cuatro separados según la variable independiente

(figura 62). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

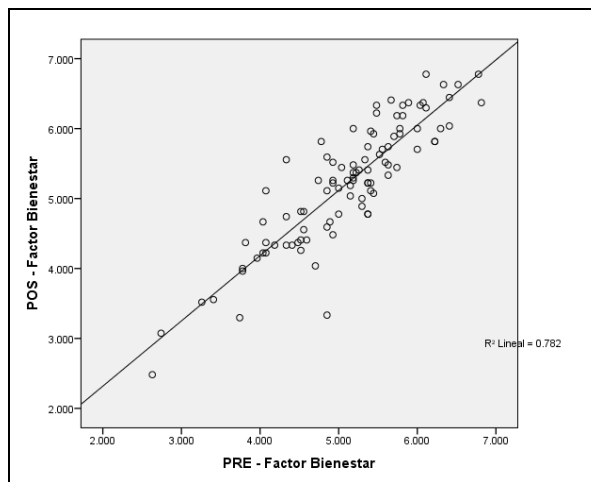


Figura 61. Diagrama de dispersión general de POS-Factor Bienestar contra PRE-Factor Bienestar. Elaboración propia

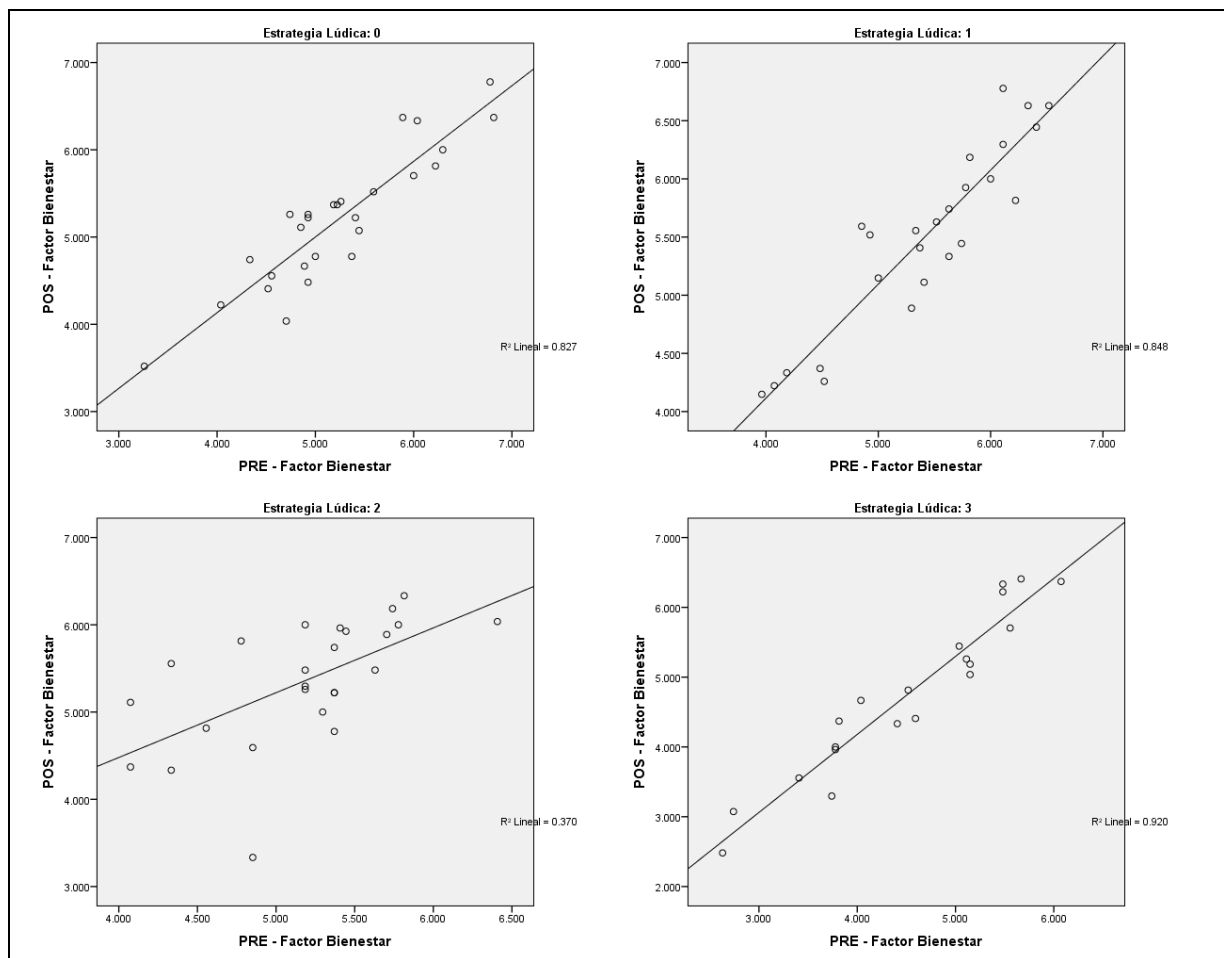


Figura 62. Diagramas de dispersión de POS-Factor Bienestar contra PRE-Factor Bienestar separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Factor Bienestar* contra *PRE-Factor Bienestar* separados según *Estrategia Lúdica* (figura 62) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Factor Bienestar* por *Estrategia Lúdica* [ $F(3; 89) = 2,023; p=0,116$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 106.

**Tabla 106**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Bienestar; Covariable: PRE-Factor Bienestar)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	59,030 <sup>a</sup>	4	14,758	87,732	0,000	0,792	350,927	1,000
Intersección	2644,185	1	2644,185	15719,274	0,000	0,994	15719,274	1,000
PRE	58,238	1	58,238	346,218	0,000	0,790	346,218	1,000
Estrategia Lúdica	0,792	3	0,264	1,570	0,202	0,049	4,709	0,401
Error	15,476	92	0,168					
Total	2718,691	97						
Total corregido	74,506	96						

a. R al cuadrado = 0,792 (R al cuadrado ajustada = 0,783)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Factor Bienestar* sobre la variable dependiente *POS-Factor Bienestar* [ $F_{PRE-Factor Bienestar} (1; 92) = 346,218; p<0,001; \eta^2=0,790; potencia observada=1,000$ ]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Factor Bienestar* [ $F_{Estrategia Lúdica} (3; 92) = 1,570; p=0,202; \eta^2=0,049; potencia observada=0,401$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest del factor *Bienestar*, una vez eliminada la influencia del pretest.

## Factor Emocionalidad

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Factor Emocionalidad*, una vez eliminada la influencia que la variable *PRE-Factor Emocionalidad* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Factor Emocionalidad*, una vez eliminada la influencia que la variable *PRE-Factor Emocionalidad* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Factor Emocionalidad* en cada grupo seguían o no una distribución normal (tabla 107), constatándose el cumplimiento de este supuesto.

**Tabla 107**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Emocionalidad)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Factor Emocionalidad	0	0,969	27	0,586	Sí
	1	0,960	25	0,424	Sí
	2	0,977	25	0,826	Sí
	3	0,978	20	0,911	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 2,520$ ;  $p=0,063$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Factor Emocionalidad* (variable dependiente) sobre *PRE-Factor Emocionalidad* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Factor Emocionalidad* contra *PRE-Factor Emocionalidad*: uno general (figura 63) y los otros cuatro separados según la variable independiente

(figura 64). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

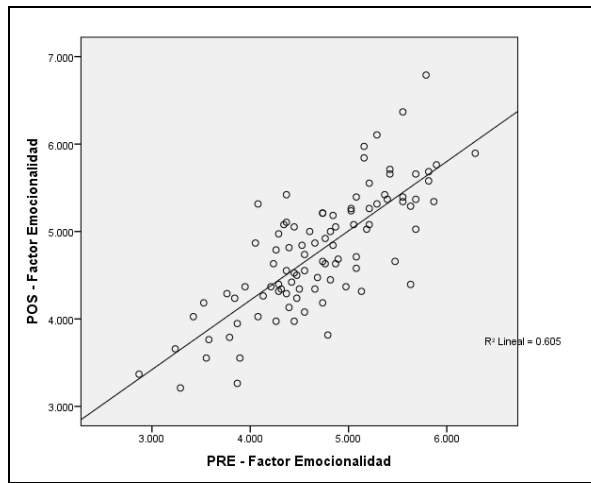


Figura 63. Diagrama de dispersión general de POS-Factor Emocionalidad contra PRE-Factor Emocionalidad. Elaboración propia

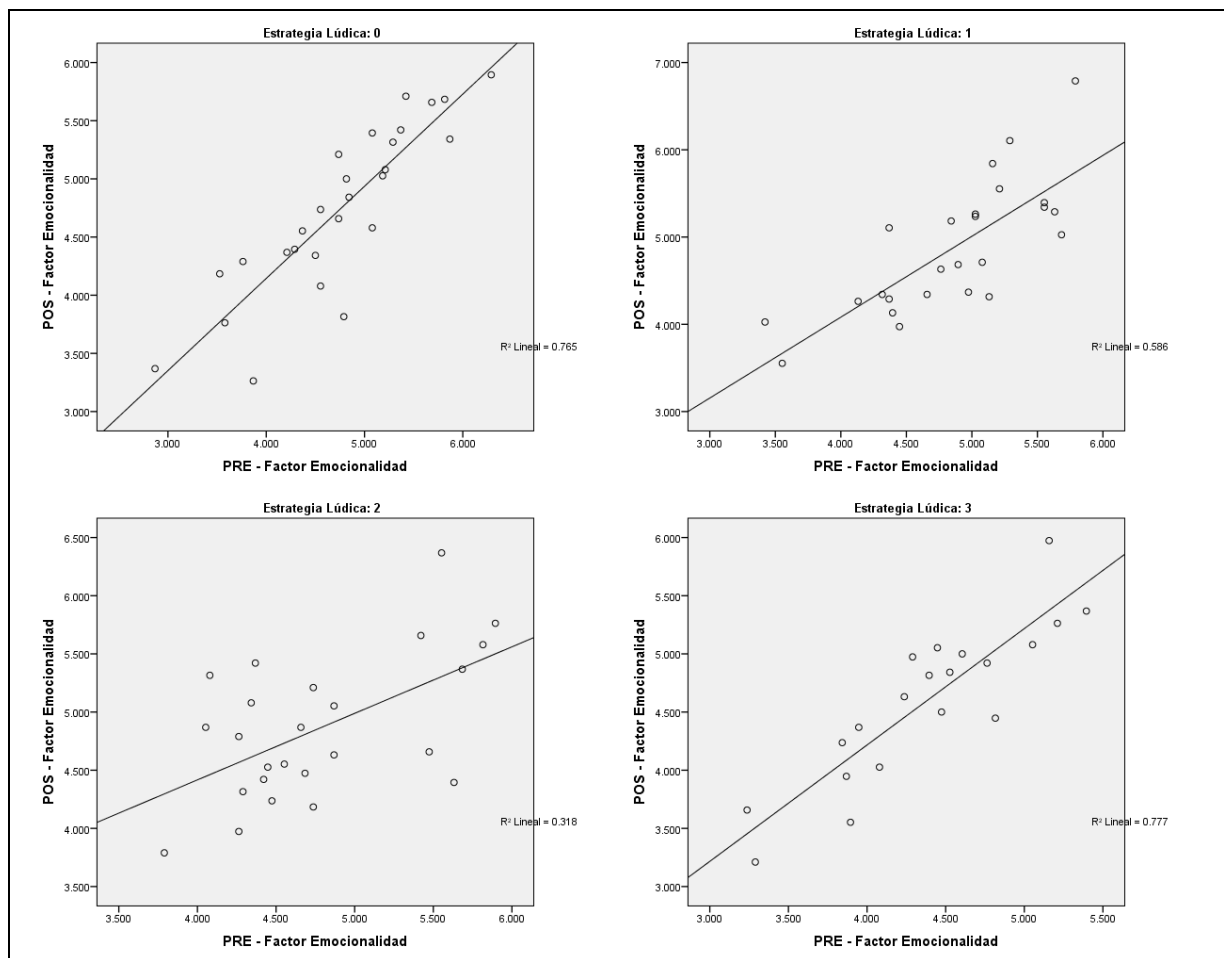


Figura 64. Diagramas de dispersión de POS-Factor Emocionalidad contra PRE-Factor Emocionalidad separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Factor Emocionalidad* contra *PRE-Factor Emocionalidad* separados según *Estrategia Lúdica* (figura 64) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Factor Emocionalidad* por *Estrategia Lúdica* [ $F(3; 89) = 1,546$ ;  $p=0,208$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 108.

**Tabla 108**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Emocionalidad; Covariable: PRE-Factor Emocionalidad)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	28,091 <sup>a</sup>	4,000	7,023	36,255	0,000	0,612	145,019	1,000
Intersección	2210,995	1,000	2210,995	11414,416	0,000	0,992	11414,416	1,000
PRE	27,783	1,000	27,783	143,432	0,000	0,609	143,432	1,000
Estrategia Lúdica	0,308	3,000	0,103	0,529	0,663	0,017	1,588	0,155
Error	17,821	92,000	0,194					
Total	2256,906	97,000						
Total corregido	45,911	96,000						

a. R al cuadrado = 0,612 (R al cuadrado ajustada = 0,595)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Factor Emocionalidad* sobre la variable dependiente *POS-Factor Emocionalidad* [ $F_{\text{PRE-Factor Emocionalidad}}(1; 92) = 143,432$ ;  $p < 0,001$ ;  $\eta^2 = 0,609$ ; potencia observada = 1,000]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Factor Emocionalidad* [ $F_{\text{Estrategia Lúdica}}(3; 92) = 0,529$ ;  $p = 0,663$ ;  $\eta^2 = 0,017$ ; potencia observada = 0,155].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest del factor *Emocionalidad*, una vez eliminada la influencia del pretest.

## Factor Sociabilidad

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Factor Sociabilidad*, una vez eliminada la influencia que la variable *PRE-Factor Sociabilidad* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Factor Sociabilidad*, una vez eliminada la influencia que la variable *PRE-Factor Sociabilidad* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Factor Sociabilidad* en cada grupo seguían o no una distribución normal (tabla 109), constatándose el cumplimiento de este supuesto.

**Tabla 109**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Sociabilidad)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Factor Sociabilidad	0	0,961	27	0,390	Sí
	1	0,960	25	0,424	Sí
	2	0,970	25	0,656	Sí
	3	0,957	20	0,478	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 1,176; p=0,323$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Factor Sociabilidad* (variable dependiente) sobre *PRE-Factor Sociabilidad* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Factor Sociabilidad* contra *PRE-Factor Sociabilidad*: uno general (figura 65) y los otros cuatro separados según la variable independiente

(figura 66). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

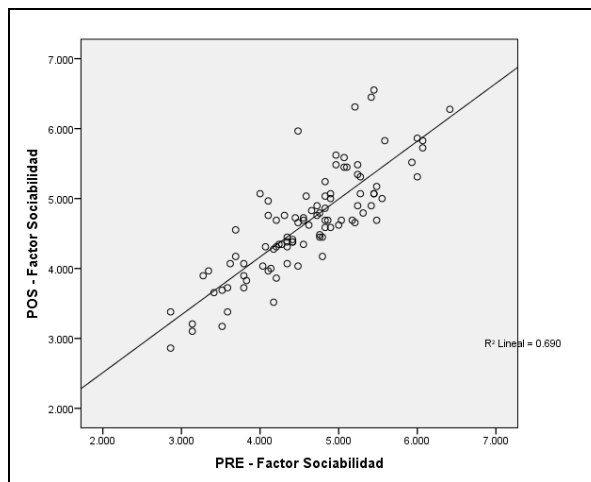


Figura 65. Diagrama de dispersión general de POS-Factor Sociabilidad contra PRE-Factor Sociabilidad. Elaboración propia

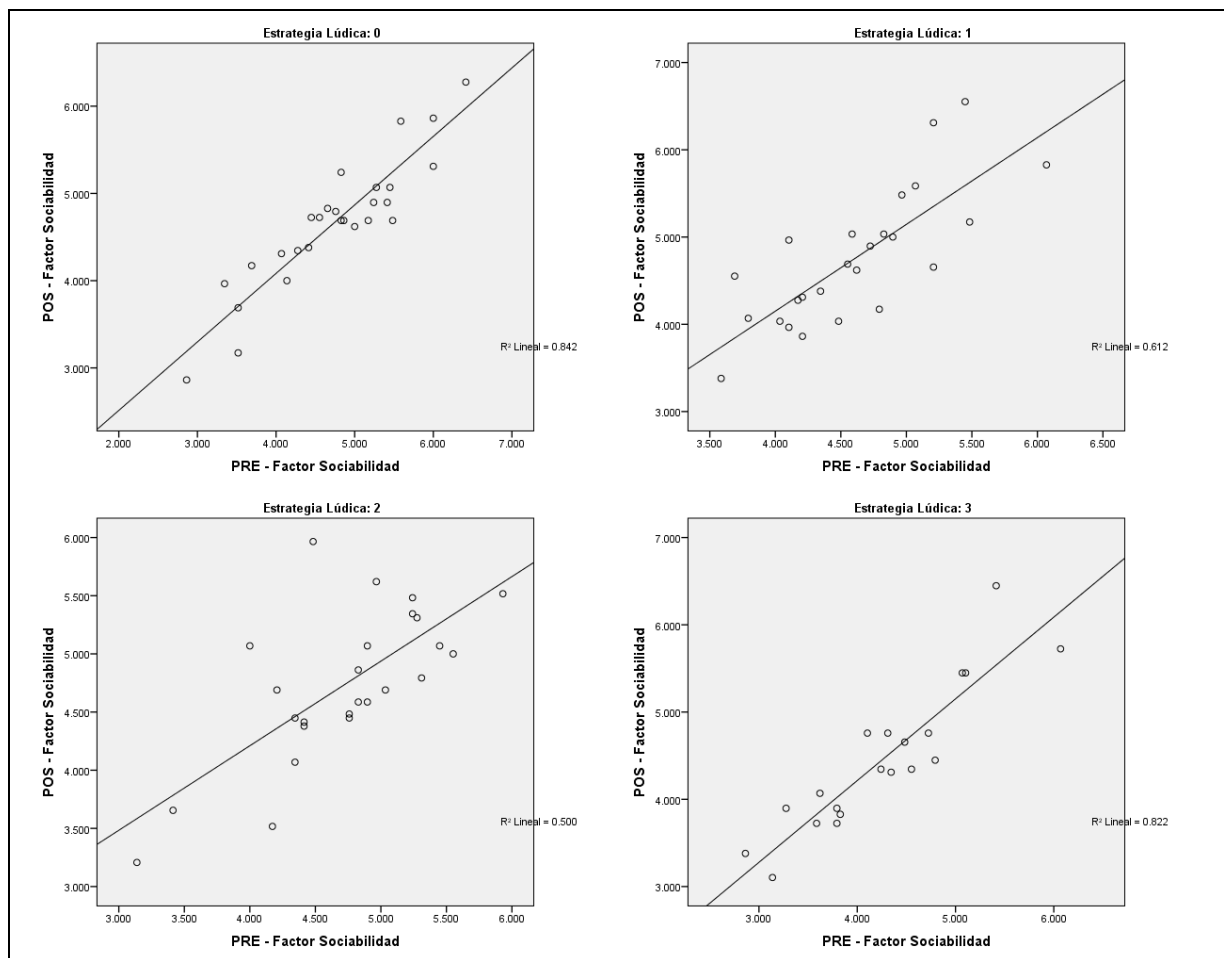


Figura 66. Diagramas de dispersión de POS-Factor Sociabilidad contra PRE-Factor Sociabilidad separados según Estrategia Lúdica. Elaboración propia



Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión. Por un lado, los diagramas de dispersión de *POS-Factor Sociabilidad* contra *PRE-Factor Sociabilidad* separados según *Estrategia Lúdica* (figura 66) mostraron líneas de regresión similares. Por otra parte, se observó una interacción estadísticamente no significativa de *PRE-Factor Sociabilidad* por *Estrategia Lúdica* [ $F(3; 89) = 0,981; p=0,405$ ], lo cual permitió constatar que se cumplía este supuesto.

Dado que todos los supuestos para la aplicación de ANCOVA de un factor se cumplieron, esta prueba se pudo llevar a cabo. Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 110.

**Tabla 110**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: POS-Factor Sociabilidad; Covariable: PRE-Factor Sociabilidad)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	37,722 <sup>a</sup>	4	9,430	54,381	0,000	0,703	217,526	1,000
Intersección	2106,227	1	2106,227	12145,748	0,000	0,992	12145,748	1,000
PRE	37,033	1	37,033	213,556	0,000	0,699	213,556	1,000
Estrategia Lúdica	0,688	3	0,229	1,323	0,272	0,041	3,970	0,342
Error	15,954	92	0,173					
Total	2159,902	97						
Total corregido	53,676	96						

a. R al cuadrado = 0,703 (R al cuadrado ajustada = 0,690)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto estadísticamente significativo de la covariable *PRE-Factor Sociabilidad* sobre la variable dependiente *POS-Factor Sociabilidad* [ $F_{PRE-Factor Sociabilidad} (1; 92) = 213,556; p<0,001; \eta^2=0,699; potencia observada=1,000$ ]. Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *POS-Factor Sociabilidad* [ $F_{Estrategia Lúdica} (3; 92) = 1,323; p=0,272; \eta^2=0,041; potencia observada=0,342$ ].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados en función de la *estrategia lúdica* empleada, no existen diferencias significativas en el posttest del factor *Sociabilidad*, una vez eliminada la influencia del pretest.

### Factor Global de Inteligencia Emocional

Primero se plantearon las hipótesis nula y alternativa que serían contrastadas mediante la prueba ANCOVA de un factor:

- $H_0$ : Entre los grupos formados en función de la variable *Estrategia lúdica* no existen diferencias significativas en la media de la variable *POS-Factor Global de IE*, una vez eliminada la influencia que la variable *PRE-Factor Global de IE* pudiera ejercer sobre esta.
- $H_1$ : Entre los grupos formados en función de la variable *Estrategia lúdica* existen diferencias significativas en la media de la variable *POS-Factor Global de IE*, una vez eliminada la influencia que la variable *PRE-Factor Global de IE* pudiera ejercer sobre esta.

Después se analizó el cumplimiento de los supuestos necesarios para aplicar esta prueba. Para empezar, se usó la prueba de Shapiro-Wilk con el objetivo de determinar si los residuos para la variable *POS-Factor Global de IE* en cada grupo seguían o no una distribución normal (tabla 111), constatándose el cumplimiento de este supuesto.

**Tabla 111**

*Prueba de Shapiro-Wilk para verificar la normalidad de los residuos (variable POS-Factor Global de IE)*

	Estrategia Lúdica	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
Residuo para POS-Factor Global IE	0	0,940	27	0,125	Sí
	1	0,958	25	0,380	Sí
	2	0,975	25	0,780	Sí
	3	0,979	20	0,927	Sí

*Nota:* Elaboración propia

A continuación se llevó a cabo la prueba de Levene para determinar si se cumplía el supuesto de homocedasticidad. El resultado de esta prueba [ $F(3; 93) = 2,197; p=0,094$ ] permitió constatar que sí se cumplía este supuesto.

Para determinar si la regresión de *POS-Factor Global de IE* (variable dependiente) sobre *PRE-Factor Global de IE* (covariable) era lineal, se realizaron cinco diagramas de dispersión (*scatterplots*) de *POS-Factor Global de IE* contra *PRE-Factor Global de IE*: uno general (figura 67) y los otros cuatro separados según la variable independiente

(figura 68). Estos diagramas sugirieron la existencia de una relación lineal positiva entre ambas variables, constatándose así el cumplimiento de este supuesto.

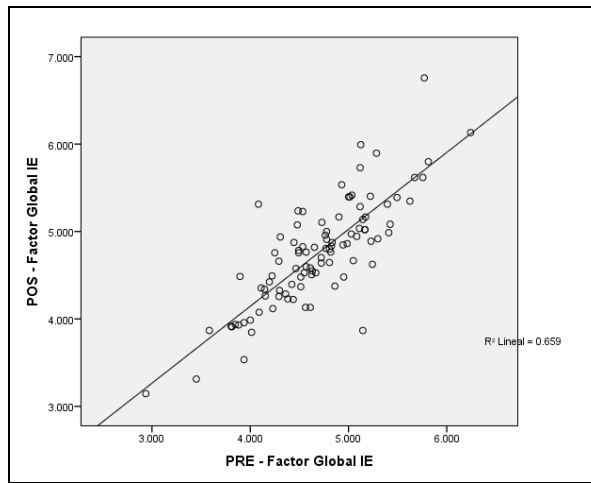


Figura 67. Diagrama de dispersión general de POS-Factor Global de IE contra PRE-Factor Global de IE. Elaboración propia

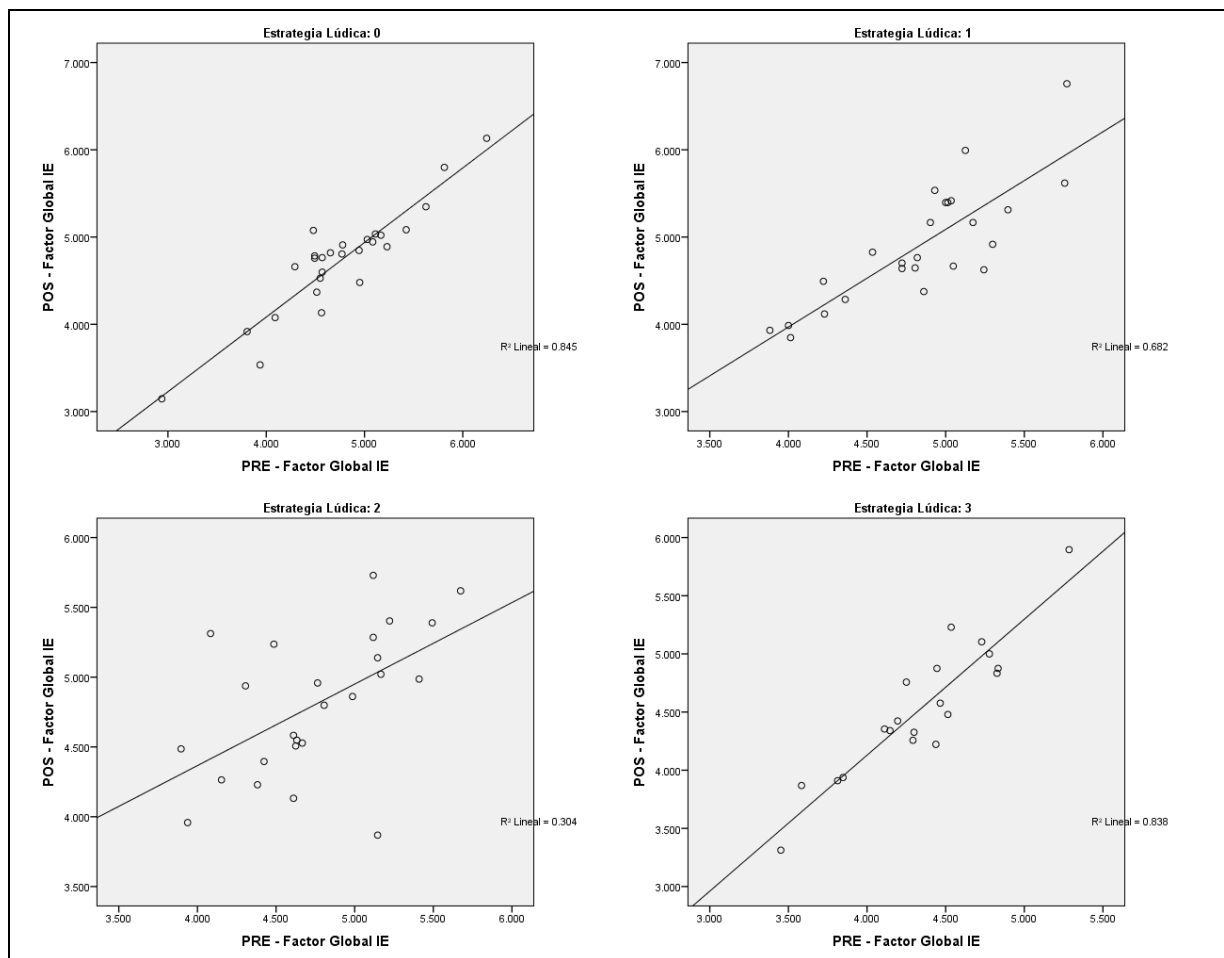


Figura 68. Diagramas de dispersión de POS-Factor Global de IE contra PRE-Factor Global de IE separados según Estrategia Lúdica. Elaboración propia

Por último, se analizó si se cumplía el supuesto de homogeneidad de las pendientes de regresión.

Por un lado, los diagramas de dispersión de *POS-Factor Global de IE* contra *PRE-Factor Global de IE* separados según *Estrategia Lúdica* (figura 68) mostraron líneas de regresión similares.

Sin embargo, por otra parte se observó una interacción estadísticamente significativa de *PRE-Factor Global de IE* por *Estrategia Lúdica* [ $F(3; 89) = 3,267$ ;  $p=0,025$ ;  $\eta^2=0,099$ ; potencia observada= $0,731$ ], lo cual evidenció que no se cumplía este supuesto.

La violación del supuesto de homogeneidad de las pendientes de regresión obligó a llevar a cabo la versión no paramétrica de la prueba ANCOVA de un factor. Para convertir las variables *PRE-Factor Global de IE* y *POS-Factor Global de IE* en rangos se crearon dos nuevas variables (*Rango de PRE-Factor Global de IE* y *Rango de POS-Factor Global de IE*), cuyos estadísticos descriptivos se muestran en la tabla 112.

**Tabla 112**

*Estadísticos de Rango de PRE-Factor Global de IE y Rango de POS-Factor Global de IE*

Variable	Estrategia Lúdica	Estadístico		
		Media	Desviación estándar	N
Rango de PRE-Factor Global de IE	0	51,83333	28,562011	27
	1	57,12000	27,480781	25
	2	52,54000	27,434741	25
	3	30,60000	22,586442	20
Rango de POS-Factor Global de IE	0	49,11111	26,306015	27
	1	54,38000	30,292422	25
	2	52,44000	27,858766	25
	3	37,82500	27,030818	20

*Nota:* Elaboración propia

Los estadísticos calculados por el software IBM SPSS Statistics 23 se muestran a continuación en la tabla 113.

**Tabla 113**

*Resultados de ANCOVA (VI: Estrategia Lúdica; VD: Rango de POS-Factor Global de IE; Covariable: Rango de PRE-Factor Global de IE)*

Origen	Tipo I de suma de cuadrados	gl	Media cuadrática	F	Sig.	Eta parcial al cuadrado	Parámetro de no centralidad	Potencia observada <sup>b</sup>
Modelo corregido	45485,363 <sup>a</sup>	4	11371,341	34,237	0,000	0,598	136,947	1,000
Intersección	232897,000	1	232897,000	701,207	0,000	0,884	701,207	1,000
Rango de PRE	45129,361	1	45129,361	135,876	0,000	0,596	135,876	1,000
Estrategia Lúdica	356,002	3	118,667	0,357	0,784	0,012	1,072	0,118
Error	30556,637	92	332,137					
Total	308939,000	97						
Total corregido	76042,000	96						

a. R al cuadrado = 0,598 (R al cuadrado ajustada = 0,581)

b. Se ha calculado utilizando alpha = 0,05

*Nota:* Elaboración propia

Los resultados de la prueba ANCOVA sugirieron la existencia de un efecto significativo de la covariable *Rango de PRE-Factor Global de IE* sobre la variable dependiente *Rango de POS-Factor Global de IE* ( $F_{\text{Rango de PRE-Factor Global de IE}}(1; 92) = 135,876$ ;  $p < 0,001$ ;  $\eta^2 = 0,596$ ; potencia observada = 1,000). Sin embargo, no hubo evidencia de un efecto estadísticamente significativo de la variable independiente *Estrategia Lúdica* sobre la variable dependiente *Rango de POS-Factor Global de IE* [ $F_{\text{Estrategia Lúdica}}(3; 92) = 0,357$ ;  $p = 0,784$ ;  $\eta^2 = 0,012$ ; potencia observada = 0,118].

Por lo tanto, se mantuvo la hipótesis nula que postula que, entre los grupos formados según la *estrategia lúdica* empleada, no existen diferencias estadísticamente significativas en el posttest del factor global de *Inteligencia Emocional*, una vez eliminada la influencia del pretest.

### 9.2.5. Quinta hipótesis (H5)

La quinta hipótesis postula que, dentro de cada grupo formado en función de la *estrategia lúdica* empleada, las variables referidas a la *inteligencia emocional* de los estudiantes presentan diferencias significativas entre los valores medidos en el pretest y en el postest.

Al igual que en el caso de la tercera hipótesis, aquí también cada par formado por una de estas variables medida en el pretest y la misma variable medida en el postest —en ambos casos se trata de valores cuantitativos continuos— corresponde a una posible hipótesis que puede contrastarse con la prueba *t* para muestras relacionadas (paramétrica) o la prueba de rangos con signo de Wilcoxon para muestras relacionadas (no paramétrica).

En consecuencia, como punto de partida se usó la prueba de Shapiro-Wilk con el objetivo de determinar si las *diferencias* entre los valores medidos en el pretest y en el postest para las 20 variables referidas a la *inteligencia emocional* de los estudiantes seguían o no una distribución normal (tabla 114).

**Tabla 114**

*Prueba de Shapiro-Wilk para las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)*

Estrategia Lúdica	Variable	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
0	DIF-Faceta Adaptabilidad	0,894	27	0,010	No
	DIF-Faceta Asertividad	0,968	27	0,553	Sí
	DIF-Faceta Autoestima	0,983	27	0,927	Sí
	DIF-Faceta Automotivación	0,963	27	0,426	Sí
	DIF-Faceta Conciencia Social	0,977	27	0,794	Sí
	DIF-Faceta Control de la Impulsividad	0,961	27	0,393	Sí
	DIF-Faceta Empatía	0,969	27	0,578	Sí
	DIF-Faceta Expresividad Emocional	0,977	27	0,787	Sí
	DIF-Faceta Felicidad	0,981	27	0,884	Sí
	DIF-Faceta Gestión de la Emoción	0,818	27	0,000	No
	DIF-Faceta Gestión del Estrés	0,956	27	0,292	Sí
	DIF-Faceta Optimismo	0,856	27	0,002	No

*Nota:* Elaboración propia

Tabla 114 (cont.)

*Prueba de Shapiro-Wilk para las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)*

Estrategia Lúdica	Variable	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
0 (cont.)	DIF-Faceta Percepción Emocional	0,962	27	0,405	Sí
	DIF-Faceta Regulación Emocional	0,968	27	0,562	Sí
	DIF-Faceta Relaciones	0,971	27	0,627	Sí
	DIF-Factor Autocontrol	0,979	27	0,830	Sí
	DIF-Factor Bienestar	0,959	27	0,350	Sí
	DIF-Factor Emocionalidad	0,969	27	0,569	Sí
	DIF-Factor Sociabilidad	0,987	27	0,977	Sí
	DIF-Factor Global de Inteligencia Emocional	0,979	27	0,850	Sí
1	DIF-Faceta Adaptabilidad	0,966	25	0,540	Sí
	DIF-Faceta Asertividad	0,956	25	0,337	Sí
	DIF-Faceta Autoestima	0,950	25	0,252	Sí
	DIF-Faceta Automotivación	0,955	25	0,324	Sí
	DIF-Faceta Conciencia Social	0,930	25	0,085	Sí
	DIF-Faceta Control de la Impulsividad	0,966	25	0,550	Sí
	DIF-Faceta Empatía	0,945	25	0,189	Sí
	DIF-Faceta Expresividad Emocional	0,940	25	0,147	Sí
	DIF-Faceta Felicidad	0,967	25	0,574	Sí
	DIF-Faceta Gestión de la Emoción	0,950	25	0,256	Sí
	DIF-Faceta Gestión del Estrés	0,908	25	0,027	No
	DIF-Faceta Optimismo	0,792	25	0,000	No
	DIF-Faceta Percepción Emocional	0,972	25	0,701	Sí
	DIF-Faceta Regulación Emocional	0,877	25	0,006	No
	DIF-Faceta Relaciones	0,975	25	0,775	Sí
	DIF-Factor Autocontrol	0,927	25	0,076	Sí
	DIF-Factor Bienestar	0,939	25	0,137	Sí
	DIF-Factor Emocionalidad	0,969	25	0,628	Sí
	DIF-Factor Sociabilidad	0,952	25	0,280	Sí
	DIF-Factor Global de Inteligencia Emocional	0,950	25	0,246	Sí
2	DIF-Faceta Adaptabilidad	0,934	25	0,108	Sí
	DIF-Faceta Asertividad	0,909	25	0,028	No
	DIF-Faceta Autoestima	0,973	25	0,715	Sí
	DIF-Faceta Automotivación	0,910	25	0,031	No
	DIF-Faceta Conciencia Social	0,938	25	0,132	Sí
	DIF-Faceta Control de la Impulsividad	0,968	25	0,605	Sí

Nota: Elaboración propia

Tabla 114 (cont.)

*Prueba de Shapiro-Wilk para las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)*

Estrategia Lúdica	Variable	Estadístico	gl	Sig.	Normalidad ( $p \geq 0,05$ )
2 (cont.)	DIF-Faceta Empatía	0,955	25	0,316	Sí
	DIF-Faceta Expresividad Emocional	0,971	25	0,677	Sí
	DIF-Faceta Felicidad	0,862	25	0,003	No
	DIF-Faceta Gestión de la Emoción	0,956	25	0,333	Sí
	DIF-Faceta Gestión del Estrés	0,923	25	0,060	Sí
	DIF-Faceta Optimismo	0,972	25	0,701	Sí
	DIF-Faceta Percepción Emocional	0,980	25	0,885	Sí
	DIF-Faceta Regulación Emocional	0,958	25	0,382	Sí
	DIF-Faceta Relaciones	0,894	25	0,014	No
	DIF-Factor Autocontrol	0,979	25	0,857	Sí
	DIF-Factor Bienestar	0,951	25	0,266	Sí
	DIF-Factor Emocionalidad	0,972	25	0,700	Sí
	DIF-Factor Sociabilidad	0,890	25	0,011	No
	DIF-Factor Global de Inteligencia Emocional	0,900	25	0,019	No
3	DIF-Faceta Adaptabilidad	0,919	20	0,095	Sí
	DIF-Faceta Asertividad	0,970	20	0,747	Sí
	DIF-Faceta Autoestima	0,940	20	0,240	Sí
	DIF-Faceta Automotivación	0,976	20	0,877	Sí
	DIF-Faceta Conciencia Social	0,954	20	0,435	Sí
	DIF-Faceta Control de la Impulsividad	0,960	20	0,541	Sí
	DIF-Faceta Empatía	0,949	20	0,347	Sí
	DIF-Faceta Expresividad Emocional	0,965	20	0,650	Sí
	DIF-Faceta Felicidad	0,926	20	0,131	Sí
	DIF-Faceta Gestión de la Emoción	0,938	20	0,215	Sí
	DIF-Faceta Gestión del Estrés	0,925	20	0,123	Sí
	DIF-Faceta Optimismo	0,964	20	0,621	Sí
	DIF-Faceta Percepción Emocional	0,976	20	0,870	Sí
	DIF-Faceta Regulación Emocional	0,906	20	0,054	Sí
	DIF-Faceta Relaciones	0,974	20	0,833	Sí
	DIF-Factor Autocontrol	0,978	20	0,903	Sí
	DIF-Factor Bienestar	0,975	20	0,851	Sí
	DIF-Factor Emocionalidad	0,960	20	0,551	Sí
	DIF-Factor Sociabilidad	0,965	20	0,640	Sí
DIF-Factor Global de Inteligencia Emocional	0,968	20	0,711	Sí	

Nota: Elaboración propia



A continuación se llevaron a cabo las pruebas que se reseñan en la tabla 115, en función de que se haya cumplido o no, en el análisis anterior, el supuesto de normalidad.

**Tabla 115**

*Contraste de las hipótesis referidas a las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)*

Estrategia	H <sub>0</sub> se refiere a*	Prueba	Resultado	Efecto	Decisión
0	Faceta Adaptabilidad	Wilcoxon	$z=-1,220$ ; $p=0,222$ ; $r=-0,166$	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Asertividad	t	$t(26) = -0,319$ ; $p=0,752$ ; $d=-0,061$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Autoestima	t	$t(26) = 0,183$ ; $p=0,856$ ; $d=0,035$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Automotivación	t	$t(26) = 0,228$ ; $p=0,822$ ; $d=0,044$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Conciencia Social	t	$t(26) = -1,509$ ; $p=0,143$ ; $d=-0,290$	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Control de la Impulsividad	t	$t(26) = 0,416$ ; $p=0,681$ ; $d=0,080$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Empatía	t	$t(26) = -0,192$ ; $p=0,850$ ; $d=-0,037$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Expresividad Emocional	t	$t(26) = 0,037$ ; $p=0,971$ ; $d=0,007$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Felicidad	t	$t(26) = -0,557$ ; $p=0,582$ ; $d=-0,107$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Gestión de la Emoción	Wilcoxon	$z=-0,024$ ; $p=0,981$ ; $r=-0,003$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Gestión del Estrés	t	$t(26) = -1,220$ ; $p=0,233$ ; $d=-0,235$	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Optimismo	Wilcoxon	$z=-0,031$ ; $p=0,976$ ; $r=-0,004$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Percepción Emocional	t	$t(26) = 0,275$ ; $p=0,785$ ; $d=0,053$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Regulación Emocional	t	$t(26) = -1,041$ ; $p=0,307$ ; $d=-0,200$	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Relaciones	t	$t(26) = -0,605$ ; $p=0,551$ ; $d=-0,116$	Pequeño	Se conservó H <sub>0</sub>
	Factor Autocontrol	t	$t(26) = -0,957$ ; $p=0,348$ ; $d=-0,184$	Pequeño	Se conservó H <sub>0</sub>
	Factor Bienestar	t	$t(26) = -0,462$ ; $p=0,648$ ; $d=-0,089$	Pequeño	Se conservó H <sub>0</sub>
	Factor Emocionalidad	t	$t(26) = -0,158$ ; $p=0,876$ ; $d=-0,030$	Pequeño	Se conservó H <sub>0</sub>
Factor Sociabilidad	t	$t(26) = -1,095$ ; $p=0,283$ ; $d=-0,211$	Pequeño a mediano	Se conservó H <sub>0</sub>	
Factor Global de IE	t	$t(26) = -0,511$ ; $p=0,614$ ; $d=-0,098$	Pequeño	Se conservó H <sub>0</sub>	
1	Faceta Adaptabilidad	t	$t(24) = 0,532$ ; $p=0,600$ ; $d=0,106$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Asertividad	t	$t(24) = -0,062$ ; $p=0,951$ ; $d=-0,012$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Autoestima	t	$t(24) = 0,554$ ; $p=0,585$ ; $d=0,111$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Automotivación	t	$t(24) = 0,285$ ; $p=0,778$ ; $d=0,057$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Conciencia Social	t	$t(24) = 2,236$ ; $p=0,035$ ; $d=0,447$	Pequeño a mediano	Se rechazó H <sub>0</sub>
	Faceta Control de la Impulsividad	t	$t(24) = 0,227$ ; $p=0,822$ ; $d=0,045$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Empatía	t	$t(24) = 0,554$ ; $p=0,584$ ; $d=0,111$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Expresividad Emocional	t	$t(24) = -0,074$ ; $p=0,942$ ; $d=-0,015$	Pequeño	Se conservó H <sub>0</sub>
	Faceta Felicidad	t	$t(24) = 0,360$ ; $p=0,722$ ; $d=0,072$	Pequeño	Se conservó H <sub>0</sub>

\*. H<sub>0</sub> postula que la media (prueba t) o la mediana (Wilcoxon) de las diferencias entre los valores del pretest y del postest es igual a 0.

Nota: Elaboración propia

Tabla 115 (cont.)

Contraste de las hipótesis referidas a las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)

Estrategia	H <sub>0</sub> se refiere a*	Prueba	Resultado	Efecto	Decisión
1 (cont.)	Faceta Gestión de la Emoción	t	t(24) = 1,340; p=0,193; d=0,268	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Gestión del Estrés	Wilcoxon	z=-0,826; p=0,409; r=-0,117	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Optimismo	Wilcoxon	z=-1,620; p=0,105; r=-0,229	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Percepción Emocional	t	t(24) = 0,357; p=0,724; d=0,071	Pequeño	Se conservó H <sub>0</sub>
	Faceta Regulación Emocional	Wilcoxon	z=-0,930; p=0,352; r=-0,132	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Relaciones	t	t(24) = -0,548; p=0,589; d=-0,110	Pequeño	Se conservó H <sub>0</sub>
	Factor Autocontrol	t	t(24) = 0,360; p=0,722; d=0,072	Pequeño	Se conservó H <sub>0</sub>
	Factor Bienestar	t	t(24) = 1,412; p=0,171; d=0,282	Pequeño a mediano	Se conservó H <sub>0</sub>
	Factor Emocionalidad	t	t(24) = 0,206; p=0,838; d=0,041	Pequeño	Se conservó H <sub>0</sub>
	Factor Sociabilidad	t	t(24) = 1,536; p=0,138; d=0,307	Pequeño a mediano	Se conservó H <sub>0</sub>
	Factor Global de IE	t	t(24) = 0,874; p=0,391; d=0,175	Pequeño	Se conservó H <sub>0</sub>
2	Faceta Adaptabilidad	t	t(24) = -0,218; p=0,829; d=-0,044	Pequeño	Se conservó H <sub>0</sub>
	Faceta Asertividad	Wilcoxon	z=-0,910; p=0,363; r=-0,129	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Autoestima	t	t(24) = 1,502; p=0,146; d=0,300	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Automotivación	Wilcoxon	z=-0,022; p=0,983; r=-0,003	Pequeño	Se conservó H <sub>0</sub>
	Faceta Conciencia Social	t	t(24) = -0,484; p=0,633; d=-0,097	Pequeño	Se conservó H <sub>0</sub>
	Faceta Control de la Impulsividad	t	t(24) = -2,287; p=0,031; d=-0,457	Pequeño a mediano	Se rechazó H <sub>0</sub>
	Faceta Empatía	t	t(24) = 0,767; p=0,451; d=0,153	Pequeño	Se conservó H <sub>0</sub>
	Faceta Expresividad Emocional	t	t(24) = 0,526; p=0,603; d=0,105	Pequeño	Se conservó H <sub>0</sub>
	Faceta Felicidad	Wilcoxon	z=-0,887; p=0,375; r=-0,125	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Gestión de la Emoción	t	t(24) = 1,129; p=0,270; d=0,226	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Gestión del Estrés	t	t(24) = 0,024; p=0,981; d=0,005	Pequeño	Se conservó H <sub>0</sub>
	Faceta Optimismo	t	t(24) = 2,021; p=0,055; d=0,404	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Percepción Emocional	t	t(24) = 1,133; p=0,268; d=0,227	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Regulación Emocional	t	t(24) = 1,380; p=0,180; d=0,276	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Relaciones	Wilcoxon	z=-0,845; p=0,398; r=-0,120	Pequeño a mediano	Se conservó H <sub>0</sub>
	Factor Autocontrol	t	t(24) = -0,037; p=0,971; d=-0,007	Pequeño	Se conservó H <sub>0</sub>
	Factor Bienestar	t	t(24) = 1,537; p=0,137; d=0,307	Pequeño a mediano	Se conservó H <sub>0</sub>
	Factor Emocionalidad	t	t(24) = 0,744; p=0,464; d=0,149	Pequeño	Se conservó H <sub>0</sub>
	Factor Sociabilidad	Wilcoxon	z=-0,557; p=0,577; r=-0,079	Pequeño	Se conservó H <sub>0</sub>
Factor Global de IE	Wilcoxon	z=-0,202; p=0,840; r=-0,029	Pequeño	Se conservó H <sub>0</sub>	
3	Faceta Adaptabilidad	t	t(19) = 0,590; p=0,562; d=0,132	Pequeño	Se conservó H <sub>0</sub>
	Faceta Asertividad	t	t(19) = 2,514; p=0,021; d=0,562	Mediano a grande	Se rechazó H <sub>0</sub>

\*. H<sub>0</sub> postula que la media (prueba t) o la mediana (Wilcoxon) de las diferencias entre los valores del pretest y del postest es igual a 0.

Nota: Elaboración propia

Tabla 115 (cont.)

*Contraste de las hipótesis referidas a las diferencias entre el pretest y el postest (inteligencia emocional de los estudiantes)*

Estrategia	H <sub>0</sub> se refiere a*	Prueba	Resultado	Efecto	Decisión
3 (cont.)	Faceta Autoestima	t	t(19) = 2,969; p=0,008; d=0,664	Mediano a grande	Se rechazó H <sub>0</sub>
	Faceta Automotivación	t	t(19) = 2,288; p=0,034; d=0,512	Mediano a grande	Se rechazó H <sub>0</sub>
	Faceta Conciencia Social	t	t(19) = 2,422; p=0,026; d=0,541	Mediano a grande	Se rechazó H <sub>0</sub>
	Faceta Control de la Impulsividad	t	t(19) = 1,716; p=0,102; d=0,384	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Empatía	t	t(19) = 1,251; p=0,226; d=0,280	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Expresividad Emocional	t	t(19) = 3,394; p=0,003; d=0,759	Mediano a grande	Se rechazó H <sub>0</sub>
	Faceta Felicidad	t	t(19) = 1,291; p=0,212; d=0,289	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Gestión de la Emoción	t	t(19) = 0,687; p=0,501; d=0,154	Pequeño	Se conservó H <sub>0</sub>
	Faceta Gestión del Estrés	t	t(19) = 3,863; p=0,001; d=0,864	Grande	Se rechazó H <sub>0</sub>
	Faceta Optimismo	t	t(19) = 2,041; p=0,055; d=0,456	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Percepción Emocional	t	t(19) = 1,514; p=0,147; d=0,339	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Regulación Emocional	t	t(19) = -1,332; p=0,199; d=-0,298	Pequeño a mediano	Se conservó H <sub>0</sub>
	Faceta Relaciones	t	t(19) = 0,519; p=0,610; d=0,116	Pequeño	Se conservó H <sub>0</sub>
	Factor Autocontrol	t	t(19) = 1,493; p=0,152; d=0,334	Pequeño a mediano	Se conservó H <sub>0</sub>
	Factor Bienestar	t	t(19) = 3,096; p=0,006; d=0,692	Mediano a grande	Se rechazó H <sub>0</sub>
	Factor Emocionalidad	t	t(19) = 3,028; p=0,007; d=0,677	Mediano a grande	Se rechazó H <sub>0</sub>
	Factor Sociabilidad	t	t(19) = 2,500; p=0,022; d=0,559	Mediano a grande	Se rechazó H <sub>0</sub>
	Factor Global de IE	t	t(19) = 3,449; p=0,003; d=0,771	Mediano a grande	Se rechazó H <sub>0</sub>

\*. H<sub>0</sub> postula que la media (prueba t) o la mediana (Wilcoxon) de las diferencias entre los valores del pretest y del postest es igual a 0.

Nota: Elaboración propia

Al igual que en la tercera hipótesis, aquí también el tamaño del efecto se mostró, en función de la prueba usada, a través de los índices  $r$  (en las pruebas de rangos con signo de Wilcoxon) y  $d$  de Cohen (en las pruebas  $t$  para muestras relacionadas). Para interpretar el significado de los tamaños del efecto, se consultó en cada caso la tabla 66.

Por último se comentan las diferencias significativas encontradas entre los valores medidos en el pretest y en el postest. A pesar de haber utilizado, según el caso, la prueba  $t$  para muestras relacionadas (paramétrica) y la prueba de rangos con signo de Wilcoxon para muestras relacionadas (no paramétrica), solo la primera reveló diferencias estadísticamente significativas, por lo que en todos los comentarios se mencionan exclusivamente diferencias entre medias (M). Las medias que aparecen en los comentarios pueden consultarse en la tabla 39.

En el grupo de control, ninguna prueba reveló diferencias estadísticamente significativas entre los valores medidos en el pretest y en el postest.

En el grupo que empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*), la prueba *t* para muestras relacionadas reveló un aumento significativo [ $t(24) = 2,236$ ;  $p=0,035$ ] en la faceta *Conciencia Social*, desde el pretest ( $M=4,60000$ ) hasta el postest ( $M=4,85091$ ), con un efecto de tamaño pequeño a mediano ( $d=0,447$ ).

En el grupo en el cual se empleó la estrategia lúdica 2 (*Aprendizaje mediante la utilización de videojuegos*), la prueba *t* para muestras relacionadas reveló una reducción estadísticamente significativa [ $t(24) = -2,287$ ;  $p=0,031$ ] en la faceta *Control de la Impulsividad*, desde el pretest ( $M=4,49778$ ) hasta el postest ( $M=4,25778$ ), con un efecto de tamaño pequeño a mediano ( $d=-0,457$ ).

En el grupo en el cual se empleó la estrategia lúdica 3 (*Ludificación del aprendizaje/ Gamification*), la prueba *t* para muestras relacionadas reveló las siguientes diferencias estadísticamente significativas entre los valores medidos en el pretest y en el postest:

- un aumento estadísticamente significativo [ $t(19) = 2,514$ ;  $p=0,021$ ] en la faceta *Asertividad*, desde el pretest ( $M=4,19444$ ) hasta el postest ( $M=4,46667$ ), con un efecto de tamaño mediano a grande ( $d=0,562$ );
- un aumento estadísticamente significativo [ $t(19) = 2,969$ ;  $p=0,008$ ] en la faceta *Autoestima*, desde el pretest ( $M=4,51818$ ) hasta el postest ( $M=4,76364$ ), con un efecto de tamaño mediano a grande ( $d=0,664$ );
- un aumento estadísticamente significativo [ $t(19) = 2,288$ ;  $p=0,034$ ] en la faceta *Automotivación*, desde el pretest ( $M=4,33500$ ) hasta el postest ( $M=4,55500$ ), con un efecto de tamaño mediano a grande ( $d=0,512$ );
- un aumento estadísticamente significativo [ $t(19) = 2,422$ ;  $p=0,026$ ] en la faceta *Conciencia Social*, desde el pretest ( $M=4,16818$ ) hasta el postest ( $M=4,39091$ ), con un efecto de tamaño mediano a grande ( $d=0,541$ );
- un aumento estadísticamente significativo [ $t(19) = 3,394$ ;  $p=0,003$ ] en la faceta *Expresividad Emocional*, desde el pretest ( $M=3,40500$ ) hasta el postest ( $M=3,84500$ ), con un efecto de tamaño mediano a grande ( $d=0,759$ );

- un aumento estadísticamente significativo [ $t(19) = 3,863$ ;  $p=0,001$ ] en la faceta *Gestión del Estrés*, desde el pretest ( $M=4,15000$ ) hasta el posttest ( $M=4,56500$ ), con un efecto de tamaño grande ( $d=0,864$ );
- un aumento estadísticamente significativo [ $t(19) = 3,096$ ;  $p=0,006$ ] en el factor *Bienestar*, desde el pretest ( $M=4,50741$ ) hasta el posttest ( $M=4,74630$ ), con un efecto de tamaño mediano a grande ( $d=0,692$ );
- un aumento estadísticamente significativo [ $t(19) = 3,028$ ;  $p=0,007$ ] en el factor *Emocionalidad*, desde el pretest ( $M=4,37632$ ) hasta el posttest ( $M=4,59342$ ), con un efecto de tamaño mediano a grande ( $d=0,677$ );
- un aumento estadísticamente significativo [ $t(19) = 2,500$ ;  $p=0,022$ ] en el factor *Sociabilidad*, desde el pretest ( $M=4,25517$ ) hasta el posttest ( $M=4,45345$ ), con un efecto de tamaño mediano a grande ( $d=0,559$ );
- y un aumento estadísticamente significativo [ $t(19) = 3,449$ ;  $p=0,003$ ] en el factor global de *Inteligencia Emocional*, desde el pretest ( $M=4,34167$ ) hasta el posttest ( $M=4,52882$ ), con un efecto de tamaño mediano a grande ( $d=0,771$ ).

### 9.3. Análisis cualitativo: Categorización de percepciones

Las entrevistas cuyas transcripciones completas se incluyen en el anexo 2 fueron procesadas mediante el software ATLAS.ti 6.2. con el objetivo encontrar —en los conceptos expresados por los estudiantes— indicios que reforzaran los resultados del análisis cuantitativo llevado a cabo en el punto anterior.

#### 9.3.1. Adquisición de conocimientos de programación orientada a objetos

Debido a que el análisis cuantitativo había indicado la existencia de una relación estadísticamente significativa —una asociación moderada— entre la estrategia lúdica 2 (*Aprendizaje mediante la utilización de videojuegos*) y la *capacidad de identificar las relaciones entre clases en UML*, primero se seleccionaron y contaron los conocimientos que —en la opinión de los estudiantes entrevistados— fueron adquiridos en la materia *Programación II* hasta el momento de la finalización de cada experiencia (tabla 116) y luego se los agrupó en función de la estrategia lúdica empleada (figura 69).

Tabla 116

*Percepción de los conocimientos adquiridos en función de la estrategia lúdica empleada.*

Estrategia lúdica	Conocimiento	Extracto de la entrevista
Grupo de Control	01-Abstracción {2}	<p>► Tengo dominado y entendido el concepto de herencia, de polimorfismo, de <b>métodos abstractos</b> (TRADI-04).</p> <p>► Entre los conceptos aprendidos a través de la cursada de Programación II, creo que reforcé mucho mis conocimientos... Conceptos sobre lo que son los objetos, la herencia, la <b>abstracción</b>. Creo que ref... Pude reforzar mucho mis conocimientos de programación orientada a objetos y de la programación en sí (TRADI-05).</p>
	03-Buenas prácticas de programación {1}	<p>► Hoy día, este... Me llevé mucho, en cuanto a Programación II, dominar la sintaxis de los lenguajes que me enseñaron, este... Las estructuras, este... De datos de esos lenguajes. El cual me permitió, hoy día, este... <b>Programar tal vez un poquito más eficientemente o, tal vez, con mayor conciencia de... De lo que estoy escribiendo</b> -digamos- y que sea eficiente para... Para el requerimiento que me están dando (TRADI-06).</p>
	04-Desarrollo de interfaces gráficas {1}	<p>► La parte de <b>hacer las partes... La parte gráfica, de las ventanas, botones</b>. Esas partes (TRADI-01).</p>
	05-Estructuras de datos {1}	<p>► Hoy día, este... Me llevé mucho, en cuanto a Programación II, dominar la sintaxis de los lenguajes que me enseñaron, este... <b>Las estructuras, este... De datos de esos lenguajes</b>. El cual me permitió, hoy día, este... Programar tal vez un poquito más eficientemente o, tal vez, con mayor conciencia de... De lo que estoy escribiendo -digamos- y que sea eficiente para... Para el requerimiento que me están dando (TRADI-06).</p>
	06-Herencia {3}	<p>► Eh... Creo que... Lo que más llego a dominar es... Eh... El tema de <b>herencia</b> y polimorfismo. Eso creo que me quedó bastante claro y... Y me pareció muy útil (TRADI-02).</p> <p>► Tengo dominado y entendido el concepto de <b>herencia</b>, de polimorfismo, de métodos abstractos (TRADI-04).</p> <p>► Entre los conceptos aprendidos a través de la cursada de Programación II, creo que reforcé mucho mis conocimientos... Conceptos sobre lo que son los objetos, la <b>herencia</b>, la abstracción. Creo que ref... Pude reforzar mucho mis conocimientos de programación orientada a objetos y de la programación en sí (TRADI-05).</p>
	07-Modelado de objetos {1}	<p>► Creo que lo que más intento... Domino e intento seguir dominando... Es entender las responsabilidades, la <b>modularización de cada objeto</b>, ¿sí? Que cada objeto tenga funcionalidades bien definidas, que sea un ente -¿no?- bien cerrado, con sus responsabilidades que nadie... que ningún otro objeto debe saber cómo son, y que se tengan que comunicar entre sí mediante mensajes, que es lo que reza -en realidad- el paradigma, antes de meterse -como digo- en cualquier otro concepto (TRADI-04).</p>
	08-Patrón MVC {1}	<p>► Eh... Que es con objetos... Objetos y... ¡<b>MVC!</b> (TRADI-03)</p>
	10-Polimorfismo {2}	<p>► Eh... Creo que... Lo que más llego a dominar es... Eh... El tema de herencia y <b>polimorfismo</b>. Eso creo que me quedó bastante claro y... Y me pareció muy útil (TRADI-02).</p> <p>► Tengo dominado y entendido el concepto de herencia, de <b>polimorfismo</b>, de métodos abstractos (TRADI-04).</p>
	11-Programación orientada a objetos {1}	<p>► Entre los conceptos aprendidos a través de la cursada de Programación II, creo que reforcé mucho mis conocimientos... Conceptos sobre lo que son los objetos, la herencia, la abstracción. Creo que ref... Pude reforzar mucho mis conocimientos de <b>programación orientada a objetos</b> y de la programación en sí (TRADI-05).</p>
	14-Sintaxis de los lenguajes {1}	<p>► Hoy día, este... Me llevé mucho, en cuanto a Programación II, dominar la <b>sintaxis de los lenguajes</b> que me enseñaron, este... Las estructuras, este... De datos de esos lenguajes. El cual me permitió, hoy día, este... Programar tal vez un poquito más eficientemente o, tal vez, con mayor conciencia de... De lo que estoy escribiendo -digamos- y que sea eficiente para... Para el requerimiento que me están dando (TRADI-06).</p>

*Nota:* Elaboración propia. (Obs.: Los datos demográficos de los estudiantes se encuentran en el anexo 2)

Tabla 116 (cont.)

Percepción de los conocimientos adquiridos en función de la estrategia lúdica empleada.

Estrategia lúdica	Conocimiento	Extracto de la entrevista
Tratamiento 1 (Desarrollo de VJ)	02-Bases de datos {2}	<p>► Bue... Vimos un poco de <b>bases de datos</b>, que -bah- vimos bastantes clases. Habremos visto tres o cuatro clases. Para mí, yo me siento más fuerte en eso justamente porque puedo ir mezclando con lo que ya sé de SQL (DESVJ-01).</p> <p>► Lo que yo manejo mejor es <b>bases de datos</b>. Ahí es donde yo me siento bastante más cómodo. Tiene que ver no solo por lo aprendido ahora sino por -¿cómo es, eh?- un poco los conocimientos previos, ¿no? (DESVJ-03).</p>
	06-Herencia {2}	<p>► Los que más domino quizás sea lo que es la programación orientada a objetos, el tema de polimorfismos, <b>herencias</b>. También porque es un tema que me apasiona. Ya lo sabía igual de antes, en cierta forma, pero creo que el que más domino es la programación orientada a objetos en sí (DESVJ-02).</p> <p>► Yo creo que el concepto más fuerte que me llevé es el tema de las <b>herencias</b> y cómo trabajar también el tema de los objetos. Mayoritariamente eso (DESVJ-04).</p>
	07-Modelado de objetos {1}	<p>► Yo creo que el concepto más fuerte que me llevé es el tema de las herencias y cómo trabajar también <b>el tema de los objetos</b>. Mayoritariamente eso (DESVJ-04).</p>
	09-Patrones de diseño {1}	<p>► Algo que estuvo muy bueno verlo es el tema referido a <b>patrones de diseño</b>, que le da cierta agilidad cuando te tenés que enfrentar a un problema. No es lo mismo enfrentarlo como yo lo hacía hace un año que como ahora. Tengo un poco más... Tengo más herramientas para hacerlo (DESVJ-01).</p>
	10-Polimorfismo {1}	<p>► Los que más domino quizás sea lo que es la programación orientada a objetos, el tema de <b>polimorfismos</b>, herencias. También porque es un tema que me apasiona. Ya lo sabía igual de antes, en cierta forma, pero creo que el que más domino es la programación orientada a objetos en sí (DESVJ-02).</p>
	11-Programación orientada a objetos {2}	<p>► Los que más domino quizás sea lo que es la <b>programación orientada a objetos</b>, el tema de polimorfismos, herencias. También porque es un tema que me apasiona. Ya lo sabía igual de antes, en cierta forma, pero creo que el que más domino es la programación orientada a objetos en sí (DESVJ-02).</p> <p>► Bueno, conceptos que empecé a dominar ahora son: lo básico de la <b>programación orientada a objetos</b>, son cosas que verdaderamente -si lo tenés bien claro todas esas cosas- se te va a hacer más fácil la programación que lo que es más avanzado en Java (DESVJ-05).</p>
Tratamiento 2 (Uso de VJ)	02-Bases de datos {1}	<p>► Lo que estamos viendo ahora... Lo de Modelo-Vista-Controlador, eh... Y <b>bases de datos</b>. Es lo que... Es lo últimos que vimos (USOVJ-01).</p>
	03-Buenas prácticas de programación {1}	<p>► Eh... Bueno, <b>buenas prácticas de programación</b>. Eh... Buenas prácticas de programación en sí (USOVJ-03).</p>
	06-Herencia {1}	<p>► Polimorfismo, <b>herencia</b>, eh... (USOVJ-03).</p>
	07-Modelado de objetos {2}	<p>► Mmm... <b>El modelado de objetos</b>.. (USOVJ-02).</p> <p>► Creo que... Armar el proyecto... Como intentar pensarlo... Como previo a este manejo de clases... Intentar pensarlo. Después <b>plantear las clases</b>, quizás me... Me mezclo un poco más. Pero... Pero el planteo, creo que estoy más fuerte (USOVJ-04).</p>
	08-Patrón MVC {1}	<p>► Lo que estamos viendo ahora... Lo de <b>Modelo-Vista-Controlador</b>, eh... Y bases de datos. Es lo que... Es lo últimos que vimos (USOVJ-01).</p>
10-Polimorfismo {1}	<p>► <b>Polimorfismo</b>, herencia, eh... (USOVJ-03)</p>	

Nota: Elaboración propia. (Obs.: Los datos demográficos de los estudiantes se encuentran en el anexo 2)

Tabla 116 (cont.)

Percepción de los conocimientos adquiridos en función de la estrategia lúdica empleada.

Estrategia lúdica	Conocimiento	Extracto de la entrevista
Tratamiento 2 (Uso de VJ) (cont.)	12-Programación web {1}	<p>► Eh... No había visto <b>JSP</b>, no sabía lo que era un <i>servlet</i>, eh... No sabía... Bueno, no sabía bastantes cosas, en realidad, la verdad que me cuesta puntualizar los conceptos, etc. Pero, la verdad que, hasta ahora, estoy aprendiendo mucho y todavía quedan un par de clases, pero no muchas más, pero me llevo -hasta ahora- mucho contenido... aplicable (USOVJ-03).</p>
	13-Relaciones entre clases {3}	<p>► Aprendí, eh... Por ejemplo, tipos de <b>relaciones entre clases</b>, eh... No digo que a la fuerza, pero sí como más subliminalmente (USOVJ-02).</p> <p>► El de las flechitas, que era un juego en el que vos tenías que ir embocando — digamos — disparándoles a las <b>relaciones de las distintas clases</b> en función del significado. Me resultó mnemotécnico (USOVJ-04).</p> <p>► Bueno, creo que los conceptos que más me quedaron grabados fueron las diferentes <b>relaciones que puede haber entre las clases</b>. Eh... Tuvimos varias horas dedicadas a eso. Además... Eh... Los videojuegos que jugamos eran algunos con respecto a eso también, así que me quedó bastante fijo... Ese... Me quedaron bastante fijos esos conceptos (USOVJ-05).</p>
Tratamiento 3 (Ludificación)	06-Herencia {4}	<p>► Y, por ahora... La parte de... de <b>herencia</b> y polimorfismo (LUDIF-02).</p> <p>► Creo que... Uy, ¡qué difícil de contestar! La verdad que dentro de lo que es Programación II... no creo que domine demasiado uno específicamente. Pero... la... lo que es <b>herencia</b> y polimorfismo, digamos, es lo que... de lo que tengo un poco más de conocimiento -por ahí-dentro de lo que es... el lenguaje que se dio y la Programación Orientada a Objetos y... Sí, sería... sería... más o menos esa ... esa... esa parte, digamos (LUDIF-03).</p> <p>► Eh... todo lo referido a <b>herencias</b> lo estuve practicando un montón y estuve haciéndome unos machetes y cosas al respecto. Eh... Particularmente con eso, fue lo que me impresionó, me interesó porque... Para poder armar bien las estructuras y demás... es lo básico, totalmente (LUDIF-04).</p> <p>► Eh... Creo que polimorfismo y <b>herencia</b> (LUDIF-05).</p>
	07-Modelado de objetos {1}	<p>► Bueno, considero que mejor domino... No sé si es una unidad en particular, pero... sería cómo encabezar el proyecto, es decir, si nos dieran un programa para armar desde cero la parte que mejor domino sería la <b>organización de las clases</b>, es decir, cómo va a estar organizado un proyecto por distintas clases, que cada clase tenga sus atributos y sus métodos y las relaciones que van a tener entre ellas para que el programa arroje los resultados esperados (LUDIF-01).</p>
	10-Polimorfismo {3}	<p>► Y, por ahora... La parte de... de herencia y <b>polimorfismo</b> (LUDIF-02).</p> <p>► Creo que... Uy, ¡qué difícil de contestar! La verdad que dentro de lo que es Programación II... no creo que domine demasiado uno específicamente. Pero... la... lo que es herencia y <b>polimorfismo</b>, digamos, es lo que... de lo que tengo un poco más de conocimiento -por ahí-dentro de lo que es... el lenguaje que se dio y la Programación Orientada a Objetos y... Sí, sería... sería... más o menos esa ... esa... esa parte, digamos (LUDIF-03).</p> <p>► Eh... Creo que <b>polimorfismo</b> y herencia (LUDIF-05).</p>
	13-Relaciones entre clases {1}	<p>► Bueno, considero que mejor domino... No sé si es una unidad en particular, pero... sería cómo encabezar el proyecto, es decir, si nos dieran un programa para armar desde cero la parte que mejor domino sería la organización de las clases, es decir, cómo va a estar organizado un proyecto por distintas clases, que cada clase tenga sus atributos y sus métodos y las <b>relaciones que van a tener entre ellas para que el programa arroje los resultados esperados</b> (LUDIF-01).</p>

Nota: Elaboración propia. (Obs.: Los datos demográficos de los estudiantes se encuentran en el anexo 2)



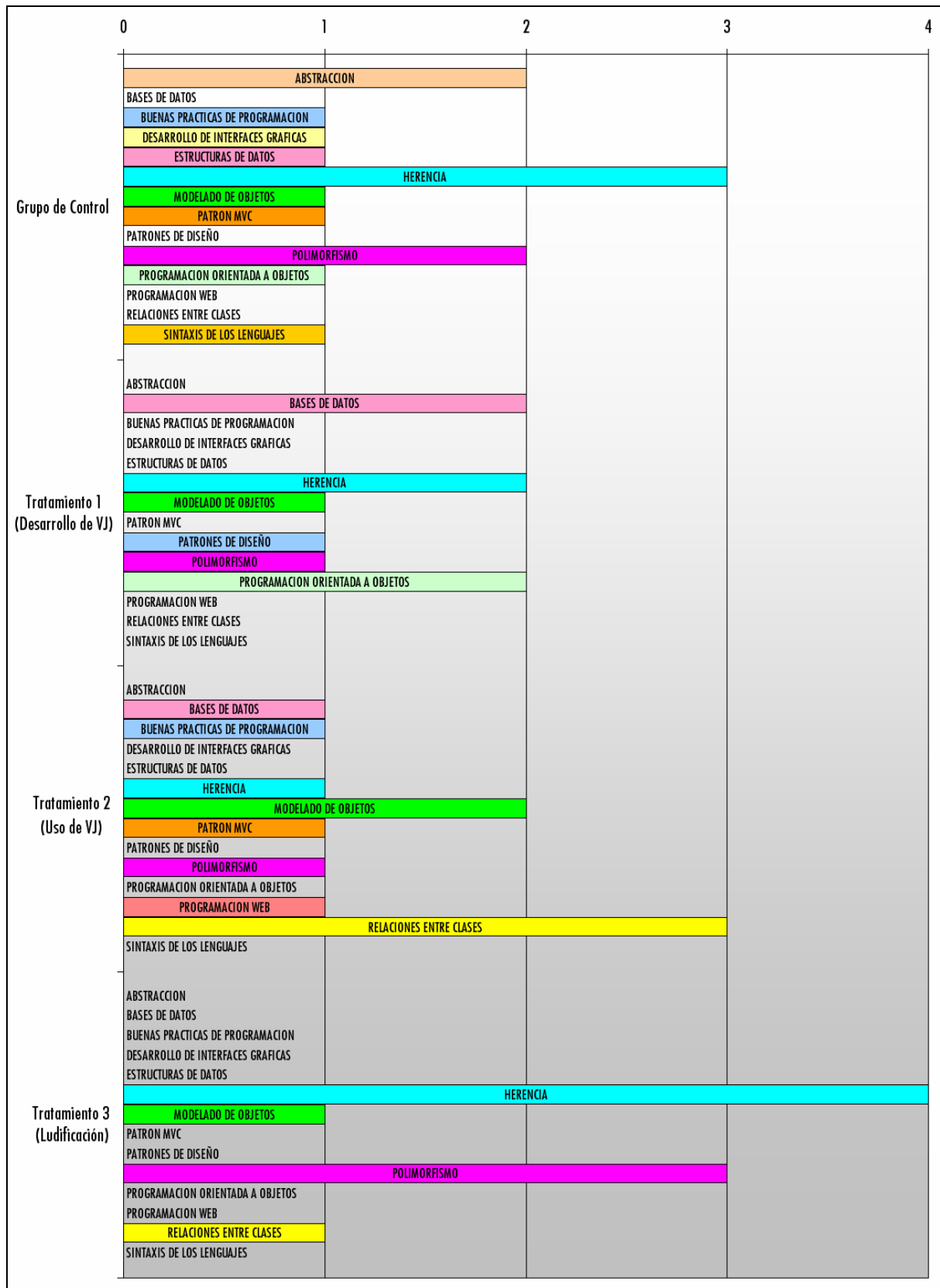


Figura 69. Percepción de los conocimientos adquiridos en función de la estrategia lúdica empleada.

Elaboración propia

Efectivamente, la *capacidad de identificar las relaciones entre clases en UML* fue mencionada por tres estudiantes del grupo que fue sometido al segundo tratamiento (*Aprendizaje mediante la utilización de videojuegos*):

Aprendí, eh... Por ejemplo, tipos de *relaciones entre clases*, eh... No digo que a la fuerza, pero sí como más subliminalmente (USOVJ-02).

El de las flechitas, que era un juego en el que vos tenías que ir embocando —digamos— disparándoles a las *relaciones de las distintas clases* en función del significado. Me resultó mnemotécnico (USOVJ-04).

Bueno, creo que los conceptos que más me quedaron grabados fueron las diferentes *relaciones que puede haber entre las clases*. Eh... Tuvimos varias horas dedicadas a eso. Además... Eh... Los videojuegos que jugamos eran algunos con respecto a eso también, así que me quedó bastante fijo... Ese... Me quedaron bastante fijos esos conceptos (USOVJ-05).

Uno de los juegos usados por estos estudiantes fue *The Java Mania Game* (figura 70), que probablemente sea el que se mencionó en el comentario de USOVJ-04.



Figura 70. Videojuego serio *The Java Mania Game*. Elaboración propia

El comentario de los estudiantes anteriores es un indicio importante para intentar explicar la relación encontrada mediante el análisis cuantitativo. El hecho de que se trate de una asociación apenas *moderada* se hace evidente por la presencia —en otro grupo, el que fue sometido al tercer tratamiento (*Ludificación del aprendizaje/Gamification*), lo que le resta fuerza a la asociación anterior— del siguiente comentario del estudiante LUDIF-01, afirmando que el tema que más dominaba eran las relaciones entre clases:

Bueno, considero que mejor domino... No sé si es una unidad en particular, pero... sería cómo encabezar el proyecto, es decir, si nos dieran un programa para armar desde cero la parte que mejor domino sería la organización de las clases, es decir, cómo va a estar organizado un proyecto por distintas clases, que cada clase tenga sus atributos y sus métodos y *las relaciones que van a tener entre ellas* para que el programa arroje los resultados esperados (LUDIF-01).

El análisis cuantitativo no había indicado la existencia de ninguna otra relación estadísticamente significativa entre cualquiera de las estrategias lúdicas empleadas y cualquiera de las otras capacidades —referidas a los conocimientos específicos de la programación orientada a objetos— que fueron evaluadas.

De hecho, en la figura 69 puede verse que los temas *modelado de objetos*, *herencia* y *polimorfismo* —que se encuentran entre los principales conceptos de la programación orientada a objetos— fueron mencionados por estudiantes de todos los grupos, independientemente de la estrategia lúdica empleada.

### 9.3.2. Relaciones interpersonales en el aula

Debido a que el análisis cuantitativo había indicado una reducción estadísticamente significativa del índice *distancia social media propia* en tres de los cuatro cursos —la excepción fue el curso en el cual se empleó la estrategia lúdica 3 (*Ludificación del aprendizaje/Gamification*)— y del índice *distancia social media del grupo* en todos los cursos, además de la existencia de diferencias estadísticamente significativas entre las medias marginales estimadas del grupo de control y las del curso en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*), así como también entre las de este último y las del curso en el cual se empleó la estrategia lúdica 3, a continuación se seleccionaron y contaron los comentarios referidos a las *relaciones en el grupo* (tabla 117) y luego se los juntó en función de la estrategia lúdica empleada (figura 71).

Tabla 117

Percepción de las relaciones en el grupo en función de la estrategia lúdica empleada.

Estrategia lúdica	Cant.	Extracto de la entrevista
Grupo de Control	0	No se hallaron comentarios referidos a las relaciones interpersonales
Tratamiento 1 (Desarrollo de VJ)	4	<p>► [El videojuego] creo que <b>estuvo bueno desarrollarlo en grupo</b>. Solo hubiera sido bastante complicado. No hubiera sido lo mismo (DESVJ-02).</p> <p>► Gaby lo desarrollaba y nos pasaba y hacíamos las pruebas. <b>Nos manejamos prácticamente por internet</b>. No nos reunimos (DESVJ-03).</p> <p>► [Desarrollar el videojuego] <b>me gustó. A mi grupo también</b>. Nos gustó. Nos gustaría haberle dedicado un poco más de tiempo porque fuimos muy puntuales en cuanto a las necesidades y a las cosas que íbamos teniendo, con la idea de llegar a la fecha límite, a la fecha de entrega (DESVJ-04).</p> <p>► <b>Nos dábamos a cada persona algo para hacer</b> y al siguiente sábado era como que teníamos que traerlo hecho para después sumarlo todo al proyecto y ver si compilaba o no (DESVJ-05).</p>
Tratamiento 2 (Uso de VJ)	1	<p>► <b>Estuvo bueno jugar a los videojuegos en grupo</b>. Porque... Como había que recurrir a los conocimientos aprendidos en clase, por ahí, <b>lo que no se acordaba uno, lo decía el otro</b>. Y de esa forma íbamos recordando y estudiando a la vez (USOVJ-05).</p>
Tratamiento 3 (Ludificación)	2	<p>► Sabía que nos iban a preparar en cuanto a... no solo lo que conlleva programar, sino además de <b>programar en grupo</b>, ya que hoy en día las empresas o cualquier negocio que necesite un sistema en sí, armado, no necesariamente tiene que ser un solo programador el que lo haga, generalmente son grupos de programadores que trabajan entre sí . . . El tema la competencia, es algo... un aspecto muy realista, porque cuando salgamos de la carrera, digamos, cuando estemos recibidos y vayamos al campo laboral, vamos a ver que se ve así tal cual en un ambiente competitivo en donde cada persona o <b>cada grupo de desarrolladores va a tener que... Que estar juntos</b>, eh... cada uno con sus conocimientos para poder sacar adelante su proyecto a comparación de otros (LUDIF-01).</p> <p>► La utilización de GitHub y GitLab . . . agiliza mucho y además prepara a los alumnos para un ambiente más de <b>trabajo colaborativo</b> y para un ambiente de trabajo fuera de la facultad . . . La verdad que <b>en el grupo que se formó</b>, por lo menos este año, no vi esa situación [de rivalidad], sino más bien una cuestión de juego bien tomado y... y risa, digamos, de poder competir sanamente y le agregé un... Para mí funcionó porque le agregé un 'plus' a venir a cursar un viernes a la noche, que era muy importante y muy difícil a veces (LUDIF-03).</p>

Nota: Elaboración propia. (Obs.: Los datos demográficos de los estudiantes se encuentran en el anexo 2)

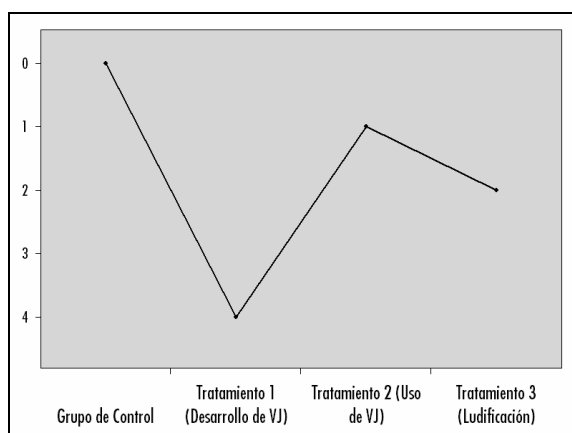


Figura 71. Percepción de las relaciones en el grupo en función de la estrategia lúdica empleada.

Elaboración propia

Comparando las figuras 71 (*Percepción de las relaciones en el grupo en función de la estrategia lúdica empleada*) y 35 (*Medias marginales estimadas de Rango de POS-DSG*) —las cuales representan, respectivamente, los resultados del análisis cualitativo y del análisis cuantitativo referidos a las relaciones interpersonales en el aula— pudieron observarse ciertas semejanzas. Si bien la reducción del índice *distancia social media del grupo* se había producido en todos los cursos, los resultados de ambos análisis coincidieron en que el grupo de control y el grupo en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*) estuvieron en los extremos de las mediciones de esta dimensión.

### 9.3.3. Inteligencia emocional de los estudiantes

El análisis cuantitativo había indicado, entre el pretest y el postest, un aumento estadísticamente significativo en la faceta *Conciencia Social* —en el grupo que empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*)—, una reducción estadísticamente significativa en la faceta *Control de la Impulsividad* —en el curso en el cual se empleó la estrategia lúdica 2 (*Aprendizaje mediante la utilización de videojuegos*)— y aumentos estadísticamente significativos en las facetas *Asertividad*, *Autoestima*, *Automotivación*, *Conciencia Social*, *Expresividad Emocional* y *Gestión del Estrés*, en los factores *Bienestar*, *Emocionalidad* y *Sociabilidad*, y en el factor global de *Inteligencia Emocional* —en el grupo en el cual se empleó la estrategia lúdica 3 (*Ludificación del aprendizaje/Gamification*)—, además de la existencia —entre los grupos formados en función de la estrategia lúdica empleada— de diferencias significativas en el postest de la faceta *Conciencia Social*, una vez eliminada la influencia del pretest, estas últimas diferencias basadas en la existencia de diferencias estadísticamente significativas entre las medias marginales estimadas del grupo de control y las del curso en el cual se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*).

Por ello, a continuación se seleccionaron y contaron los comentarios referidos a la *inteligencia emocional de los estudiantes* (tabla 118) y luego se los agrupó en función de la estrategia lúdica empleada (figura 72).

Tabla 118

*Percepción de las facetas de la inteligencia emocional en función de la estrategia lúdica empleada.*

Estrategia lúdica	Faceta de la IE	Extracto de la entrevista
Grupo de Control	Ninguna	No se hallaron comentarios referidos a las facetas de la inteligencia emocional
Tratamiento 1 (Desarrollo de VJ)	03-Autoestima {1}	► Fue algo que me costó al principio, pero <b>pude tomar la mejor elección</b> , que fue dividir en partes que estén desacopladas (DESVJ-01).
	04-Automotivación {3}	► <b>Lo que más me motivó, y supongo que a mi compañero también, fue hacer un juego</b> , ¿no? Nunca me había imaginado en la facu poder llegar a hacerlo. Sí, desde chiquito me fascinaba el tema de los jueguitos y siempre quería hacer uno. Cruzarse con esto acá fue bastante bueno (DESVJ-01). ► <b>Y la motivación era terminarlo y aprobar</b> (DESVJ-03). ► <b>[Me motivó] aprender a hacer un juego que es...</b> que creo que va a ser uno de... o es el tema que yo tengo pensado para entregar como trabajo final de la carrera (DESVJ-04).
	05-Conciencia Social {1}	► El videojuego era como que lo veía imposible... realizarlo. A medida que uno se va proponiendo hacer las cosas -con ayuda del profesor, obviamente- se le va haciendo más fácil el poder programar y llegar a la meta final. Pero, la verdad que me asombró porque <b>-con voluntad y confianza... y el grupo también- se pudo llegar . . .</b> Después -poco a poco- <b>nos fuimos animando entre nosotros</b> (DESVJ-05).
	10-Gestión de la Emoción {1}	► No sé por qué ese juego me gustaba, me marcó como una etapa de mi infancia y bueno, cuando tuvimos la posibilidad de hacer un juego, nuestro juego estaba basado un poco también en eso . . . <b>Quise... quisimos hacer un poco algo parecido a ese juego que algo... algo tenía porque realmente fue algo que marcó una etapa importante de mi primaria</b> (DESVJ-02).
	11-Gestión del Estrés {2}	► Fuimos muy puntuales en cuanto a las necesidades y a las cosas que íbamos teniendo, con la idea de llegar a la fecha límite, a la fecha de entrega. Sí, <b>fue un esfuerzo en el sentido de... desde aprender algo nuevo, organizar los tiempos</b> (DESVJ-04). ► <b>Le tuvimos que dedicar muchísimo -pero muchísimo- tiempo</b> (DESVJ-05).
15-Relaciones {2}	► <b>Cosas donde no tengamos que depender del otro</b> (DESVJ-01). ► Y... Nada, aparte <b>lo hicimos con amigos, con personas que nos llevábamos bien</b> , así que, nada, pudimos hacerlo bien. Y sí, le dedicamos mucho tiempo, mucho esfuerzo (DESVJ-02).	
Tratamiento 2 (Uso de VJ)	04-Automotivación {1}	► Le dedicamos tiempo, sí, porque... Más que nada, el de las flechitas, me acuerdo, por el tema de que <b>-al principio- no recordábamos bien cuáles eran... Intentábamos avanzar y no avanzábamos. Pero... Creo que después de practicarlo bastante, algunas cosas fueron quedando</b> (USOVJ-04).
	13-Percepción Emocional {1}	► Bueno, me gustó utilizarlos. Voy a ser sincero, la verdad que no los utilicé demasiado, pero sí <b>a mis compañeros les gustó</b> y los vi muy eh... -como se dice acá en Argentina- 'copados' con ellos (USOVJ-03).
Tratamiento 3 (Ludificación)	04-Automotivación {1}	► <b>Hacerlo de esta manera se me hizo una forma más... más dinámica y más motivadora</b> , de alguna forma (LUDIF-04).
	07-Empatía {2}	► Como en todos lados, <b>hay personas que están más capacitadas que otras, y otras que no. Y capaz que se ven, eh... en cierta forma, desmotivadas por esa competencia</b> que no es, en cierta forma, tan... justa (LUDIF-02). ► Eh... Me parece que <b>aporta competitividad, un poco, y motivación para que... que no se duerman los chicos</b> , que traten de practicar para ver si... si pueden elegir un buen tema de parcial (LUDIF-05).

*Nota:* Elaboración propia. (Obs.: Los datos demográficos de los estudiantes se encuentran en el anexo 2)

Tabla 118 (cont.)

Percepción de las facetas de la inteligencia emocional en función de la estrategia lúdica empleada.

Estrategia lúdica	Faceta de la IE	Extracto de la entrevista
Tratamiento 3 (Ludificación) (cont.)	11-Gestión del Estrés {1}	► Prácticamente <b>me obligó a estar al día en la materia</b> , eh... no perder estabilidad en cuanto a las clases, esto significa que, por <b>cada semana, además de que aprendí algo nuevo, si o si tenía que reflexionar en lo que habíamos visto la clase anterior</b> para poder estar al día con el desafío y no quedar atrás. (LUDIF-01).
	13-Percepción Emocional {2}	► Al principio... debo decir que me era difícil, por el hecho de que... <b>tengo cierto rechazo a la cuestión competitiva...</b> eh... porque tengo esa idea de... el otro que, por ser competitivo, trata de hacerte el mal y no de compartir la competitividad en un buen sentido. Pero, la verdad que <b>en el grupo que se formó, por lo menos este año, no vi esa situación, sino más bien una cuestión de juego bien tomado y... y risa, digamos, de poder competir sanamente</b> y le agregé un... Para mí funcionó porque le agregé un <i>plus</i> a venir a cursar un viernes a la noche, que era muy importante y muy difícil a veces (LUDIF-03).  ► Eso me incentivó un poquito más para poder... eh... dedicarle más porque soy eh... - mal que mal- <b>soy una persona que me gusta competir, me gusta jugar</b> (LUDIF-04).

Nota: Elaboración propia. (Obs.: Los datos demográficos de los estudiantes se encuentran en el anexo 2)

Estrategia Lúdica	Facetas de la IE	Análisis Cualitativo			Análisis Cuantitativo
Grupo de Control					
Tratamiento 1 (Desarrollo de VJ)	01-Adaptabilidad				
	02-Asertividad				
	03-Autoestima	■	■	■	
	04-Automotivación	■	■	■	
	05-Conciencia Social	■			AUMENTO SIGNIFICATIVO
	06-Control de la Impulsividad				
	07-Empatía				
	08-Expresividad Emocional				
	09-Felicidad				
	10-Gestión de la Emoción	■	■	■	
	11-Gestión del Estrés	■	■	■	
	12-Optimismo				
	13-Percepción Emocional				
	14-Regulación Emocional				
	15-Relaciones	■	■	■	
Tratamiento 2 (Uso de VJ)	01-Adaptabilidad				
	02-Asertividad				
	03-Autoestima				
	04-Automotivación	■			
	05-Conciencia Social				
	06-Control de la Impulsividad				REDUCCIÓN SIGNIFICATIVA
	07-Empatía				
	08-Expresividad Emocional				
	09-Felicidad				
	10-Gestión de la Emoción				
	11-Gestión del Estrés				
	12-Optimismo				
	13-Percepción Emocional	■			
	14-Regulación Emocional				
	15-Relaciones				
Tratamiento 3 (Ludificación)	01-Adaptabilidad				
	02-Asertividad				AUMENTO SIGNIFICATIVO
	03-Autoestima				AUMENTO SIGNIFICATIVO
	04-Automotivación	■			AUMENTO SIGNIFICATIVO
	05-Conciencia Social				AUMENTO SIGNIFICATIVO
	06-Control de la Impulsividad				
	07-Empatía	■	■	■	
	08-Expresividad Emocional				AUMENTO SIGNIFICATIVO
	09-Felicidad				
	10-Gestión de la Emoción				
	11-Gestión del Estrés	■			AUMENTO SIGNIFICATIVO
	12-Optimismo				
	13-Percepción Emocional	■	■	■	
	14-Regulación Emocional				
	15-Relaciones				

Figura 72. Percepción de las facetas de la IE en función de la estrategia lúdica empleada. Elaboración propia

Se observaron ciertas coincidencias entre los resultados de los análisis cuantitativo y cualitativo. En primer lugar —y tal como ya había ocurrido al analizar las *relaciones interpersonales en el aula*—, los estudiantes del grupo de control no se expresaron sobre ningún aspecto referido a la dimensión estudiada (en este caso, la *inteligencia emocional*), por lo que a la inexistencia de diferencias estadísticamente significativas entre el pretest y el postest en ese curso le correspondió otra inexistencia: la de percepciones para analizar.

En el grupo donde se empleó la estrategia lúdica 1 (*Aprendizaje mediante el desarrollo de videojuegos*) —a favor del cual el análisis cuantitativo había señalado diferencias estadísticamente significativas en la faceta *Conciencia Social*— surgió el siguiente comentario que es un indicio importante para intentar explicar esas diferencias:

El videojuego era como que lo veía imposible... realizarlo. A medida que uno se va proponiendo hacer las cosas -con ayuda del profesor, obviamente- se le va haciendo más fácil el poder programar y llegar a la meta final. Pero, la verdad que me asombró porque —con voluntad y confianza... y el grupo también— se pudo llegar . . . Después —poco a poco— nos fuimos animando entre nosotros (DESVJ-05).

En el grupo donde se empleó la estrategia lúdica 2 (*Aprendizaje mediante la utilización de videojuegos*) no se encontraron indicios relevantes, pero sí en el grupo en el cual se empleó la estrategia lúdica 3 (*Ludificación del aprendizaje/Gamification*). La faceta *Automotivación* —en la cual el análisis cuantitativo había señalado diferencias estadísticamente significativas— puede relacionarse con el siguiente comentario:

Hacerlo de esta manera [cursar la materia de forma ludificada] se me hizo una forma más... más dinámica y más motivadora, de alguna forma (LUDIF-04).

En este último grupo, el siguiente comentario también es un indicio importante para intentar explicar las diferencias estadísticas encontradas en la faceta *Gestión del Estrés*:

[Cursar la materia de forma ludificada] prácticamente me obligó a estar al día en la materia, eh... no perder estabilidad en cuanto a las clases, esto significa que, por cada semana, además de que aprendí algo nuevo, sí o sí tenía que reflexionar en lo que habíamos visto la clase anterior para poder estar al día con el desafío y no quedar atrás. (LUDIF-01).



## DISCUSIÓN Y CONCLUSIONES

---

La enseñanza de la programación ha recorrido un largo camino desde los primeros cursos dictados en ambientes académicos en la década de 1950 hasta la actualidad, cuando —más que nunca— es imperativo formar un gran número de programadores para sostener el desarrollo de la sociedad del conocimiento en que vivimos.

Esto representa un enorme desafío, debido a que —entre otras razones— a menudo la programación es vista como una materia difícil (T. Jenkins, 2002, p. 55), y es por ello que en este estudio se investigó empíricamente el potencial de tres estrategias lúdicas (*Aprendizaje mediante el desarrollo de videojuegos*, *Aprendizaje mediante la utilización de videojuegos* y *Ludificación del aprendizaje/Gamification*) como alternativas para la mejora de la enseñanza de esta disciplina en la Educación Superior.

### 10.1. Discusión

A continuación se presenta la discusión de los resultados de la implementación de cada una de las estrategias lúdicas anteriores, a la luz de los estudios previos oportunamente mencionados en el marco teórico.

#### 10.1.1. Discusión sobre *el aprendizaje mediante la utilización de videojuegos en la enseñanza y el aprendizaje de la programación*

Los resultados del estudio indican que la implementación del *aprendizaje mediante la utilización de videojuegos* —específicamente, juegos serios— en la enseñanza y el aprendizaje de la programación tuvo un impacto positivo sobre la adquisición de los conocimientos específicos de la disciplina, particularmente en lo referido a la *capacidad de identificar las relaciones entre clases en UML*. Esto está acorde —en líneas generales— con lo que informan García-Mundo, Vargas-Enríquez, Genero y Piattini (2014, pp. 305-308)

en su estudio *¿Contribuye el Uso de Juegos Serios a Mejorar el Aprendizaje en el Área de la Informática?*

### **10.1.2. Discusión sobre *el aprendizaje mediante el desarrollo de videojuegos* en la enseñanza y el aprendizaje de la programación**

En cuanto a la implementación del *aprendizaje mediante el desarrollo de videojuegos* —específicamente, juegos serios— en la enseñanza y el aprendizaje de la programación, los resultados del estudio indican que esta tuvo un impacto positivo sobre las relaciones interpersonales en el aula, particularmente en lo referido a la disminución de la *distancia social media del grupo* y el aumento de los valores medidos en la faceta *Conciencia Social* —que forma parte de la inteligencia emocional rasgo de los estudiantes—. A grandes rasgos, esto está en consonancia con las afirmaciones de Roschelle, Pea, Hoadley, Gordin y Means (2000, p. 80), los hallazgos del proyecto MAGICAL (European Commission - Education, Audiovisual & Culture Executive Agency, 2014) y los resultados del plan Globaloria (Kafai y Burke, 2015, pp. 322-323).

### **10.1.3. Discusión sobre *la ludificación del aprendizaje/Gamification* en la enseñanza y el aprendizaje de la programación**

Los resultados de la investigación sugieren que la implementación de la *ludificación del aprendizaje/Gamification* en la enseñanza y el aprendizaje de la programación tuvo un impacto positivo sobre la inteligencia emocional rasgo de los estudiantes, particularmente en lo referido al aumento de los valores medidos en las facetas *Asertividad*, *Autoestima*, *Automotivación*, *Conciencia Social*, *Expresividad Emocional* y *Gestión del Estrés*, en los factores *Bienestar*, *Emocionalidad* y *Sociabilidad*, y en el factor global de *Inteligencia Emocional*. En líneas generales, esto está acorde con los resultados presentados en los estudios previos de Caton y Greenhill (2013, p. 92); Mitchell, Danino y May (2013, p. 399) y Hakulinen, Auvinen y Korhonen (2015, p. 18), sobre todo en lo que se refiere a la motivación, la gestión del estrés y la sociabilidad.

## 10.2. Conclusiones

Antes de presentar las conclusiones de esta investigación, es necesario recordar que el objetivo de esta investigación fue evaluar el impacto que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tiene sobre las siguientes dimensiones:

- la adquisición de los conocimientos específicos de la disciplina;
- las relaciones interpersonales en el aula;
- la inteligencia emocional de los estudiantes.

La consecución de este objetivo se logró a través del empleo de una metodología claramente definida —basada en los métodos mixtos (Creswell, 2014, p. 234) y sostenida en su mayor parte por instrumentos estándar, como la *Escala de la Distancia Social en el Aula* (Cunningham et al., 1951, p. 405) y el *TEIQue* (Petrides y Furnham, 2003, p. 47)— que permitió llegar a un número de conclusiones a partir de la interpretación de los resultados mostrados en el capítulo anterior, obtenidos en relación a las diferentes dimensiones del objetivo propuesto —oportunamente descritas en el marco teórico— y a la luz de los estudios previos también mencionados allí. Así, la convergencia de estas conclusiones que se presentan a continuación determinó el cumplimiento del objetivo general.

### 10.2.1. Primera conclusión: la implementación de estrategias lúdicas tuvo un efecto positivo sobre la adquisición de los conocimientos específicos de la programación orientada a objetos

El impacto que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tuvo sobre la adquisición de los conocimientos específicos de la programación orientada a objetos se pudo apreciar al detectarse una asociación estadísticamente significativa entre el *aprendizaje mediante la utilización de videojuegos* y la *adquisición de la capacidad de identificar las relaciones entre clases en UML*. Se podría concluir a partir de allí que los estudiantes que usaron juegos serios adquirieron mejor esa capacidad que aquellos que no lo hicieron. Pero eso sería cometer una falacia que los antiguos filósofos denominaban *cum hoc, ergo propter hoc* (junto con esto, por lo tanto, debido a esto), porque la existencia de una correlación no implica necesariamente que

exista una causalidad. A pesar de ello, a veces una correlación sí puede ser un indicador de una relación causal. Por eso es importante haber obtenido mediante el análisis cualitativo más evidencias de que realmente los juegos serios pueden facilitar la adquisición de ciertas competencias. En este caso, al ser el UML un lenguaje visual, el refuerzo de su aprendizaje mediante juegos visuales como son los videojuegos es una posibilidad que tiene un potencial nada despreciable, acorde con que “los videojuegos motivan a los alumnos y favorecen su rendimiento” (Revuelta Domínguez y Guerra Antequera, 2012).

Con respecto a los otros conocimientos y las otras capacidades específicos de la programación orientada a objetos que se midieron en este estudio, el análisis cuantitativo no indicó ninguna asociación entre ellos y las estrategias lúdicas utilizadas. Eso es algo que coincidió en gran medida con los resultados del análisis cualitativo, ya que —en la opinión de los estudiantes— la mayoría de los conocimientos fueron efectivamente adquiridos, independientemente de la estrategia empleada.

Se puede concluir, por lo tanto, que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tuvo cierto efecto positivo sobre la adquisición de los conocimientos específicos de la programación orientada a objetos, aunque hay que enfatizar que no se la debe considerar como una receta universal, ya que fue solamente una de las estrategias (la utilización de videojuegos serios) la que produjo resultados que se destacaron sobre los de las demás.

### **10.2.2. Segunda conclusión: la implementación de estrategias lúdicas tuvo un efecto positivo sobre las relaciones interpersonales en el aula**

El efecto que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tuvo sobre las relaciones interpersonales en el aula se pudo apreciar cuando se detectaron, entre los grupos, diferencias estadísticamente significativas en el postest de la *distancia social media del grupo*, una vez eliminada la influencia del pretest —o sea que esas diferencias en el postest no se podrían explicar en base a los valores del pretest—. Esta vez, la estrategia que se destacó fue el *aprendizaje mediante el desarrollo de videojuegos*, ya que los estudiantes que programaron juegos serios tuvieron en sus postests —en promedio— menores valores de este índice

(la *distancia social media del grupo*) que los estudiantes de los otros cursos, o sea que tenían, al momento de finalizar la experiencia, una mayor aceptación por parte de los demás integrantes de su grupo.

Pero las diferencias estadísticamente significativas no se restringieron a esta última estrategia. Por el contrario, para el índice que acabamos de mencionar, la diferencia entre los valores del *postest* y el *pretest* fue estadísticamente significativa en todos los cursos, incluso en el grupo de control, y para la *distancia social media propia* la disminución fue estadísticamente significativa en tres de los cuatro grupos —solamente no lo fue en el grupo que participó de la experiencia de ludificación—. Podría suponerse que estos índices cayeron naturalmente en todos los grupos por el solo hecho de que los estudiantes convivieron a lo largo de cierto intervalo de tiempo, lo que los aproximó. No obstante, el hecho de que el *aprendizaje mediante el desarrollo de videojuegos* se haya destacado probablemente se deba más a que esta estrategia obligó a que los alumnos trabajaran en equipos, promoviendo la comunicación y los intercambios, algo que —según Q. Burke y Kafai (2014, p. 690)— también ocurre en muchas comunidades *online* de desarrolladores de juegos. Por ello, no son sorprendentes estos resultados del análisis cuantitativo, los que —además— encuentran su confirmación en los comentarios positivos de los estudiantes que desarrollaron videojuegos serios y participaron de las entrevistas de la fase cualitativa.

En consecuencia, se puede concluir que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tuvo algún efecto positivo sobre las relaciones interpersonales en el aula, pues entre los cursos se midieron diferencias estadísticamente significativas en los valores promedio de la *distancia social media del grupo*, destacándose el *aprendizaje mediante el desarrollo de videojuegos*, y en casi todos los grupos se midieron también diferencias significativas entre los *postests* y los *pretests* de ambos índices sociométricos.

### **10.2.3. Tercera conclusión: la implementación de estrategias lúdicas tuvo un efecto positivo sobre la inteligencia emocional de los estudiantes**

El impacto que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tuvo sobre la inteligencia emocional de los estudiantes se

pudo apreciar al detectarse, entre los grupos, diferencias estadísticamente significativas en el postest de la faceta *Conciencia Social*, una vez eliminada la influencia del pretest. También esta vez, la estrategia que se destacó fue el *aprendizaje mediante el desarrollo de videojuegos*, ya que los estudiantes que programaron juegos serios tuvieron en sus postests —en promedio— valores más altos de esta faceta que los estudiantes de los otros cursos, o sea que se percibían en mayor medida, al momento de finalizar la experiencia, como tejedores de redes con habilidades sociales superiores. Este resultado no sorprende, debido a que —como ya se vio— en este curso también se observó la mayor disminución en la distancia social media del grupo.

No obstante, las diferencias estadísticamente significativas no se restringieron a esta última estrategia. Por el contrario, para el índice que acabamos de mencionar, la diferencia entre los valores del protest y el pretest fue estadísticamente significativa en el curso mencionado y también en el grupo que participó de la experiencia de ludificación. En este último grupo, además, se observó la mayor cantidad de aumentos estadísticamente significativos entre el pretest y el postest: en las facetas *Asertividad*, *Autoestima*, *Automotivación*, *Conciencia Social*, *Expresividad Emocional* y *Gestión del Estrés*, en los factores *Bienestar*, *Emocionalidad* y *Sociabilidad*, y en el factor global de *Inteligencia Emocional*. El hecho de que haya sido precisamente en el grupo que participó de la *ludificación del aprendizaje/Gamification* donde se observaron tantas variaciones se explica, en parte, porque esta estrategia obligó a que los alumnos trabajaran en equipos que competían entre sí, en una carrera contra el tiempo. Por ello, no son sorprendentes estos resultados del análisis cuantitativo, los que —además— encuentran su confirmación en los comentarios positivos de los estudiantes que participaron de la experiencia y se ofrecieron para las entrevistas llevadas a cabo para la fase cualitativa y en la siguiente afirmación de B. Burke (2014): “la ludificación se puede usar para involucrar y motivar a las personas a realizar cambios de comportamiento . . . positivos en sus vidas”.

Por lo tanto, se puede concluir que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tuvo cierto efecto positivo sobre la inteligencia emocional de los estudiantes, aunque hay que enfatizar que no se la debe considerar como una panacea, ya que en una de las estrategias (el *aprendizaje mediante la*

*utilización de videojuegos*) también se observó un resultado negativo (la reducción entre la media del pretest y la media del posttest en la faceta *Control de la Impulsividad*).

#### **10.2.4. Conclusión general del estudio**

La conclusión general de este estudio —alcanzada a partir de las conclusiones que se acaban de presentar— es que la implementación de estrategias lúdicas en la enseñanza y el aprendizaje de la programación tuvo —a grandes rasgos— un impacto positivo sobre la adquisición de los conocimientos específicos de la disciplina, las relaciones interpersonales en el aula y la inteligencia emocional de los estudiantes.

### **10.3. Principales aportaciones**

Esta investigación pretende ser un aporte sobre la aplicación de métodos de enseñanza innovadores y sobre el estudio de los mismos, para satisfacer las necesidades de educación actuales.

En primer lugar, los resultados de este trabajo proporcionan evidencias sobre la viabilidad de aplicar eficazmente tres estrategias lúdicas (*Aprendizaje mediante el desarrollo de videojuegos*, *Aprendizaje mediante la utilización de videojuegos* y *Ludificación del aprendizaje/Gamification*) en la Educación Superior.

En segundo lugar —y por tratarse de un análisis comparativo—, esta investigación puede ayudar a comprender cómo las estrategias lúdicas estudiadas influyen en los procesos de aprendizaje, las competencias que se desarrollan y el rendimiento académico, impactando sobre la motivación y el compromiso de los estudiantes.

En tercer lugar, esta investigación puede servir de guía sobre estrategias lúdicas para instituciones y profesores de *programación* interesados en su aplicación. En particular, para el lector interesado, el extenso marco teórico elaborado puede servir como una introducción a los temas de *la programación y su enseñanza*, *las relaciones interpersonales en las aulas*, *la inteligencia emocional*, *el juego y la ludología.*, así como también a las tres estrategias lúdicas estudiadas, que fueron presentadas y explicadas con suficiente grado de detalle como para establecer un punto de partida sólido hacia estudios posteriores.

En cuarto lugar, los videojuegos desarrollados y utilizados durante la investigación pueden ser fuente de inspiración sobre las formas de integrar estas estrategias innovadoras en la enseñanza.

En quinto lugar, los resultados de este estudio tienen implicaciones para la investigación y la práctica futuras. Se propuso aquí una metodología y unos instrumentos que pueden reutilizarse o adaptarse para nuevas investigaciones.

En resumen, este estudio ha demostrado empíricamente el potencial de las estrategias lúdicas como una alternativa para la mejora de la enseñanza de la programación en la Educación Superior.

Una vez finalizada esta investigación —pero, a menudo, también durante su desarrollo—, han quedado en evidencia varias de sus limitaciones. Reflexionar sobre ellas es importante para así poder conocer su alcance y tenerlas en consideración en el futuro.

#### **10.4. Limitaciones del estudio**

En primer lugar, las características de la población estudiada limitan la capacidad de generalizar los hallazgos del estudio. Debido a que este se centró en los estudiantes de la materia Programación II del segundo año de la Tecnicatura Superior en Informática Aplicada, carrera que se estudia en el Instituto Nacional Superior del Profesorado Técnico (Universidad Tecnológica Nacional), una institución de Educación Superior localizada en la Ciudad Autónoma de Buenos Aires, República Argentina, las conclusiones de esta investigación solo pueden aplicarse a contextos que tengan una demografía comparable a la de esta institución.

En segundo lugar, las características de las muestras también son un factor limitante. Para aumentar la validez y la confiabilidad del estudio, idealmente las muestras se deberían haber elegido al azar. Sin embargo, eso no fue posible. En este estudio, los participantes de la fase cuantitativa eran estudiantes de cursos preestablecidos y los de la fase cualitativa participaron de manera voluntaria. Especialmente el análisis de datos cualitativos se basó en gran medida en las transcripciones de las entrevistas de 21 participantes, que podrían no representar



bien las variedades de todos los estudiantes. Además, algunos estudiantes que no completaron el curso y se inscribieron de nuevo al año siguiente debieron ser descartados, lo que llevó a que las muestras tuvieran diferentes tamaños. Y a pesar de haber logrado, en total, la participación de casi una centena de estudiantes, estos constituyeron cuatro grupos relativamente pequeños, dejando poco margen para la depuración de los datos. Por ello, la potencia estadística alcanzada no es demasiado alta y, en consecuencia, las conclusiones del estudio son propias de esta experiencia puntual y no pueden generalizarse. Para que esto fuera posible, se debería haber contado con muestras de mayor tamaño, representativas de un conjunto de estudiantes e instituciones educativas más amplio.

En tercer lugar, también las características de los tratamientos limitan la capacidad de generalizar los hallazgos de la investigación. El grupo que desarrolló videojuegos lo hizo utilizando el lenguaje de programación Java y el *framework* LibGDX. Las conclusiones basadas en ese tratamiento solo serían aplicables a poblaciones donde los estudiantes usen esas mismas herramientas u otras similares. A su vez, el grupo que jugó con los videojuegos serios utilizó los productos desarrollados por sus antecesores, es decir, juegos que no fueron desarrollados por profesionales sino por estudiantes. Estos juegos, por lo general, resultaron ser sorprendentemente parecidos a muchos de los juegos de la primera generación del *software* educativo: no enseñaban, sino que evaluaban los conocimientos previos de los estudiantes. También la ludificación de la materia Programación II en 2018 fue bastante limitada, ya que se basó en el patrón PBL (*Points, Badges, Leaderboards*) que es una de las formas más básicas de esta estrategia.

En cuarto lugar, las características de los instrumentos son otro factor limitante. Habría sido preferible que la investigación se enfocara en el desarrollo de la *competencia emocional*, pero la adopción temprana del TEIQue —cuando el marco teórico aún estaba en construcción y, consecuentemente, los conceptos del trabajo todavía no se encontraban claramente definidos— obligó a que el estudio, casi desde el inicio, se orientara a evaluar el impacto de la implementación de estrategias lúdicas sobre la *inteligencia emocional*, ya que este cuestionario de autoinforme estándar de

fácil aplicación no mide la competencia emocional sino la inteligencia emocional rasgo. A su vez, la prueba objetiva de programación orientada a objetos en Java —como todas las pruebas objetivas— también tiene sus limitaciones: puede medir ciertos conocimientos, pero no la capacidad de crear programas —para lo cual se requeriría implementar una evaluación basada en el desempeño, como son, por ejemplo, los trabajos para hacer en casa o los miniproyectos de laboratorio. Por último, cabe mencionar que la principal limitación del test sociométrico utilizado en este estudio es su antigüedad. Existen actualmente muchas otras alternativas más modernas —como, por ejemplo, el ARS (*Análisis de Redes Sociales*)— que podrían haberse utilizado en su lugar.

Por último, el reparto del tiempo es otra limitación de este estudio. Idealmente, el grupo de control y los grupos correspondientes a los tratamientos deberían haber sido estudiados simultáneamente, pero esto no fue posible. Por ello, ciertos factores podrían amenazar la validez del diseño metodológico adoptado, generando efectos que podrían confundirse con los de los tratamientos. Por ejemplo, muchos eventos desconocidos ocurridos entre el pretest y el posttest —la *historia*— podrían haber afectado los resultados medidos en este último. La influencia de la historia y de otros factores (Campbell y Stanley, 1973, p. 17), como por ejemplo la *maduración* —procesos internos de los participantes que se dan naturalmente con el pasar del tiempo— y la *administración de tests* —la influencia del pretest sobre la medición del posttest—, se trató de reducir mediante la inclusión y evaluación del grupo de control, ya que los potenciales efectos también se deberían manifestar en este último. El hecho de no haber podido poner, entre el pretest y el posttest, a los estudiantes en cuarentena —como en un verdadero experimento— es una limitación del diseño cuasiexperimental empleado.

## 10.5. Prospectiva

A pesar de las limitaciones mencionadas, consideramos que este estudio ha logrado exponer —a rasgos generales— cómo la implementación de tres diferentes estrategias lúdicas en la enseñanza y el aprendizaje de la programación impactó sobre la adquisición

de los conocimientos específicos de esta disciplina, las relaciones interpersonales en el aula y la inteligencia emocional de los estudiantes, demostrando empíricamente el potencial de estas estrategias como una alternativa para la mejora de la enseñanza de la programación en la Educación Superior. Es por ello que no se descarta la realización de futuros estudios para perfeccionar los resultados obtenidos.

En primer lugar, sería interesante replicar el estudio de modo horizontal en diferentes instituciones de la Educación Superior, para expandir la muestra y aumentar así la potencia de las pruebas estadísticas, consolidando los resultados y posibilitando la generalización de las conclusiones.

En segundo lugar, podría analizarse si otras variables —como, por ejemplo, la edad, el sexo y la situación laboral de los estudiantes— influyen sobre la eficacia de las estrategias lúdicas aplicadas. En esta investigación, esas variables solo fueron utilizadas para describir las muestras —estadística descriptiva— pero no para formular y contrastar hipótesis —estadística inferencial—.

En tercer lugar, los tratamientos podrían ampliarse para abarcar el desarrollo de videojuegos con otras de las herramientas de programación mencionadas en el capítulo 6, como las API, las bibliotecas y los motores de juego —además de los *frameworks*—, y se podría enseñar un poco más de diseño de videojuegos serios —incluyendo cuestiones narratológicas, por ejemplo—, para aumentar el valor pedagógico de los juegos creados por los estudiantes.

En cuarto lugar, los instrumentos utilizados podrían perfeccionarse. El TEIQue podría cambiarse por un instrumento que permita medir la competencia emocional en lugar de la inteligencia emocional rasgo, el test sociométrico podría reemplazarse por el ARS (*Análisis de Redes Sociales*) y las pruebas objetivas de Java podrían ser sustituidas por miniproyectos de laboratorio, para medir adecuadamente la capacidad de desarrollar software, en lugar de los conocimientos sobre los lenguajes de programación. Además, podrían incorporarse otros instrumentos y técnicas utilizados frecuentemente en la investigación educativa, como los estudios de caso y las observaciones, por ejemplo.

Por último, podría mejorarse el reparto del tiempo para que el grupo de control y los grupos correspondientes a los tratamientos sean estudiados simultáneamente, evitando así que ciertos factores como la historia y la maduración generen efectos que podrían confundirse con los de los tratamientos.

Desde esta investigación hemos querido contribuir a la mejora de la enseñanza de la programación en la Educación Superior, analizando la eficacia de tres estrategias lúdicas como posibles alternativas pedagógicas. Varios resultados interesantes han surgido de nuestro análisis, pero también se ha abierto un gran espacio para nuevas investigaciones. ¡Esto recién empieza!

# REFERENCIAS

---



- Aarseth, E., Smedstad, S. M. y Sunnanå, L. (2003). A multi-dimensional typology of games. En *What's in a game? - Game taxonomies, typologies and frameworks* (pp. 48-53). University of Utrecht.
- Abelson, H., Sussman, G. J. y Sussman, J. (1984). *Structure and interpretation of computer programs*. Cambridge, USA: MIT Press.
- About CSEdWeek. (s. f.). Recuperado de: <http://csedweek.org/about>
- About EU Code Week. (s. f.). Recuperado de: <http://codeweek.eu/about>
- About the Serious Games Initiative. (2017, agosto 15). Recuperado de: <http://www.wilsoncenter.org/about-the-serious-games-initiative>
- About Us | Games For Change. (s. f.). Recuperado de: <http://www.gamesforchange.org/who-we-are/about-us>
- About Us | Ozobot. (s. f.). Recuperado de: <http://ozobot.com/about-us>
- Abt, C. C. (1987). *Serious Games*. Lanham: University Press of America.
- Accredible BadgeBuilder. (s. f.). Recuperado de: <http://badge.design>
- Agapito, J. L., Martinez, J. C. y Casano, J. D. (2014). Xiphias: A Competitive Classroom Control System to Facilitate the Gamification of Academic Evaluation of Novice C++ Programmers. En *Proceedings of the International Symposium on Computing for Education (ISCE 2014)* (pp. 9-15). Boracay: Philippine Society of Information Technology Educators.
- Agresti, A. y Finlay, B. (2009). *Statistical Methods for the Social Sciences*. Pearson Prentice Hall.
- Aguirre, J. (2009). Panorama de la historia de la Computación Académica en la Argentina. En J. Aguirre y R. Carnota (Eds.), *Historia de la informática en*

- Latinoamérica y el Caribe: investigaciones y testimonios* (pp. 21-42). Río Cuarto: Universidad Nacional de Río Cuarto.
- Anderson, J. R. y Reiser, B. J. (1985). The LISP Tutor: it approaches the effectiveness of a human tutor. *BYTE*, 10(4), 159–175.
- Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., ... Wittrock, M. C. (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's taxonomy*. (L. W. Anderson y D. R. Krathwohl, Eds.). New York: Longman.
- Anderson, R. y Cerro, D. (2012). *Pro Arduino*. New York: Apress.
- AntMe! (s. f.). Recuperado de: <http://www.antme.net/en>
- Arruga i Valeri, A. (1974). *Introducción al test sociométrico*. Barcelona: Herder.
- Aspray, W. (2016). *Participation in Computing*. Cham: Springer International Publishing.
- Aumont, J. (2001). *La estética hoy*. (M. A. Galmarini, Trad.). Madrid: Ediciones Cátedra.
- Babini, N. y Vilensky, C. (2003). *La Argentina y la computadora: crónica de una frustración*. Buenos Aires: Dunken.
- Badger, M. (2009). *Scratch 1.4: Beginners Guide. Learn to program while creating interactive stories, games, and multimedia projects using Scratch*. Birmingham: Packt Publishing.
- Baker, A., Navarro, E. O. y Van Der Hoek, A. (2005). An experimental card game for teaching software engineering processes. *Journal of Systems and Software*, 75(1), 3-16.
- Balluerka Lasa, N. y Vergara Iraeta, A. I. (2002). *Diseños de investigación experimental en psicología: modelos y análisis de datos mediante el SPSS 10.0*. Pearson Educación.
- Banerji, N. y Tate, L. (2013). *Test Review: Trait Emotional Intelligence Questionnaire (TEIQue)*. Londres: The British Psychological Society.
- Bar-On, R. (1997). *The Emotional Quotient Inventory (EQ-i): A Test of Emotional Intelligence*. Toronto: Multi-Health Systems, Inc.
- Bar-On, R. (2006). The Bar-On model of emotional-social intelligence (ESI). *Psicothema*, 18, 13–25.



- Bartha, P. (2016). Analogy and Analogical Reasoning. En E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2016). Metaphysics Research Lab, Stanford University. Recuperado de: <http://plato.stanford.edu/archives/win2016/entries/reasoning-analogy>
- Bartle, R. (1990). Who plays MUAs? *Comms Plus!*, 18–19.
- Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. *Journal of MUD research*, 1(1), 19.
- Bastin, G. (1966). *Los Tests Sociométricos*. Buenos Aires: Editorial Kapelusz.
- Bathke, A. y Brunner, E. (2003). A nonparametric alternative to analysis of covariance. En M. G. Akritas y D. N. Politis (Eds.), *Recent advances and trends in nonparametric statistics* (pp. 109–120). Elsevier.
- Bauer, A., Butler, E. y Popović, Z. (2017). Dragon architect: open design problems for guided learning in a creative computational thinking sandbox game. En *Proceedings of FDG'17*. Hyannis, EE.UU.
- Bellucci, M. (2017, septiembre 7). Faltan programadores, cada año quedan 5 mil puestos vacantes y buscan hasta chicos. *Clarín*. Buenos Aires. Recuperado de: [http://www.clarin.com/sociedad/falta-programadores-quedan-mil-puestos-vacantes-pais\\_0\\_B1FHsMIB-.html](http://www.clarin.com/sociedad/falta-programadores-quedan-mil-puestos-vacantes-pais_0_B1FHsMIB-.html)
- Berners-Lee, T. y Fischetti, M. (2000). *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. New York: Harper Business.
- Bisquerra Alzina, R. (2009). *Psicopedagogía de las emociones*. Madrid: Ed. Síntesis.
- Bisquerra Alzina, R. y Pérez Escoda, N. (2007). Las competencias emocionales. Recuperado de: <http://e-spacio.uned.es/fez/view.php?pid=bibliuned: EducacionXXI-2007numero10-823>
- Bisson, G. (1986, octubre). Atari Still Makes the Grade. *Antic*, 5(6), 39-52.
- Blas, M. J., Hauque, F., Re, S. y Castellaro, M. (2017). A support tool designed as didactic material for teaching and learning programming. En *Actas de las 46 JAIIO - XLIII CLEI* (pp. 1–10). Córdoba: IEEE.

- Blohm, I. y Leimeister, J. M. (2013). Gamification: Design of IT-based enhancing services for motivational support and behavioral change. *Business & Information Systems Engineering (BISE)*, 5(4), 275–278.
- Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H. y Krathwohl, D. R. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook 1: Cognitive domain*. (B. S. Bloom, Ed.). New York: David McKay.
- Bogost, I. (2011, agosto 9). «Gamification Is Bullshit». *The Atlantic*. Recuperado de: <http://www.theatlantic.com/technology/archive/2011/08/gamification-is-bullshit/243338>
- Bogost, I. (2015). Why gamification is bullshit. En S. P. Walz y S. Deterding (Eds.), *The gameful world: Approaches, issues, applications* (pp. 65-80). Cambridge: The MIT Press.
- Borchers, D. (2001, octubre 6). Vor 50 Jahren fing alles an: das erste «Elektronenhirn» in Deutschland. *heise online*. Recuperado de: <http://www.heise.de/newsticker/meldung/Vor-50-Jahren-fing-alles-an-das-erste-Elektronenhirn-in-Deutschland-51722.html>
- Botero Tabares, R. de J. y Trefftz Gómez, H. (2011). Medra para el aprendizaje en lógica de programación orientada a objetos mediante un juego. En *II Encuentro Internacional y VI Nacional de Investigación en Ingeniería de Sistemas e Informática – EIISI 2011*. UPTC, Tunja, Colombia.
- Bowen, B. D. (1997, enero 1). Educators embrace Java. *JavaWorld*. Recuperado de: <http://www.javaworld.com/article/2076867/educators-embrace-java.html>
- Burguera, A. y Guerrero, J. (2014). Lenguaje ensamblador en el siglo XXI: Desarrollo de videojuegos como elemento motivador. *ReVisión*, 7(2).
- Burke, B. (2014). *Gamify: How gamification motivates people to do extraordinary things*. Brookline: Bibliomotion, Inc.
- Burke, Q. y Kafai, Y. B. (2014). Decade of Game-Making for Learning: From Tools to Communities. En M. C. Angelides y H. Agius (Eds.), *Handbook of Digital Games* (pp. 689-709). Hoboken, USA: John Wiley & Sons, Inc.

- Caillois, R. (1986). *Los juegos y los hombres: la máscara y el vértigo*. (J. Ferreiro, Trad.). México: Fondo de Cultura Económica. (Obra original publicada en 1958).
- Calero, A. (2013). Versión argentina del TMMS para adolescentes: una medida de la inteligencia emocional percibida. *Cuadernos de Neuropsicología/Panamerican Journal of Neuropsychology*, 7(1), 104-119.
- Camilo Hernandez, C., Silva, L., Alencar Segura, R., Schimiguel, J., Fernández Paradela Ledón, M., Mendes Bezerra, L. N. y Frango Silveira, I. (2010). Teaching programming principles through a game engine. *CLEI electronic journal*, 13(2). Recuperado de: <http://www2.clei.org/cleiej/papers/v13i2p3.pdf>
- Campaña, J. R., Marín, A. E., Ros, M., Sánchez, D., Medina, J. M., Vila, M. A., ... Martin-Bautista, M. J. (2016). Metodologías activas y gamificación en las asignaturas de iniciación a la programación. En M. Torres Gil y J. J. Cañadas Martínez (Eds.), *Actas de las XXII Jornadas sobre la Enseñanza Universitaria de la Informática* (pp. 245-252). Almería: Editorial Universidad de Almería.
- Campbell, D. T. y Stanley, J. C. (1973). *Diseños experimentales y cuasiexperimentales en la investigación social*. Buenos Aires: Amorrortu Editores.
- Cañeque, H., Castro, C. y Greco, H. (1999). El juego es vida. En *El juego. Debates y aportes desde la didáctica* (pp. 30-45). Buenos Aires: Ediciones Novedades Educativas.
- Carnota, R. y Borches, C. (2015). Sobre personajes, instituciones y palabras. La Sociedad Argentina de Cálculo en su primera etapa (1960-1962). En L. G. Rodríguez Leal y R. Carnota (Eds.), *Historias de las TIC en América Latina y el Caribe: inicios, desarrollos y rupturas*. Madrid: Fundación Telefónica.
- Carnota, R., Factorovich, P. y Pérez, M. O. (2009). IBM Go Home!... Conflictos políticos y académicos y perfiles profesionales en los primeros años de la carrera de Computación Científica de la FCEyN-UBA (1963-1971). En J. Aguirre y R. Carnota (Eds.), *Historia de la informática en Latinoamérica y el Caribe: investigaciones y testimonios* (pp. 147-166). Río Cuarto: Universidad Nacional de Río Cuarto.

- Carnota, R. y Pérez, M. O. (2009). Continuidad formal y ruptura real: la segunda vida de Clementina. En J. Aguirre y R. Carnota (Eds.), *Historia de la informática en Latinoamérica y el Caribe: investigaciones y testimonios* (pp. 125-146). Río Cuarto: Universidad Nacional de Río Cuarto.
- Castaño Collado, C. (2008). Nuevas tecnologías y género. La segunda brecha digital y las mujeres. *TELOS*, 75, 24-33.
- Caton, H. y Greenhill, D. (2013). The effects of gamification on student attendance and team performance in a third-year undergraduate game production module. En *European conference on games based learning* (p. 88). Academic Conferences International Limited.
- Catrobat. (s. f.). Recuperado de: <http://developer.catrobat.org>
- CeeBot: Have fun programming. (s. f.). Recuperado de: <http://www.ceebot.com/ceebot/index-e.php>
- Cellan-Jones, R. (2011, noviembre 28). Coding - the new Latin. *BBC News*. Recuperado de: <http://www.bbc.com/news/technology-15916677>
- Clark, W. (1986). The LINC Was Early and Small. En *Proceedings of the ACM Conference on The History of Personal Workstations* (pp. 133-155). New York, NY, USA: ACM. Recuperado de: <http://www.digibarn.com/stories/linc/documents/LINC-Personal-Workstation/LINC-Personal-Workstation.pdf>
- Classcraft. (s. f.). Recuperado de: <http://www.classcraft.com/es>
- ClassDojo. (s. f.). Recuperado de: <http://www.classdojo.com/es-ES>
- Cocos2d. (s. f.). Recuperado de: <http://cocos2d.org>
- Code Warriors: Hakitzu Battles. (s. f.). Recuperado de: <http://www.codewarriorsgame.com>
- Codecademy - learn to code, interactively, for free. (s. f.). Recuperado de: <http://www.codecademy.com>
- Code.org 2017 Annual Report. (s. f.). Recuperado de: <http://code.org/about/2017>
- Cohn, D. (2005, septiembre 20). C'mon Kids, Let's Play Programmer. *WIRED*. Recuperado de: <http://www.wired.com/2005/09/cmon-kids-lets-play-programmer>

- Colmerauer, A. y Roussel, P. (1996). The birth of Prolog. En R. G. Gibson y T. J. Bergin (Eds.), *History of programming languages II* (pp. 331-367). New York: ACM Press.
- Compaine, B. M. (Ed.). (2001). *The digital divide: facing a crisis or creating a myth?* Cambridge, Mass: MIT Press.
- Conover, W. J. y Iman, R. L. (1982). Analysis of Covariance Using the Rank Transformation. *Biometrics*, 38(3), 715.
- Consortium for Research on Emotional Intelligence in Organizations. (s. f.). Recuperado de: <http://www.eiconsortium.org>
- Coonradt, C. A. (1984). *The Game of Work: How to enjoy Work as much as Play*. Orem, Utah: Liberty Press.
- Corbató, F. J., Merwin-Daggett, M. y Daley, R. C. (1962). An experimental time-sharing system. En *Proceedings of the Spring Joint Computer Conference* (pp. 335-344). ACM. Recuperado de: <http://larch-www.lcs.mit.edu:8001/~corbato/sjcc62>
- Creswell, J. W. (2014). *Research design: qualitative, quantitative, and mixed methods approaches* (4ª ed.). Thousand Oaks: SAGE Publications.
- Csikszentmihályi, M. (1975). *Beyond Boredom and Anxiety*. San Francisco: Jossey-Bass Publishers.
- Csikszentmihályi, M. (1982). Does being human matter? On some interpretive problems of comparative ludology. *The Behavioral and Brain Sciences*, 5, 160.
- Csikszentmihályi, M. (1990). *Flow: The Psychology of Optimal Experience*. New York: Harper & Row.
- Cunningham, R., Elzi, A., Hall, J. A., Farrell, M. y Roberts, M. (1951). *Understanding group behavior of boys and girls*. New York: Bureau of Publications, Teachers College, Columbia University.
- Cvijikj, I. P. y Michahelles, F. (2011). The toolkit approach for end-user participation in the internet of things. En D. Uckelmann, M. Harrison y F. Michahelles (Eds.), *Architecting the Internet of Things* (pp. 65-96). Berlin: Springer.
- Dagiene, V., Stupuriene, G., Vinikiene, L. y Zakauskas, R. (2017). Introduction to Bebras Challenge Management: Overview and Analyses of Developed Systems. En

- International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 232–243). Springer.
- Dann, W., Cooper, S. C. y Pausch, R. (2004). *Learning to Program with Alice*. Pearson Prentice Hall.
- Dawson, M. (2011). *Beginning C++ through game programming* (3ª ed.). Boston: Course Technology.
- de Halleux, J. P. (2017, octubre 23). MakeCode for Minecraft makes learning to code super fun. Recuperado de: <http://www.microsoft.com/en-us/research/blog/code-minecraft>
- Decreto 459/10 - Crea Programa «Conectar Igualdad.Com.Ar» de incorporación de la nueva tecnología para el aprendizaje de alumnos y docentes (2010). Recuperado de: <http://www.conectarigualdad.gob.ar/archivos/archivoSeccion/DecretoCreaci%C3%B3nCI.pdf>
- Delgado de la Torre, R. (2004). *Iniciación a la probabilidad y la estadística*. Bellaterra: Universitat Autònoma de Barcelona. Servei de Publicacions.
- Deterding, S., Dixon, D., Khaled, R. y Nacke, L. (2011). From game design elements to gamefulness: defining gamification. En *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments* (pp. 9–15). Tampere, Finlandia: ACM.
- DeVries, R. y Kamii, C. (1975). *Why Group Games? A Piagetian Perspective*. Urbana, Illinois: College of Education, University of Illinois.
- Díaz Matarranz, J. J. y González Urbano, E. (2011). La competencia emocional y su evaluación en las ciencias sociales. En P. Mirallés Martínez, S. Molina Puché y A. Santisteban Fernández (Eds.), *La evaluación en el proceso de enseñanza y aprendizaje de las ciencias sociales* (Vol. I, pp. 193–202). Murcia: Asociación Universitaria de Profesorado de Didáctica de las Ciencias Sociales.
- Dicheva, D., Dichev, C., Agre, G. y Angelova, G. (2015). Gamification in Education: A Systematic Mapping Study. *Educational Technology & Society*, 18(3), 75–88.

- Dijkstra, E. W. (1968). Letters to the editor: go to statement considered harmful. *Communications of the ACM*, 11(3), 147–148.
- Dijkstra, E. W. (1982). EWD 498: How do we tell truths that might hurt? (18/06/1975). En D. Gries (Ed.), *Selected Writings on Computing: A personal Perspective*. New York: Springer-Verlag.
- Dillon, R. (2011). *The golden age of video games: The birth of a multibillion dollar industry*. Boca Raton: CRC Press.
- Direct3D. (s. f.). Recuperado de: <http://docs.microsoft.com/en-us/windows/desktop/direct3d>
- DirectX SDK. (s. f.). Recuperado de: <http://docs.microsoft.com/en-us/windows/desktop/directx-sdk--august-2009->
- diSessa, A. A. (1987). The third revolution in computers and education. *Journal of Research in Science Teaching*, 24(4), 343–367.
- diSessa, A. A. (2001). *Changing Minds: Computers, Learning, and Literacy*. Cambridge, EE. UU.: The MIT Press.
- diSessa, A. A. y Abelson, H. (1986). Boxer: A reconstructible computational medium. *Communications of the ACM*, 29(9), 859–868.
- Djaouti, D., Alvarez, J. y Jessel, J.-P. (2011). Classifying serious games: the G/P/S model. En P. Felicia (Ed.), *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches* (pp. 118–136). Hershey: IGI Global.
- Doménech Betoret, F. (2011). *Evaluar e investigar en la situación educativa universitaria. Un nuevo enfoque desde el EEES*. Castellón de la Plana: Publicacions de la Universitat Jaume I.
- Du Boulay, B. (1986). Some Difficulties of Learning to Program. *Journal of Educational Computing Research*, 2(1), 57-73.
- Dugard, P. y Todman, J. (1995). Analysis of Pre-test-Post-test Control Group Designs in Educational Research. *Educational Psychology*, 15(2), 181-198.

- Durán, W. O., Zoltan, C., Lew, L. S., Cortés, C. D. y García, N. S. (2009). COMIC: El primer lenguaje y compilador argentino, desarrollado en el Instituto de Cálculo en 1965. En J. Aguirre y R. Carnota (Eds.), *Historia de la informática en Latinoamérica y el Caribe: investigaciones y testimonios* (pp. 109-124). Río Cuarto: Universidad Nacional de Río Cuarto.
- Earp, J., Dagnino, F., Kiili, K., Kiili, C., Tuomi, P. y Whitton, N. (2013). Learner Collaboration in Digital Game Making: An Emerging Trend. En D. Parmigiani, V. Pennazio y A. Traverso (Eds.), *Learning and Teaching with Media and Technology. ATEE-SIREM Winter Conference Proceedings* (pp. 439-447). Génova, Italia.
- Educación. (s. f.). Recuperado de: <http://juegosargentinos.org/educacion>
- Egenfeldt-Nielsen, S., Smith, J. H. y Tosca, S. P. (2008). *Understanding video games: the essential introduction*. New York: Routledge.
- else Heart.Break(). (s. f.). Recuperado de: <http://www.elseheartbreak.com>
- Ensmenger, N. (2010). *The computer boys take over: computers, programmers, and the politics of technical expertise*. Cambridge: MIT Press.
- Esnaola Horacek, G. A. (2004). *La construcción de la identidad social a través de los videojuegos: un estudio del aprendizaje en el contexto institucional de la escuela* (Tesis doctoral). Universidad de Valencia, Valencia, España.
- Esnaola Horacek, G. A. (2014). Videojuegos para aprender... ¡también en la escuela! *Voces en el Fénix*, 5(40), 126-133.
- Esnaola Horacek, G. A. y Galli, M. G. (2016, julio). Juegos, juguetes y videojuegos. El valor formativo de la actividad lúdica. *Para Juanito. Revista de Educación Popular y Pedagogías Críticas*, 4(9), 17-24.
- Esnaola Horacek, G. A., Iparraguirre, A., Averbuj, G. y Oulton, M. L. (2015). Argentina. En M. J. P. Wolf (Ed.), *Video Games Around the World* (pp. 35-56). Cambridge: MIT Press.



- European Commission - Education, Audiovisual & Culture Executive Agency. (2014). *MAGICAL - MAking Games In CollaborAtion for Learning* (Final Report - Public Part).
- Evans, N. (1986). *The future of the microcomputer in schools*. Basingstoke: Macmillan Education.
- Extremera Pacheco, N. y Fernández-Berrocal, P. (2002). *Cuestionario MSCEIT (Versión Española 2.0) de Mayer, Salovey and Caruso Emotional Intelligence Test*. Toronto: Multi-Health Systems Publishers.
- Extremera Pacheco, N. y Fernández-Berrocal, P. (2004). La inteligencia emocional: métodos de evaluación en el aula. *Revista Iberoamericana de Educación*, 34(1), 1-12.
- Factorovich, P. y Jacovkis, P. M. (2009). La elección de la primera computadora universitaria en Argentina. En J. Aguirre y R. Carnota (Eds.), *Historia de la informática en Latinoamérica y el Caribe: investigaciones y testimonios* (pp. 83-98). Río Cuarto: Universidad Nacional de Río Cuarto.
- Feldgen, M. y Clúa, O. (2004). Games as a motivation for freshman students learn programming. En *Frontiers in Education, 2004* (pp. 11-16). IEEE.
- Fernández Prados, J. S. (2000). *Sociología de los grupos escolares: sociometría y dinámica de grupos*. Universidad de Almería, Servicio de Publicaciones.
- Fernández-Berrocal, P. y Extremera Pacheco, N. (2005). La Inteligencia Emocional y la educación de las emociones desde el Modelo de Mayer y Salovey. *Revista Interuniversitaria de Formación del Profesorado*, 19(3), 63-93.
- Fernández-Berrocal, P., Extremera Pacheco, N. y Ramos, N. (2004). Validity and reliability of the Spanish modified version of the Trait Meta-Mood Scale. *Psychological reports*, 94(3), 751-755.
- Feurzeig, W. (2011). Educational Technology at BBN. En D. Walden y R. Nickerson (Eds.), *A Culture of Innovation: Insider Accounts of Computing and Life at BBN* (pp. 281-351). East Sandwich, EE.UU.: Waterside Publishing. Recuperado de: <http://www.walden-family.com/bbn/bbn-print2.pdf>

- Feurzeig, W., Papert, S., Bloom, M., Grant, R. y Solomon, C. (1969). *Programming-Languages as a Conceptual Framework for Teaching Mathematics. Final Report on the First Fifteen Months of the LOGO Project.* (No. 1889). Cambridge, EE. UU.: Bolt, Beranek and Newman, Inc. Recuperado de: <http://eric.ed.gov/?id=ED038034>
- Field, A. P. (2009). *Discovering statistics using SPSS (and sex, drugs and rock «n» roll).* Los Angeles: SAGE Publications.
- Fontela, C. (2003). *Programación Orientada a Objetos. Técnicas Avanzadas de Programación.* Buenos Aires: Nueva Librería.
- Foote, S. (2015). *Learning to program.* Upper Saddle River: Addison-Wesley.
- Frasca, G. (1999). Ludology meets narratology: Similitude and differences between (video) games and narrative. *Parnasso*, 3. Recuperado de: <http://www.ludology.org/articles/ludology.htm>
- Frasca, G. (2001). *Videogames of the oppressed: Videogames as a means for critical thinking and debate* (Tesis de maestría). Georgia Institute of Technology, Atlanta, Estados Unidos.
- Frasca, G. (2003). Ludologists love stories, too: notes from a debate that never took place. En M. Copier y J. Raessens (Eds.), *Actas de Level Up: Digital Games Research Conference.* Utrecht: DiGRA y University of Utrecht. Recuperado de: [http://www.ludology.org/articles/frasca\\_levelUP2003.pdf](http://www.ludology.org/articles/frasca_levelUP2003.pdf)
- Frasca, G. (2009). Juego, videojuego y creación de sentido. Una introducción. *Revista Internacional de Comunicación Audiovisual, Publicidad y Literatura*, 1(7), 37–44.
- Fraser, S., Beck, K., Caputo, B., Mackinnon, T., Newkirk, J. y Poole, C. (2003). Test driven development (TDD). En *International Conference on Extreme Programming and Agile Processes in Software Engineering* (pp. 459–462). Springer.
- Free Software Foundation. (s. f.). GCC, the GNU Compiler Collection. Recuperado de: <http://gcc.gnu.org>
- Freeman, L. C. (2004). *The development of social network analysis: a study in the sociology of science.* Vancouver: Empirical Press.

- Freiberger, P. y Swaine, M. (2000). *Fire in the Valley: the making of the personal computer* (2ª ed.). New York: McGraw-Hill.
- Fundación Argentina de Videojuegos. (s. f.). Recuperado de: <http://fundav.com>
- Gamarekian, B. (1982, abril 9). 8-Year-old's Birthday Party in a Computer Center. *The New York Times*. Recuperado de: <http://www.nytimes.com/1982/04/09/style/8-year-old-s-birthday-party-in-a-computer-center.html>
- Game of Work. (s. f.). Recuperado de: <http://www.gameofwork.com>
- Gamma, E., Helm, R., Johnson, R. y Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education.
- García-Mundo, L., Vargas-Enríquez, J., Genero, M. y Piattini, M. (2014). ¿Contribuye el Uso de Juegos Serios a Mejorar el Aprendizaje en el Área de la Informática? En *Actas de las XX Jornadas de Enseñanza Universitaria de la Informática* (pp. 303-310). Oviedo.
- Garlick, R. y Akl, R. G. (2007). Motivating and Retaining CS2 Students with a Competitive Game Programming Project. *International Network for Engineering Education and Research (iNEER) Special Volume: Innovations 2007-World Innovations in Engineering Education and Research*, 1-9.
- Gartner's 2011 Hype Cycle Special Report Evaluates the Maturity of 1,900 Technologies. (s. f.). Recuperado de: <http://www.gartner.com/newsroom/id/1763814>
- Garzón, G. (2016, julio). La educación en juego. *Para Juanito. Revista de Educación Popular y Pedagogías Críticas*, 4(9), 10-16.
- Gee, E. R. y Tran, K. M. (2016). Video Game Making and Modding. En B. Guzzetti y M. Lesley (Eds.), *Handbook of Research on the Societal Impact of Digital Media*. Hershey: IGI Global.
- Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave Macmillan.
- Globaloria. (s. f.). Recuperado de: <http://globaloria.com>
- Gloss. (s. f.). Recuperado de: <http://hackage.haskell.org/package/gloss>

- Goldberg, A. y Ross, J. (1981, agosto). Is the Smalltalk-80 System for Children? *BYTE*, 6(8), 348-267.
- Goleman, D. (1996). *Inteligencia emocional*. Barcelona: Editorial Kairós.
- Goleman, D. (1998). *La práctica de la inteligencia emocional*. Barcelona: Editorial Kairós.
- González González, C. S. y Mora Carreño, A. (2015). Técnicas de gamificación aplicadas en la docencia de Ingeniería Informática. *ReVision*, 8(1). Recuperado de: <http://www.aenui.net/ojs/index.php?journal=revision&page=article&op=viewArticle&path%5B%5D=152&path%5B%5D=290>
- Green, D. y Green, D. (1982, abril 26). Karel the Robot: An educational Pascal simulator. *InfoWorld*, 4(16), 28-30.
- Grover, S. y Pea, R. D. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43.
- Guzdial, M. (2003). A media computation course for non-majors. En *ACM SIGCSE Bulletin* (Vol. 35, pp. 104-108). ACM.
- Guzdial, M. (2016). *Learner-Centered Design of Computing Education: Research on Computing for Everyone*. San Rafael: Morgan & Claypool Publishers.
- Guzdial, M. y Morrison, B. (2016). Growing computer science education into a STEM education discipline. *Communications of the ACM*, 59(11), 31-33.
- Hafner, K. y Lyon, M. (1998). *Where wizards stay up late: the origins of the Internet*. New York: Touchstone.
- Hakulinen, L., Auvinen, T. y Korhonen, A. (2015). The Effect of Achievement Badges on Students' Behavior: An Empirical Study in a University-Level Computer Science Course. *International Journal of Emerging Technologies in Learning (iJET)*, 10(1), 18.
- Halma, A. (2007). *Robo: Een Introductie tot Programmeren*. Universiteit van Amsterdam. Recuperado de: <http://www.robomind.net/downloads/RoboIntroductie.pdf>
- Harel, I. y Papert, S. (Eds.). (1991). *Constructionism*. Norwood: Ablex Publishing Corporation.

- Hayes, E. R. y Games, I. A. (2008). Making Computer Games and Design Thinking: A Review of Current Software and Strategies. *Games and Culture*, 3(3-4), 309-332.
- Hemmendinger, D. (2007). The ACM and IEEE-CS guidelines for undergraduate CS education. *Communications of the ACM*, 50(5), 46-53.
- Herman, T. (2014). *A Functional Start to Computing with Python*. Boca Raton: CRC Press.
- Hernández Sampieri, R., Fernández Collado, C. y Baptista Lucio, P. (2014). *Metodología de la investigación* (6ª ed.). México, D.F.: McGraw-Hill Education.
- Hiltzik, M. A. (1999). *Dealers of lightning: Xerox PARC and the dawn of the computer age*. New York: HarperCollins.
- Hour of Code. (s. f.). Recuperado de: <http://code.org/learn>
- Huizinga, J. (1972). *Homo ludens*. (E. Imaz, Trad.). Madrid: Alianza/Emecé. (Obra original publicada en 1938).
- Human Resource Machine. (s. f.). Recuperado de: <http://tomorrowcorporation.com/humanresourcemachine>
- Hunicke, R., LeBlanc, M. y Zubek, R. (2004). MDA: A formal approach to game design and game research. En *Proceedings of the Challenges in Game AI Workshop at the 19th National Conference on Artificial Intelligence*.
- Hunt, A. (2014). *Learn to program with Minecraft plugins. Create flaming cows in Java using CanaryMod* (2ª ed.). Dallas: The Pragmatic Bookshelf.
- Huotari, K. y Hamari, J. (2012). Defining gamification: a service marketing perspective. En *Proceeding of the 16th international academic MindTrek conference* (pp. 17-22). ACM.
- Ingalls, D. (1981). Design principles behind Smalltalk. *BYTE Magazine*, 6(8), 286-298.
- Ingalls, D., Kaehler, T., Maloney, J., Wallace, S. y Kay, A. C. (1997). Back to the future: the story of Squeak, a practical Smalltalk written in itself. En *ACM SIGPLAN Notices* (Vol. 32, pp. 318-326). ACM.
- INSNA. (s. f.). Recuperado de: <http://www.insna.org>

- Ivory, J. D. (2016). A brief history of video games. En R. Kowert y T. Quandt (Eds.), *The Video Game Debate: Unravelling the Physical, Social, and Psychological Effects of Video Games* (pp. 1-21). New York: Routledge, Taylor & Francis Group.
- Jacobs, G. M., Renandya, W. A. y Power, M. (2016). *Simple, Powerful Strategies for Student Centered Learning*. Springer International Publishing.
- Jacobson, I., Booch, G. y Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Madrid: Addison-Wesley.
- Jacovkis, P. M. (2011). Un lugar para Clementina. *La Ménsula*, (13), 1-8.
- Java 3D API. (s. f.). Recuperado de: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138252.html>
- Jenkins, H. (2000). Art form for the digital age. *MIT Technology Review*, 103(5), 117–120.
- Jenkins, H. (2005). Games, the new lively art. En J. Raessens y J. Goldstein (Eds.), *Handbook of Computer Game Studies* (pp. 175–189). Cambridge: MIT Press.
- Jenkins, T. (2002). On the difficulty of learning to program. En *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, pp. 53–58).
- jMonkeyEngine. (s. f.). Recuperado de: <http://jmonkeyengine.org>
- Johnson, L., Adams Becker, S., Cummins, M., Estrada, V., Freeman, A. y Ludgate, H. (2013). *NMC Horizon Report: Edición Educación Superior 2013*. Austin, Estados Unidos: The New Media Consortium.
- Juul, J. (2003). The Game, the Player, the World: Looking for a Heart of Gameness. En M. Copier y J. Raessens (Eds.), *Level Up: Digital Games Research Conference Proceedings* (pp. 30-45). Utrecht: Utrecht University. Recuperado de: <http://www.jesperjuul.net/text/gameplayerworld>
- Juul, J. (2009). Ludology. En B. Perron y M. J. P. Wolf (Eds.), *The Video Game Theory Reader 2* (pp. 362-364). New York: Routledge, Taylor & Francis Group, Inc.
- Kaasbøll, J. J. (1998). Exploring didactic models for programming. En *NIK 98 - Norwegian Computer Science Conference* (pp. 195–203).

- Kaczmarczyk, L. y Dopplick, R. (2014). *Rebooting the Pathway to Success: Preparing Students for Computing Workforce Needs in the United States*. New York: Association for Computing Machinery. Recuperado de: [http://inis.iaea.org/search/search.aspx?orig\\_q=RN:13668864](http://inis.iaea.org/search/search.aspx?orig_q=RN:13668864)
- Kafai, Y. B. (1995). *Minds in Play: Computer Game Design As a Context for Children's Learning*. Hillsdale: L. Erlbaum Associates Inc.
- Kafai, Y. B. y Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. Cambridge, Estados Unidos: MIT Press.
- Kafai, Y. B. y Burke, Q. (2015). Constructionist Gaming: Understanding the Benefits of Making Games for Learning. *Educational Psychologist*, 50(4), 313-334.
- Kafai, Y. B. y Burke, Q. (2016). *Connected Gaming: What Making Video Games Can Teach Us about Learning and Literacy*. Cambridge, Estados Unidos: MIT Press.
- Kahoot! (s. f.). Recuperado de: <http://kahoot.com>
- Kalinichenko, E. (s. f.). International Olympiad in Informatics – Statistics. Recuperado de: <http://stats.ioinformatics.org>
- Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. San Francisco: Pfeiffer.
- Karsenty, R. (2014). Mathematical Ability. En S. Lerman (Ed.), *Encyclopedia of Mathematics Education* (pp. 372–375). Dordrecht: Springer Netherlands.
- Kay, A. C. (1972). A Personal Computer for Children of All Ages. En *Proceedings of the ACM Annual Conference - Volume 1*. New York: ACM. Recuperado de: <http://www.mprove.de/diplom/gui/Kay72a.pdf>
- Kay, A. C. (1977). Microelectronics and the Personal Computer. *Scientific American*. Recuperado de: <http://mnielsen.github.io/notes/kay/micro.pdf>
- Kay, A. C. (1996). The early history of Smalltalk. En T. J. Bergin y R. G. Gibson (Eds.), *History of programming languages* (Vol. II, pp. 511-598). New York: ACM Press/Addison-Wesley.
- Kay, A. C. (2007). *Children Learning by Doing: Squeak Etoys on the OLPC XO* (VPRI Research Note No. RN-2007-006-a) (pp. 1–51).

- Kelleher, C. y Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83–137.
- Kemeny, J. G. y Kurtz, T. E. (1968). Dartmouth Time-Sharing. *Science*, 162, 223–228.
- Kenderov, P. S. y Maneva, N. M. (Eds.). (1989). IOI. Proceedings of the International Olympiad in Informatics. Pravetz: Union of the Mathematicians in Bulgaria.
- Kent, S. L. (2001). *The ultimate history of video games: from Pong to Pokémon and beyond: the story behind the craze that touched our lives and changed the world*. New York: Three Rivers Press.
- Kodu | About. (s. f.). Recuperado de: <http://www.kodugamelab.com/about>
- Kölling, M. (2010). *Introduction to programming with Greenfoot: object-oriented programming in Java with games and simulations*. Upper Saddle River: Prentice Hall.
- Kölling, M. (2015). Lessons from the Design of Three Educational Programming Environments: Blue, BlueJ and Greenfoot. *International Journal of People-Oriented Programming*, 4(1), 5-32.
- Kölling, M. y Henriksen, P. (2005). Game programming in introductory courses with direct state manipulation. En *ACM SIGCSE Bulletin*. ACM Press.
- Kölling, M. y Rosenberg, J. (1996). Blue: a language for teaching object-oriented programming. En *ACM SIGCSE Bulletin* (Vol. 28, pp. 190–194). ACM.
- Korn Ferry. (s. f.). Recuperado de: <http://www.kornferry.com>
- Kotrlik, J. W. y Williams, H. A. (2003). The incorporation of effect size in information technology, learning, and performance research. *Information Technology, Learning, and Performance Journal*, 21(1).
- Kramer, W. (2015). What makes a game good? *Game & Puzzle Design*, 1(2), 84-86.
- Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, 41(4), 212–218.



- Krogue, K. (2012, septiembre 18). 5 Gamification Rules From The Grandfather Of Gamification. Recuperado de: <http://www.forbes.com/sites/kenkrogue/2012/09/18/5-gamification-rules-from-the-grandfather-of-gamification>
- Kroll, J. (2000, marzo 5). «Emotion Engine»? I Don't Think So. *Newsweek*. Recuperado de: <http://www.newsweek.com/emotion-engine-i-dont-think-so-156675>
- Kumar, B. y Khurana, P. (2012). Gamification in education-learn computer programming with fun. *International Journal of Computers and Distributed Systems*, 2(1), 46–53.
- Lambert, K. A. y Osborne, M. (2012). *Fundamentals of Python: first programs*. Boston: Course Technology.
- Leal, J. P., Paiva, J. C. y Peixoto de Queirós, R. A. (2015). Odin: A service for gamification of learning activities. En *International Symposium on Languages, Applications and Technologies* (pp. 194–204). Springer.
- Leuner, B. (1966). Emotionale Intelligenz und Emanzipation. *Praxis der Kinderpsychologie und Kinderpsychiatrie*, 15(6), 196–203.
- Lewis, P. H. (1987, agosto 18). PERIPHERALS; It's, Well, HyperCard. *The New York Times*. Recuperado de: <http://www.nytimes.com/1987/08/18/science/peripherals-it-s-well-hypercard.html>
- Ley 25.922 - Ley de Promoción de la Industria del Software (2014). Recuperado de: [http://www.agencia.mincyt.gob.ar/upload/ley\\_25922-2.pdf](http://www.agencia.mincyt.gob.ar/upload/ley_25922-2.pdf)
- Li, C., Dong, Z., Untch, R. H. y Chasteen, M. (2013). Engaging Computer Science Students through Gamification in an Online Social Network Based Collaborative Learning Environment. *International Journal of Information and Education Technology*, 72–77.
- Li, S. (2002). *Rock 'em, sock 'em Robocode!* (developerWorks). IBM Corporation.
- LibGDX. (s. f.). Recuperado de: <http://libgdx.badlogicgames.com>
- Livingstone, I. y Hope, A. (2011). *Next Gen: Transforming the UK into the world's leading talent hub for the video games and visual effects industries*. National Endowment for Science, Technology and the Arts. Recuperado de: [http://www.nesta.org.uk/sites/default/files/next\\_gen\\_wv.pdf](http://www.nesta.org.uk/sites/default/files/next_gen_wv.pdf)

- Loguidice, B. y Barton, M. (2014). *Vintage game consoles: an inside look at Apple, Atari, Commodore, Nintendo, and the greatest gaming platforms of all time*. New York: Focal Press, Taylor & Francis Group.
- Loh, C. S., Sheng, Y. y Ifenthaler, D. (2015). Serious Games Analytics: Theoretical Framework. En C. S. Loh, Y. Sheng y D. Ifenthaler (Eds.), *Serious Games Analytics. Methodologies for Performance Measurement, Assessment, and Improvement* (pp. 3-30). Cham: Springer International Publishing.
- Lohr, S. (2017, agosto 24). As Coding Boot Camps Close, the Field Faces a Reality Check. *The New York Times*. Recuperado de: <http://www.nytimes.com/2017/08/24/technology/coding-boot-camps-close.html>
- Lomax, R. G. y Hahs-Vaughn, D. L. (2012). *Statistical Concepts: A Second Course* (4<sup>a</sup> ed.). New York: Routledge.
- LÖVE. (s. f.). Recuperado de: <http://love2d.org>
- Ludificación, mejor que Gamificación como traducción de Gamification. (2012, junio 22). Recuperado de: <http://www.fundeu.es/recomendacion/ludificacion-mejor-que-gamificacion-como-traducccion-de-gamification-1390>
- Lunazzi, J. M. (1947). Ludología y Ludoterapia. *Boletín del Museo Social Argentino*, XXXV(297-298), 79-83.
- LWJGL - Lightweight Java Game Library. (s. f.). Recuperado de: <http://www.lwjgl.org>
- Maigaard, P. (1950). About Ludology. En C. Gini (Ed.), *Proceedings of the 14th International Congress of Sociology* (pp. 362-373). Roma: Institut International de Sociologie.
- Malone, T. W. (1980). *What makes things fun to learn? A study of intrinsically motivating computer games*. Palo Alto: XEROX PARC.
- MaMaMedia.com. (s. f.). Recuperado de: <http://worldwideworkshop.org/programs/past-programs/learning/mamamedia-com>
- Marcano Lárez, B. E. (2008). Juegos serios y entrenamiento en la sociedad digital. *Revista Electrónica Teoría de la Educación: Educación y Cultura en la Sociedad de la Información*, 9(3). Recuperado de: <http://gredos.usal.es/jspui/handle/10366/56633>

- Maxwell, S. E. y Delaney, H. D. (2004). *Designing experiments and analyzing data: a model comparison perspective* (2da. ed.). Mahwah: Lawrence Erlbaum Associates.
- May, J. y Dhillon, G. (2009). Interpreting beyond syntactics: A semiotic learning model for computer programming languages. *Journal of Information Systems Education*, 20(4), 431.
- Mayer, J. D., Caruso, D. R. y Salovey, P. (1999). Emotional intelligence meets traditional standards for an intelligence. *Intelligence*, 27(4), 267–298.
- Mayer, J. D. y Salovey, P. (1997). What is emotional intelligence? En P. Salovey y D. J. Sluyter (Eds.), *Emotional development and emotional intelligence: Educational implications* (pp. 3-31). New York: Basic Books.
- McCormick, R. y James, M. (1995). *Evaluación del curriculum en los centros escolares*. Madrid: Ediciones Morata, S. L.
- McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., ... Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4), 125–180.
- Meike, R. (2012, octubre 24). Digital Diner: Arduino Uno vs BeagleBone vs Raspberry Pi. Recuperado de: <http://digitaldiner.blogspot.com.ar/2012/10/arduino-uno-vs-beaglebone-vs-raspberry.html>
- Membrey, P. y Hows, D. (2015). *Learn Raspberry Pi 2 with Linux and Windows 10*. New York: Apress.
- Meyer, B. (1988). *Object-oriented software construction* (Vol. 2). Prentice hall New York.
- Meyer, J. (2010). *The essential guide to HTML5: Using games to learn HTML5 and JavaScript*. New York: Apress.
- MHS Inc. (s. f.). Recuperado de: <http://www.mhs.com/MHS-Talent?prodname=msceit>
- Michael, D. R. y Chen, S. (2006). *Serious Games: Games that Educate, Train and Inform*. Boston: Thomson Course Technology.

- Mikulic, I. M., Caballero, R. Y., Crespi, M. C. y Radusky, P. D. (2013). Adaptación argentina del test de Inteligencia Emocional de Mayer-Salovey-Caruso (MSCEIT V 2.0). *Anuario de investigaciones*, 20(1), 377–386.
- Mitchell, N., Danino, N. y May, L. (2013). Motivation and manipulation: A gamification approach to influencing undergraduate attitudes in computing. En *European Conference on Games Based Learning* (p. 394). Academic Conferences International Limited.
- Mondolfo, R. (1966). *Heráclito: textos y problemas de su interpretación*. (O. Caletti, Trad.). México: Siglo XXI.
- MonoGame. (s. f.). Recuperado de: <http://www.monogame.net>
- Moodle. (s. f.). Recuperado de: <http://moodle.org/?lang=es>
- Moon, J. (2012, enero 9). What Hacker Apprenticeships Tell Us About the Future of Education. *The Atlantic*. Recuperado de: <http://www.theatlantic.com/technology/archive/2012/01/what-hacker-apprenticeships-tell-us-about-the-future-of-education/251039>
- Mora Carreño, A., Riera, D., González González, C. S. y Arnedo-Moreno, J. (2015). A Literature Review of Gamification Design Frameworks. Presentado en 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games), Skovde, Suecia.
- Morales Moras, J. (2015). *Serious games: Diseño de videojuegos con una agenda educativa y social*. Barcelona: Editorial UOC.
- Moreno, J. L. (1934). *Who shall survive?: A new approach to the problem of human interrelations*. Washington, DC: Nervous and Mental Disease Publishing Co.
- Moreno, J. y Montaña, E. (2009). ProBot: Juego para el aprendizaje de lógica de programación. *Nuevas Ideas en Informática Educativa*, 5, 1-7.
- Morris, C. W. (1985). *Fundamentos de la teoría de los signos*. (R. Grasa, Trad.). Barcelona: Paidós. (Obra original publicada en 1938).
- Morris, D., Uppal, G. y Wells, D. (2017). *Teaching Computational Thinking and Coding in Primary Schools*. SAGE Publications.

- Moss, F. (2011). *The sorcerers and their apprentices: How the digital magicians of the MIT Media Lab are creating the innovative technologies that will transform our lives*. New York: Crown Business.
- Muijs, D. (2004). *Doing quantitative research in education with SPSS*. London ; Thousand Oaks: Sage Publications.
- Mumuki. (s. f.). Recuperado de: <http://program.ar/mumuki-aprender-y-ensinar-a-programar>
- Munné, F. (1980). *Psicosociología del tiempo libre: un enfoque crítico*. México D.F.: Trillas.
- Nagle, J. (2015). *Getting to know Alice*. New York: The Rosen Publishing Group, Inc.
- NATO Science Committee. (1969). *Software Engineering* (Report on the NATO Software Engineering Conference 1968). Garmisch, Alemania. Recuperado de: <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>
- Needle, D. (1984, julio 2). Philips will make MSX computer. *InfoWorld*, 6(27), 9.
- New Gaming Boom: Newzoo Ups Its 2017 Global Games Market Estimate to \$116.0Bn Growing to \$143.5Bn in 2020. (2017, noviembre 28). Recuperado de: <http://newzoo.com/insights/articles/new-gaming-boom-newzoo-ups-its-2017-global-games-market-estimate-to-116-0bn-growing-to-143-5bn-in-2020>
- Nichols, R. (2013). Who plays, who pays? Mapping video game production and consumption globally. En N. B. Huntemann y B. Aslinger (Eds.), *Gaming Globally: Production, Play, and Place* (pp. 19–39). New York: Palgrave Macmillan.
- Nicholson, S. (2012). Strategies for meaningful gamification: Concepts behind transformative play and participatory museums. Presentado en Meaningful Play 2012, Lansing, Michigan. Recuperado de: <http://scottnicholson.com/pubs/meaningfulstrategies.pdf>
- Nicholson, S. (2015). A RECIPE for Meaningful Gamification. En T. Reiners y L. C. Wood (Eds.), *Gamification in Education and Business* (pp. 1-20). Cham: Springer International Publishing.
- Nickerson, R. S. (2015). *Conditional reasoning: the unruly syntactics, semantics, thematics, and pragmatics of «if»*. New York: Oxford University Press.

- Nisnevich, A. (2014, octubre 17). Some words on Untrusted. Recuperado de: [http://alex.nisnevich.com/blog/2014/10/17/some\\_words\\_on\\_untrusted.html](http://alex.nisnevich.com/blog/2014/10/17/some_words_on_untrusted.html)
- No One Left Behind. (s. f.). Recuperado de: <http://no1leftbehind.eu>
- Norman, M. D. (2018). *Easy Minecraft Mod Programming: Learn to Code Minecraft Mods with Kotlin*. CreateSpace Independent Publishing Platform.
- OCDE. (1991). *Proyecto Atenea: Informe de Evaluación*. Madrid: Ministerio de Educación y Ciencia.
- Olejnik, S. F. y Algina, J. (1985). A Review of Nonparametric Alternatives To Analysis of Covariance. *Evaluation Review*, 9(1), 51-83.
- Orejuela, H. F., García, A. A., Hurtado, J. A. y Collazos, C. A. (2013). Analizando y Aplicando la Gamificación en el Proceso ChildProgramming. *Revista Colombiana de Computación*, 14(2), 7-23.
- Osorio, J. y Nieves, J. (2014). The Beginnings of Computer Use in Primary and Secondary Education in Spain. En A. Tatnall y B. Davey (Eds.), *Reflections on the History of Computers in Education. Early Use of Computers and Teaching about Computing in Schools* (pp. 121–130). Springer.
- O'Sullivan, D. y Igoe, T. (2004). *Physical computing: sensing and controlling the physical world with computers*. Boston: Thomson.
- Paisley, W. y Chen, M. (1984). The Second Electronic Revolution: The Computer and Children. *Annals of the International Communication Association*, 8, 106-136.
- Paiva, J. C., Leal, J. P. y Peixoto de Queirós, R. A. (2016). Design and Implementation of an IDE for Learning Programming Languages Using a Gamification Service. En R. A. Peixoto de Queirós y M. Teixeira Pinto, *Gamification-Based E-Learning Strategies for Computer Programming Education* (pp. 295-308). Hershey: IGI Global.
- Pajek: analysis and visualization of large networks. (s. f.). Recuperado de: <http://mrvar.fdv.uni-lj.si/pajek>
- Pallant, J. (2011). *SPSS survival manual: a step by step guide to data analysis using SPSS* (4ª ed.). Crows Nest: Allen & Unwin.

- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books.
- Papert, S. y Harel, I. (1991). Situating Constructionism. En I. Harel y S. Papert (Eds.), *Constructionism* (pp. 1-11). Norwood: Ablex Publishing Corporation.
- Parker, J. R. (2017). *Python: An Introduction to Programming*. Dulles: Mercury Learning and Information LLC.
- Patella-Rey, P. J. (2012, octubre 16). Gamification, Playbor & Exploitation. Recuperado de: <http://thesocietypages.org/cyborgology/2012/10/15/gamification-playbor-exploitation-2>
- Pattis, R. E. (1981). *Karel the Robot: A Gentle Introduction to the Art of Programming*. New York: John Wiley & Sons.
- Pea, R. D. y Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New ideas in psychology*, 2(2), 137-168.
- Pellas, N. y Peroutseas, E. (2016). Gaming in Second Life via Scratch4SL: Engaging High School Students in Programming Courses. *Journal of Educational Computing Research*, 54(1), 108-143.
- Pelling, N. (2011, agosto 9). The (short) prehistory of «gamification». Recuperado de: <http://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification>
- Pencil Code. (s. f.). Recuperado de: <http://pencilcode.net>
- Pérez-González, J. C., Petrides, K. V. y Furnham, A. (2005). Measuring trait emotional intelligence. En R. Schulze y R. D. Roberts (Eds.), *Emotional intelligence: An international handbook* (pp. 181-201). Cambridge: Hogrefe & Huber.
- Pérez-González, J. C., Petrides, K. V. y Furnham, A. (2007). La medida de la inteligencia emocional rasgo. En *Manual de inteligencia emocional* (pp. 81-98). Ediciones Pirámide.
- Petrides, K. V. (2009). Psychometric Properties of the Trait Emotional Intelligence Questionnaire (TEIQue). En C. Stough, D. H. Saklofske y J. D. A. Parker (Eds.), *Assessing Emotional Intelligence* (pp. 85-101). Boston, MA: Springer US.

- Petrides, K. V. y Furnham, A. (2001). Trait emotional intelligence: psychometric investigation with reference to established trait taxonomies. *European Journal of Personality*, 15(6), 425-448.
- Petrides, K. V. y Furnham, A. (2003). Trait emotional intelligence: behavioural validation in two studies of emotion recognition and reactivity to mood induction. *European Journal of Personality*, 17(1), 39-57.
- Phaser. (s. f.). Recuperado de: <http://phaser.io>
- Plan 111mil. (2016, octubre 18). Recuperado de: <http://www.argentina.gob.ar/111mil>
- Planells de la Maza, A. J. (2013). La emergencia de los Game Studies como disciplina propia: investigando el videojuego desde las metodologías de la Comunicación. *Historia y Comunicación Social*, 18, 519-528.
- PLANIED - Plan Nacional Integral de Educación Digital. (2016). Recuperado de: <http://planied.educ.ar/institucional/acerca-de>
- PNIDE - Plan Nacional Inclusión Digital Educativa. (2015). Recuperado de: <http://www.tic.siteal.org/politicas/1275/plan-nacional-inclusion-digital-educativa>
- Portuondo, J. A. (1971). *El test sociométrico*. Madrid: Biblioteca Nueva.
- Prensky, M. (2001). *Digital Game-based Learning*. New York: McGraw-Hill.
- Prensky, M. (2006). *Don't Bother Me Mom—I'm Learning!: How Computer and Video Games are Preparing Your Kids for 21st Century Success - and How You Can Help!* St. Paul: Paragon House.
- Programa Codo a Codo. (s. f.). Recuperado de: <http://www.buenosaires.gob.ar/educacion/codoacodo/preguntas-frecuentes>
- Program.AR. (2014). Recuperado de: <http://program.ar/de-que-se-trata>
- Psychometric Lab. (s. f.). Recuperado de: <http://www.psychometriclab.com>
- Puri, M. L. y Sen, P. K. (1969). Analysis of covariance based on general rank scores. *The Annals of Mathematical Statistics*, 40(2), 610-618.
- Pygame. (s. f.). Recuperado de: <http://www.pygame.org>
- Quade, D. (1967). Rank analysis of covariance. *Journal of the American Statistical Association*, 62(320), 1187-1200.



- ¿Qué es Minecraft? (s. f.). Recuperado de: <http://minecraft.net/es-es/what-is-minecraft>
- Ramírez Cogollor, J. L. (2014). *Gamificación: mecánicas de juegos en tu vida personal y profesional*. México: Alfaomega.
- Randel, J. M., Morris, B. A., Wetzel, C. D. y Whitehill, B. V. (1992). The effectiveness of games for educational purposes: A review of recent research. *Simulation & gaming*, 23(3), 261–276.
- Real Academia Española. Diccionario Usual. (2018). Recuperado de: <http://dle.rae.es/srv/search?m=30&w=videojuego>
- Reas, C. y Fry, B. (2007). *Processing: a programming handbook for visual designers and artists*. Cambridge, Mass: MIT Press.
- REDES. (s. f.). Recuperado de: <http://revista-redes.rediris.es>
- Reggini, H. C. (1982). *Alas para la mente. LOGO: Un lenguaje de computadoras y un estilo de pensar*. Buenos Aires: Ed. Galápagos.
- Reggini, H. C. (1988). *Computadoras: ¿creatividad o automatismo?* Buenos Aires: Ediciones Galápagos.
- Repenning, A. (1991). Creating user interfaces with Agentsheets. En *Proceedings of the 1991 IEEE Symposium on Applied Computing* (pp. 191–196). IEEE.
- Repenning, A. (2011). Making programming more conversational. En *Visual Languages and Human-Centric Computing (VL/HCC), 2011 IEEE Symposium on* (pp. 191–194). IEEE.
- Repenning, A. y Ambach, J. (1996). Tactile Programming: A unified manipulation paradigm supporting program comprehension, composition and sharing. En *Proceedings of the 1996 IEEE Symposium on Visual Languages* (pp. 102–109). IEEE.
- Repenning, A., Webb, D. y Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. En *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 265–269). ACM Press.
- Resnick, M. (1997). *Turtles, termites, and traffic jams: Explorations in massively parallel microworlds*. Cambridge: The MIT Press.

- Resnick, M., Ocko, S. y Papert, S. (1988). LEGO, Logo, and design. *Children's Environments Quarterly*, 5(4), 14-18.
- Resolución 263/15 del Consejo Federal de Educación (2015). Recuperado de: <http://www.me.gov.ar/consejo/resoluciones/res15/263-15.pdf>
- Resolución 441/2000 del Ministerio de Educación de la Nación (2000). Recuperado de: <http://www.educ.ar/sitios/educar/institucional/estatuto>
- Revuelta Domínguez, F. I. y Guerra Antequera, J. (2012). ¿Qué aprendo con videojuegos? Una perspectiva de meta-aprendizaje del videojugador. *RED. Revista de Educación a Distancia*, 33. Recuperado de: <http://www.um.es/ead/red/33/revuelta.pdf>
- Richardson, C. (2016). *Learn to program with Minecraft: transform your world with the power of Python*. San Francisco: No Starch Press.
- rise.global Scorecard. (s. f.). Recuperado de: <http://www.rise.global/genie/index>
- Ritchie, D. M. (1993). The Development of the C Language. En *The Second ACM SIGPLAN Conference on History of Programming Languages* (pp. 201–208). New York: ACM Press.
- Robertson, D. y Breen, B. (2013). *Brick by brick: How LEGO rewrote the rules of innovation and conquered the global toy industry*. New York: Crown Business.
- Robertson, M. (2010, octubre 6). Can't play, won't play. Recuperado de: [www.hideandseek.net/2010/10/06/cant-play-wont-play](http://www.hideandseek.net/2010/10/06/cant-play-wont-play)
- Robins, A., Rountree, J. y Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172.
- Rodríguez, F. y Santiago, R. (2015). *Gamificación: Cómo motivar a tu alumnado y mejorar el clima en el aula*. Madrid: Grupo Océano.
- Roschelle, J. M., Pea, R. D., Hoadley, C. M., Gordin, D. N. y Means, B. M. (2000). Changing how and what children learn in school with computer-based technologies. *The future of children*, 10(2), 76–101.
- Roy, S. R. (2015). *Promoting Trait Emotional Intelligence in Leadership and Education*. Hershey: IGI Global.

- Royce, W. W. (1970). Managing the Development of Large Software Systems. En *Technical Papers of Western Electronic Show and Convention (WesCon)* (pp. 1-9). Los Angeles.
- Rumbaugh, J., Booch, G. y Jacobson, I. (2000). *El lenguaje unificado de modelado: manual de referencia*. Addison Wesley.
- Russell, J. P. (2001). *Java Programming for the Absolute Beginner*. Roseville: Prima Publishing.
- Saarni, C. (1990). Emotional competence: How emotions and relationships become integrated. En R. A. Thompson (Ed.), *Nebraska Symposium on Motivation, 1988: Socioemotional development* (Vol. 36, pp. 115-182). Lincoln: University of Nebraska Press. Recuperado de: <http://books.google.com.ar/books?id=kMsKKk6dovcC>
- Sadosky, M. (2011). *Sadosky por Sadosky: vida y pensamiento del pionero de la computación argentina*. (R. Carnota y C. Borches, Eds.). Buenos Aires: Fundación Sadosky.
- Salen, K. y Zimmerman, E. (2003). *Rules of Play: Game Design Fundamentals*. Cambridge: MIT Press.
- Salovey, P. y Mayer, J. D. (1990). Emotional intelligence. *Imagination, cognition and personality*, 9(3), 185-211.
- Salovey, P., Mayer, J. D., Goldman, S. L., Turvey, C. y Palfai, T. P. (1995). Emotional attention, clarity, and repair: Exploring emotional intelligence using the Trait Meta-Mood Scale. En J. W. Pennebaker (Ed.), *Emotion, disclosure, & health* (pp. 125-154). American Psychological Association.
- Scott, M. J. y Ghinea, G. (2013). Educating Programmers: A Reflection on Barriers to Deliberate Practice. Presentado en 2nd Annual HEA STEM Conference, Birmingham, UK: The Higher Education Academy.
- Scratch - Imagine, Program, Share. (s. f.). Recuperado de: <http://scratch.mit.edu/statistics>
- ScratchJr. (s. f.). Recuperado de: <http://www.scratchjr.org>
- Scriven, M. (1967). The methodology of evaluation. En R. W. Tyler, R. M. Gagné y M. Scriven (Eds.), *Perspectives of Curriculum Evaluation*. American Educational Research

- Association Monograph Series on Curriculum Evaluation* (Vol. 1, pp. 39-83). Chicago: Rand McNally.
- SDL - Simple DirectMedia Layer. (s. f.). Recuperado de: <http://www.libsdl.org>
- Sebesta, R. W. (2012). *Concepts of programming languages* (10ª ed.). Boston: Pearson.
- Second Life. (s. f.). Recuperado de: <http://secondlife.com>
- Secretary-General's address to the World Summit on the Information Society. (2003, diciembre 10). Recuperado de: <http://www.un.org/sg/en/content/sg/statement/2003-12-10/secretary-generals-address-world-summit-information-society>
- Seldes, G. (1924). *The 7 Lively Arts*. New York: Harper & Brothers Publishers.
- Serrano, M. I. (2014, septiembre 9). «Programación», la nueva asignatura que convertirá a los madrileños en aprendices de Bill Gates. *ABC.es*. Madrid. Recuperado de: <http://www.abc.es/madrid/20140909/abcp-programacion-nueva-asignaturaaprendices-bill-20140908.html>
- SFML - Simple and Fast Multimedia Library. (s. f.). Recuperado de: <http://www.sfml-dev.org>
- Shabanah, S. (2011). Computer Games for Algorithm Learning. En P. Felicia (Ed.), *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches* (pp. 1036-1063). Hershey: IGI Global.
- Shabanah, S. y Chen, J. X. (2009). Simplifying algorithm learning using serious games. En *Proceedings of the 14th Western Canadian Conference on Computing Education* (pp. 34-41). ACM.
- Shapiro, N. (1984, abril). New standards are being set in many fields. *Popular Mechanics*, 161(4), 10.
- Sheikh, R. (2017, diciembre 20). Video games: How big is industry's racial diversity problem? *BBC News*. Recuperado de: <http://www.bbc.co.uk/news/technology-42357678>
- Sherrod, A. (2015). *iOS for Game Programmers*. Dulles: Mercury Learning and Information, LLC.

- Sierra, K. y Bates, B. (2017). *OCA Java SE 8 Programmer I Exam Guide (Exams 1Z0-808)*. McGraw-Hill Education.
- Simões, A. (2016). Using Game Frameworks to Teach Computer Programming. En R. A. Peixoto de Queirós y M. Teixeira Pinto, *Gamification-Based E-Learning Strategies for Computer Programming Education* (pp. 221-237). Hershey: IGI Global.
- Smith, D. K. y Alexander, R. C. (1988). *Fumbling the future: how Xerox invented, then ignored, the first personal computer*. New York: W. Morrow.
- Smith, M. (2016, enero 30). Computer Science For All. Recuperado de: <http://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>
- Soares, A., Fonseca, F. y Martin, N. L. (2015). Teaching introductory programming with game design and problem-based learning. *Issues in Information Systems*, 16(III), 128-137.
- SQUEEZED MIDDLE is named Oxford Dictionaries Word of the Year 2011. (2011, noviembre 22). Recuperado de: <http://www.prnewswire.com/news-releases/squeezed-middle-is-named-oxford-dictionaries-word-of-the-year-2011-134361588.html>
- Squire, K. (2003). Video games in education. *International Journal of Intelligent Games & Simulation*, 2(1), 49-62.
- Stake, R. E. (1976). *Evaluating Educational Programmes: The Need and the Response*. París: Organisation for Economic Co-operation and Development (OECD).
- Stallings, W. (2005). *Sistemas operativos: Aspectos internos y principios de diseño* (5.ª ed.). Madrid: Pearson Ed.
- Stamper, R. (1996). Signs, information, norms and systems. En B. Holmqvist, P. B. Andersen, H. K. Klein y R. Posner (Eds.), *Signs of work: semiosis and information processing in organisations* (pp. 349-399). Berlin: Walter de Gruyter & Co.
- Stanchev, L. (2014). *Learning Java through games*. Boca Raton: CRC Press.
- StarLogo Nova. (s. f.). Recuperado de: <http://www.slnova.org/#info>
- Stroustrup, B. (1986). *The C++ Programming Language*. Reading: Addison-Wesley.

- Swacha, J. y Baszuro, P. (2013). Gamification-based e-learning platform for computer programming education. Presentado en X World Conference on Computers in Education, Toruń, Polonia. Recuperado de: <http://wce2013.umk.pl>
- Tanenbaum, A. S. (2009). *Sistemas operativos modernos*. (A. Vidal Romero Elizondo, Trad.) (3ra. Ed.). México: Pearson Education.
- Teixes, F. (2015). *Gamificación: fundamentos y aplicaciones*. Barcelona: UOC.
- The ACM-ICPC. (2017). Recuperado de: <http://icpc.baylor.edu>
- The College Board. (2017). Recuperado de: <http://www.collegeboard.org>
- Thomas International Ltd. (s. f.). Recuperado de: <http://www.thomasinternational.net/es-es/assessments/assessments-we-offer/teique>
- Thompson, S. (1999). *Haskell: The Craft of Functional Programming* (2.<sup>a</sup> ed.). Harlow: Addison-Wesley.
- TIS-100. (s. f.). Recuperado de: <http://www.zachtronics.com/tis-100>
- UNESCO. (1985). *Desarrollo de la informática en los sistemas de educación de países de América Latina y el Caribe*. Santiago de Chile: Oficina Regional de Educación de la UNESCO para América Latina y el Caribe.
- Unity. (s. f.). Recuperado de: <http://unity3d.com>
- Unreal Engine. (s. f.). Recuperado de: <http://www.unrealengine.com>
- Untrusted. (s. f.). Recuperado de: <http://alexnisnevich.github.io/untrusted>
- UVa Online Judge. (s. f.). Recuperado de: <http://uva.onlinejudge.org>
- Van Eck, R. (2006). Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless. *Educause Review*, 41(2), 16-30.
- Vardalas, J. N. (2001). *The Computer Revolution in Canada: Building National Technological Competence*. Cambridge: MIT Press.
- Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies*, 1(2), 42-64.
- Vegso, J. (2005). Interest in CS as a Major Drops Among Incoming Freshmen. *Computing Research News*, 17(3). Recuperado de: [http://cra.org/crn/2005/05/interest\\_in\\_cs\\_as\\_a\\_major\\_drops\\_among\\_incoming\\_freshmen](http://cra.org/crn/2005/05/interest_in_cs_as_a_major_drops_among_incoming_freshmen)

- Vera, P. M., Moreno, E. J. y Rodríguez, R. A. (2015). Los Mini-Juegos como herramienta para reforzar conceptos de programación en el ámbito Universitario. Presentado en III Congreso Internacional de Videojuegos y Educación, Buenos Aires.
- Vera, P. M., Moreno, E. J., Rodríguez, R. A., Vázquez, M. C. y Valles, F. E. (2016). Aplicación de Técnicas de Gamificación para la Enseñanza de Programación a Alumnos de Primer Año de Ingeniería. Presentado en XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016), Morón, Argentina. Recuperado de: <http://sedici.unlp.edu.ar/handle/10915/54653>
- Verry, T. (2013, abril 23). BeagleBone Black: A RasPi alternative with Arduino-like capabilities for \$45. *ExtremeTech*. Recuperado de: <http://www.extremetech.com/extreme/154100-beaglebone-black-raspberry-pi-and-arduino-on-a-single-board-for-just-45>
- Vidal Duarte, E. (2013). Reforzando el Primer Curso de Programación con Pygame: una librería gráfica para Python. En *Actas de las Jornadas Chilenas de Computación 2013*. Temuco, Chile.
- Voki. (s. f.). Recuperado de: <http://www.voki.com>
- Werbach, K. y Hunter, D. (2012). *For the win: how game thinking can revolutionize your business*. Philadelphia: Wharton Digital Press.
- What makes a game good? (1997, julio). *Next Generation*, 31, 40-49.
- Wilkinson, B., Williams, N. y Armstrong, P. (2013). Improving Student Understanding, Application and Synthesis of Computer Programming Concepts with Minecraft. En *The European Conference on Technology in the Classroom 2013*.
- Wilson, C., Sudol, L. A., Stephenson, C. y Stehlik, M. (2010). *Running On Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. New York: Association for Computing Machinery / Computer Science Teachers Association. Recuperado de: <http://runningonempty.acm.org/fullreport2.pdf>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wirth, N. (1971). The programming language Pascal. *Acta informatica*, 1(1), 35-63.

- Wirth, N. (1996). Recollections about the development of Pascal. En T. J. Bergin y R. G. Gibson (Eds.), *History of programming languages* (Vol. II, pp. 97-120). New York: ACM Press/Addison-Wesley.
- Wolber, D., Abelson, H., Spertus, E. y Looney, L. (2011). *App Inventor. Create your own Android apps*. Sebastopol, EE.UU.: O'Reilly Media, Inc.
- Wolf, M. J. P. (2001). Genre and the Video Game. En M. J. P. Wolf (Ed.), *The Medium of the Video Game* (pp. 113-136). Austin: University of Texas Press.
- Wolf, M. J. P. (Ed.). (2012). *Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming*. Santa Barbara: ABC-CLIO.
- Wolf, M. J. P. y Perron, B. (2003). *The Video Game Theory Reader*. New York: Routledge, Taylor & Francis Group, Inc.
- Wriston, W. B. (2007). *Bits, bytes, and balance sheets: the new economic rules of engagement in a wireless world* (Vol. 557). Stanford: Hoover Institution Press.
- X3DOM. (s. f.). Recuperado de: <http://doc.x3dom.org>
- Zabala, G., Pérez Cerrato, L., Blanco, S., Morán, R. y Teragni, M. (2016). Minecraft Programable: una herramienta para aprender programación en nivel medio. *Virtualidad, Educación y Ciencia*, 7(12), 113-124.
- Zichermann, G. y Cunningham, C. (2011). *Gamification by design: Implementing game mechanics in web and mobile apps*. O'Reilly Media, Inc.
- Zimmerman, E. (2013, julio). Gaming literacy: Game design as a model for literacy in the twenty-first century. *Intersemiose*, II(4), 155-165.
- Zorn, C., Wingrave, C. A., Charbonneau, E. y LaViola Jr, J. J. (2013). Exploring Minecraft as a conduit for increasing interest in programming. En *FDG* (pp. 352-359).
- Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), 25-32.



# ANEXOS

---



## A1. TEIQue (TRAIT EMOTIONAL INTELLIGENCE QUESTIONNAIRE)

Apellidos y nombre:

Sexo: V M

Edad:

Fecha: / /



### Instrucciones

- Por favor, responda cada una de las afirmaciones expuestas más abajo poniendo un círculo alrededor del número que mejor refleja su grado de acuerdo o desacuerdo con cada afirmación.
- Lea cada afirmación y escoja la respuesta que se corresponda mejor con su manera más frecuente de ser, pensar o actuar. NO piense demasiado sobre el significado exacto de la afirmación. Responda con sinceridad. NO hay respuestas correctas o incorrectas. Hay siete posibles respuestas a cada afirmación, variando desde “Completamente en Desacuerdo” (nº 1) hasta “Completamente de Acuerdo” (nº 7).

		DESACUERDO COMPLETAMENTE	1	2	3	4	5	6	7	ACUERDO COMPLETAMENTE
1.	Soy capaz de manejar los sentimientos de los demás	1	2	3	4	5	6	7		
2.	En general, no me doy cuenta de las emociones y sentimientos que están sintiendo los demás	1	2	3	4	5	6	7		
3.	Cuando recibo muy buenas noticias me cuesta calmarme	1	2	3	4	5	6	7		
4.	Tiendo a ver dificultades en cada oportunidad más que oportunidades en cada dificultad	1	2	3	4	5	6	7		
5.	Soy pesimista en la mayoría de las cosas	1	2	3	4	5	6	7		
6.	No tengo muchos recuerdos felices	1	2	3	4	5	6	7		
7.	Creo que comprendo bien las necesidades y deseos de los demás	1	2	3	4	5	6	7		
8.	En general creo que las cosas me irán bien en la vida	1	2	3	4	5	6	7		
9.	Me cuesta reconocer mis propias emociones	1	2	3	4	5	6	7		
10.	No tengo habilidades sociales	1	2	3	4	5	6	7		
11.	Me cuesta trabajo decirle a otra persona que la quiero, incluso cuando deseo hacerlo	1	2	3	4	5	6	7		
12.	Algunas personas me admiran por ser tan tranquilo/a	1	2	3	4	5	6	7		
13.	Casi nunca pienso en viejos amigos del pasado	1	2	3	4	5	6	7		
14.	Generalmente me resulta fácil decir a otras personas cuánto significan para mí	1	2	3	4	5	6	7		
15.	Habitualmente debo estar bajo presión para, de verdad, trabajar duro	1	2	3	4	5	6	7		
16.	Me implico, sin pensar lo suficiente, en cosas que más tarde desearía poder dejar	1	2	3	4	5	6	7		
17.	Puedo “leer” los sentimientos de la gente como en un libro abierto	1	2	3	4	5	6	7		
18.	Soy capaz de influir en los sentimientos de los demás	1	2	3	4	5	6	7		
19.	Normalmente me resulta difícil calmar a personas que están enfadadas	1	2	3	4	5	6	7		
20.	Cuando hay algún problema o discusión en casa me resulta difícil mantener el control de la situación	1	2	3	4	5	6	7		
21.	Generalmente espero lo mejor en la vida	1	2	3	4	5	6	7		
22.	Algunas personas me han dicho que me admiran por mi honestidad	1	2	3	4	5	6	7		
23.	No me gusta escuchar los problemas de mis amigos/as	1	2	3	4	5	6	7		
24.	Soy capaz de “ponerme en la piel” de los demás y sentir sus emociones	1	2	3	4	5	6	7		
25.	Creo que estoy lleno/a de defectos	1	2	3	4	5	6	7		
26.	Me cuesta mucho trabajo abandonar u olvidarme de las cosas que me gustan	1	2	3	4	5	6	7		
27.	Siempre encuentro modos de expresar mi afecto hacia otras personas cuando quiero hacerlo	1	2	3	4	5	6	7		
28.	Creo que poseo buenas cualidades	1	2	3	4	5	6	7		
29.	Tiendo a precipitarme en lo que hago sin pararme a pensar antes	1	2	3	4	5	6	7		
30.	Me resulta difícil hablar sobre mis sentimientos íntimos, incluso con mis amigos más cercanos	1	2	3	4	5	6	7		
31.	No soy capaz de hacer las cosas tan bien como la mayoría de la gente	1	2	3	4	5	6	7		
32.	Nunca estoy realmente seguro/a de lo que estoy sintiendo	1	2	3	4	5	6	7		
33.	Normalmente soy capaz de expresar mis emociones cuando lo deseo	1	2	3	4	5	6	7		
34.	Cuando estoy en desacuerdo con alguien me resulta fácil decirselo	1	2	3	4	5	6	7		
35.	Me cuesta motivarme por lo que hago	1	2	3	4	5	6	7		
36.	Sé cómo animarme cuando me siento mal	1	2	3	4	5	6	7		
37.	En general me resulta difícil describir mis sentimientos	1	2	3	4	5	6	7		
38.	Me resulta difícil no entristecerme cuando alguien me habla de algo malo que le ha ocurrido	1	2	3	4	5	6	7		
39.	Cuando algo me sorprende bruscamente me resulta difícil dejar de pensar en ello	1	2	3	4	5	6	7		
40.	A menudo me detengo a pensar sobre mis sentimientos	1	2	3	4	5	6	7		
41.	Tiendo a ver el vaso “medio vacío” en vez de “medio lleno”	1	2	3	4	5	6	7		
42.	A menudo me resulta difícil ver las cosas desde el punto de vista de otra persona	1	2	3	4	5	6	7		

© K.V. Petrides (2001), <http://www.psychometriclab.com>. Adaptación española de J.C. Pérez-González (2002), <http://www.eduomo.com>  
E-mail: [jcperez@edu.uned.es](mailto:jcperez@edu.uned.es)

		DESACUERDO COMPLETAMENTE				ACUERDO COMPLETAMENTE		
43.	Soy un seguidor, no un líder	1	2	3	4	5	6	7
44.	Las personas de mi entorno más cercano se quejan de que no les trato bien	1	2	3	4	5	6	7
45.	Muchas veces no consigo tener claro qué emoción estoy sintiendo	1	2	3	4	5	6	7
46.	No consigo influir sobre los sentimientos de otras personas	1	2	3	4	5	6	7
47.	Si estoy celoso/a de alguien me cuesta ocultarlo	1	2	3	4	5	6	7
48.	Me pongo nervioso/a por situaciones que los demás consideran llevaderas	1	2	3	4	5	6	7
49.	Me cuesta comprender el dolor de quienes lo están pasando mal por algún motivo	1	2	3	4	5	6	7
50.	Alguna vez me he aprovechado injustamente del trabajo de otras personas	1	2	3	4	5	6	7
51.	Afronto los cambios en mi vida con facilidad y de manera eficaz	1	2	3	4	5	6	7
52.	No creo tener ningún poder sobre los sentimientos de los demás	1	2	3	4	5	6	7
53.	Ante una tarea difícil no abandono fácilmente	1	2	3	4	5	6	7
54.	Me gusta esforzarme incluso en las cosas que no son especialmente importantes	1	2	3	4	5	6	7
55.	Siempre asumo la responsabilidad cuando hago algo mal	1	2	3	4	5	6	7
56.	Tiendo a cambiar de opinión frecuentemente	1	2	3	4	5	6	7
57.	Cuando discuto con alguien solamente puedo ver mi punto de vista	1	2	3	4	5	6	7
58.	Las cosas, al final, tienden a salir bien	1	2	3	4	5	6	7
59.	Cuando estoy en desacuerdo con alguien generalmente prefiero callarme antes que crear una disputa	1	2	3	4	5	6	7
60.	Si quisiera me resultaría fácil hacer sentir mal a alguien	1	2	3	4	5	6	7
61.	Creo que soy una persona tranquila	1	2	3	4	5	6	7
62.	A menudo siento dificultad para mostrar mi afecto a las personas más allegadas	1	2	3	4	5	6	7
63.	Pienso que hay muchas razones para esperar lo peor en la vida	1	2	3	4	5	6	7
64.	Me resulta difícil expresarme con claridad	1	2	3	4	5	6	7
65.	No me importa cambiar mi rutina diaria con cierta frecuencia	1	2	3	4	5	6	7
66.	La mayoría de la gente suele gustar o agradar más que yo	1	2	3	4	5	6	7
67.	Quienes son cercanos a mí no suelen quejarse de mi comportamiento hacia ellos	1	2	3	4	5	6	7
68.	Tengo dificultad para expresar mis emociones tal y como a mí me gustaría	1	2	3	4	5	6	7
69.	Por lo general soy capaz de adaptarme a nuevas situaciones	1	2	3	4	5	6	7
70.	Me cuesta trabajo adaptarme a los cambios	1	2	3	4	5	6	7
71.	Me considero un/a buen/a negociador/a	1	2	3	4	5	6	7
72.	Puedo relacionarme fácilmente con la gente	1	2	3	4	5	6	7
73.	En general soy una persona con alta motivación	1	2	3	4	5	6	7
74.	He robado cosas cuando era niño/a	1	2	3	4	5	6	7
75.	En general estoy encantado/a con mi vida	1	2	3	4	5	6	7
76.	Me cuesta controlarme cuando me siento muy feliz	1	2	3	4	5	6	7
77.	No me cuesta trabajo esforzarme ante una tarea	1	2	3	4	5	6	7
78.	Cuando tomo una decisión siempre estoy seguro de que es la más acertada	1	2	3	4	5	6	7
79.	Si fuera a una "cita a ciegas", a la otra persona le defraudaría mi aspecto	1	2	3	4	5	6	7
80.	Normalmente me cuesta adaptar mi comportamiento a las personas con las que estoy	1	2	3	4	5	6	7
81.	Soy capaz de identificarme con los pensamientos y sentimientos de otras personas	1	2	3	4	5	6	7
82.	Intento controlar las presiones para así manejar mis niveles de ansiedad	1	2	3	4	5	6	7
83.	No me considero un/a inútil	1	2	3	4	5	6	7
84.	Me cuesta controlar mis emociones	1	2	3	4	5	6	7
85.	Afronto la mayoría de las dificultades en mi vida de una manera tranquila y serena	1	2	3	4	5	6	7
86.	Si quisiera me resultaría fácil hacer enfadar a alguien	1	2	3	4	5	6	7
87.	En general, me gusta	1	2	3	4	5	6	7
88.	Creo que estoy lleno/a de virtudes	1	2	3	4	5	6	7
89.	En general no encuentro la vida agradable	1	2	3	4	5	6	7
90.	Suelo calmarme con rapidez después de haberme enfadado con alguien	1	2	3	4	5	6	7
91.	Puedo mantener la calma incluso cuando me siento muy feliz	1	2	3	4	5	6	7
92.	No se me da bien consolar a los demás cuando se sienten mal	1	2	3	4	5	6	7
93.	Soy capaz de resolver conflictos o disputas entre la gente	1	2	3	4	5	6	7
94.	Nunca antepongo el placer al trabajo	1	2	3	4	5	6	7
95.	Me resulta fácil ponerme en el lugar de otra persona para comprenderla	1	2	3	4	5	6	7
96.	A menudo tengo problemas por no controlarme	1	2	3	4	5	6	7
97.	Me es fácil encontrar las palabras apropiadas con las que describir mis sentimientos	1	2	3	4	5	6	7

2

		DESACUERDO COMPLETAMENTE					ACUERDO COMPLETAMENTE	
98.	Creo que la mayor parte de mi vida será agradable	1	2	3	4	5	6	7
99.	Soy una persona corriente	1	2	3	4	5	6	7
100.	Tiendo a enfadarme o a excitarme con facilidad	1	2	3	4	5	6	7
101.	Normalmente intento evitar pensamientos desagradables y pensar en alternativas positivas	1	2	3	4	5	6	7
102.	No me gusta planificar las cosas	1	2	3	4	5	6	7
103.	Puedo comprender lo que siente una persona sólo con mirarla	1	2	3	4	5	6	7
104.	Creo que la vida es hermosa	1	2	3	4	5	6	7
105.	Me resulta fácil calmarme después de haber sido asustado/a	1	2	3	4	5	6	7
106.	Me gusta estar al mando de las cosas	1	2	3	4	5	6	7
107.	Me cuesta cambiar las opiniones de los demás	1	2	3	4	5	6	7
108.	Se me dan bien las charlas informales sin importancia, como en las salas de espera o en los ascensores con personas desconocidas	1	2	3	4	5	6	7
109.	Puedo controlar mis impulsos	1	2	3	4	5	6	7
110.	En realidad, no me gusta mi aspecto físico	1	2	3	4	5	6	7
111.	Tiendo a hablar bien y claramente	1	2	3	4	5	6	7
112.	Veó que no afronto bien el estrés	1	2	3	4	5	6	7
113.	La mayoría de las veces sé exactamente por qué siento lo que estoy sintiendo en ese momento	1	2	3	4	5	6	7
114.	Encuentro dificultad para calmarme después de haber recibido una gran sorpresa	1	2	3	4	5	6	7
115.	Me describiría a mí mismo/a como una persona capaz de defender mis derechos con firmeza y sin vacilación	1	2	3	4	5	6	7
116.	En general, no soy una persona feliz	1	2	3	4	5	6	7
117.	Cuando alguien me ofende soy capaz de mantener la calma	1	2	3	4	5	6	7
118.	La mayoría de las cosas que trato de hacer bien parecen exigirme mucho esfuerzo	1	2	3	4	5	6	7
119.	Nunca he mentido para no herir los sentimientos de otra persona	1	2	3	4	5	6	7
120.	Me cuesta conectar con las personas, incluso con aquéllas más cercanas a mí	1	2	3	4	5	6	7
121.	Antes de tomar una decisión tengo en cuenta las ventajas y desventajas	1	2	3	4	5	6	7
122.	No sé cómo hacer sentir mejor a los demás cuando lo necesitan	1	2	3	4	5	6	7
123.	Normalmente me resulta difícil cambiar mis actitudes y puntos de vista	1	2	3	4	5	6	7
124.	No suelo hablar sobre cómo me siento	1	2	3	4	5	6	7
125.	En general estoy satisfecho/a con mis relaciones más personales	1	2	3	4	5	6	7
126.	Puedo identificar mis emociones desde el momento en que comienzan	1	2	3	4	5	6	7
127.	En general, me agrada anteponer los intereses de la gente a los míos propios	1	2	3	4	5	6	7
128.	Me siento a gusto de estar vivo/a	1	2	3	4	5	6	7
129.	Tiendo a obtener una gran satisfacción simplemente con hacer algo bien	1	2	3	4	5	6	7
130.	Para mí es muy importante mantener el contacto con mis mejores amigos y con mi familia	1	2	3	4	5	6	7
131.	Habitualmente tengo pensamientos alegres	1	2	3	4	5	6	7
132.	Tengo muchas discusiones con mis familiares y amigos/as	1	2	3	4	5	6	7
133.	No tengo dificultad para expresar mis emociones con palabras	1	2	3	4	5	6	7
134.	Me cuesta disfrutar de la vida	1	2	3	4	5	6	7
135.	Soy capaz de influir sobre otras personas	1	2	3	4	5	6	7
136.	Cuando estoy bajo presión tiendo a perder la calma	1	2	3	4	5	6	7
137.	Me cuesta cambiar mi comportamiento	1	2	3	4	5	6	7
138.	Los demás se fijan en mí	1	2	3	4	5	6	7
139.	Algunos me dicen que me pongo nervioso/a muy fácilmente	1	2	3	4	5	6	7
140.	Puedo encontrar diferentes maneras de controlar mis emociones cuando lo deseo	1	2	3	4	5	6	7
141.	Creo que yo sería un/a buen/a vendedor/a	1	2	3	4	5	6	7
142.	Suelo perder el interés en lo que hago	1	2	3	4	5	6	7
143.	En general, soy un esclavo de mis hábitos y costumbres	1	2	3	4	5	6	7
144.	Normalmente defiendiendo mis opiniones incluso si tengo que discutir con gente importante	1	2	3	4	5	6	7
145.	Me considero una persona flexible	1	2	3	4	5	6	7
146.	Necesito que me estimulen para implicarme plenamente en lo que hago	1	2	3	4	5	6	7
147.	Incluso cuando estoy discutiendo con alguien puedo adoptar su perspectiva	1	2	3	4	5	6	7
148.	En general soy capaz de afrontar situaciones estresantes	1	2	3	4	5	6	7
149.	Intento evitar a las personas que pueden estresarme o ponerme nervioso/a	1	2	3	4	5	6	7
150.	A menudo me dejo llevar por un capricho sin pensar en las consecuencias	1	2	3	4	5	6	7
151.	En una discusión tiendo a ceder incluso cuando sé que estoy en lo cierto	1	2	3	4	5	6	7
152.	Cuando hay algún problema o discusión en el trabajo me resulta difícil mantener el control de la situación	1	2	3	4	5	6	7
153.	Algunas de mis respuestas a este cuestionario no son 100% sinceras	1	2	3	4	5	6	7



## A2. TRANSCRIPCIÓN DE LAS ENTREVISTAS CON LOS ESTUDIANTES

### ESTUDIANTES QUE CURSARON PROGRAMACIÓN II SIN ESTRATEGIAS LÚDICAS (TRADI)

#### TRADI-01 (Docente, 34 años, hombre)

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Fueron parte de los videos de explicaciones de programación en Java. Eran... Lo bueno de eso era que se podía volver para atrás o para adelante dependiendo de la velocidad para aprender.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *De Programación II, eh... La parte de hacer las partes... La parte gráfica, de las ventanas, botones. Esas partes.*

Entrevistador: ¿Pensás que sería posible cursar esta materia utilizando juegos? ¿Cómo? ¿Tomaría mucho tiempo y esfuerzo?

Entrevistado: *Y... Haciendo... Estudiándola con videojuegos sería igual —para mí— como ver videos. Eh... Tendría que tener —obviamente— algún puntaje, alguna manera para ir viendo las marcas que uno va... que va logrando. O sea, para ir sabiendo el logro que uno va alcanzando en el videojuego.*

#### TRADI-02 (Docente / Programador, 28 años, hombre)

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Eh... De los recursos que vimos en Programación II, creo que lo que más me impactó fue lo de... Los patrones de diseño que se vieron. Fue como una cosa que no conocía y que... O que había sentido nombrar en algún momento, pero nunca lo había visto en detalle.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Es difícil esa... A ver... Esperá... Un segundo... Eh... Creo que... Lo que más llego a dominar es... Eh... El tema de herencia y polimorfismo. Eso creo que me quedó bastante claro y... Y me pareció muy útil.*

Entrevistador: ¿Pensás que sería posible cursar esta materia utilizando juegos? ¿Cómo? ¿Tomaría mucho tiempo y esfuerzo?

Entrevistado: *Yo creo que sí, que... Es posible hacer algo así, pero... Sí, llevaría mucho tiempo y esfuerzo, y el cómo es complicado de resolver, porque es... Habría que plantearlo bien, el tema de... Eh... Cómo sería el juego, pero yo creo que sí, que sería un buen recurso para... Para aprender cualquier tema.*

### **TRADI-03 (No trabaja, 21 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Eh... El recurso de... MinGW. Eh... Porque me mencionaron que era un... Es un recurso que se usa en tercer año y ya empezamos a verlo en segundo año.*

Entrevistador: ¿No será LibGDX?

Entrevistado: *Eh... Sí, es. Perdón... Sí, se me mezcló...*

Entrevistador: MinGW es el compilador de C, de primer año... Bueno... ¿Y LibGDX ya lo usan en segundo?

Entrevistado: *Claro, sí.*

Entrevistador: Bueno... ¿Y qué es?

Entrevistado: *Es una aplicación, eh... Que sirve para... Para hacer —tipo— programas para Android y otras... Y otros... Eh...*

Entrevistador: Y para hacer juegos...

Entrevistado: *Claro, sí.*

Entrevistador: Buenísimo... Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?



Entrevistado: *Eh... Y, creo que el... ¿Cómo se llama? Eh... Bue, me olvidé el nombre, de esto último que vimos. ¿Cómo se llamaba?*

Entrevistador: ¿Para qué sirve?

Entrevistado: *No, el de... El que se divide en varias carpetas en NetBeans. Se me fue...*

Entrevistador: ¿Y para qué sirve? ¿Gestión de versiones? ¿Pruebas unitarias?

Entrevistado: *Eh... Que es con objetos... Objetos y... ¡MVC!*

Entrevistador: Modelo-Vista-Controlador. Buenísimo... ¿Pensás que sería posible cursar esta materia utilizando juegos? ¿Cómo? ¿Tomaría mucho tiempo y esfuerzo?

Entrevistado: *Eh... No es imposible, pero hay que prestarle bastante atención y tener algo de tiempo para ejecutarlo...*

#### **TRADI-04 (Docente, 24 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *¿Qué tema fue el que más me atrapó, el que más me gustó? Fue el hecho de la programación por eventos. Eh... Siempre fue algo que... que me interesó, ver... Saber cómo una máquina reaccionaba ante las interacciones del usuario, ya sea mover un 'mouse', tocar un botón, cómo hacían los juegos para discriminar qué funcionalidad hacía cada botón, eh... Etcétera. Y siempre es algo que con la programación estructurada no podía lograr, y que —cuando recién tenía mis primeros conocimientos de objetos— no se me ocurría otra forma que encerrar cosas en ciclos infinitos esperando que algo suceda, hasta que luego vi programación dirigida ya por eventos, con los 'listeners', con los manejadores, los 'handlers' y demás. Y bueno, eso ya me abrió un panorama mucho más... Mucho más rico, en cuanto a la cantidad de cosas que se pueden hacer, ya sea juegos, ya sea aplicaciones eh... de cualquier índole.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Eh... Es algo que me cuesta responder, porque —obviamente— uno nunca deja de... De capacitarse y de hilar más fino en cuanto a los... A las herramientas, ¿no? Pero me parece que... Que antes de entrar —y si bien tengo dominado y entendido el concepto de herencia, de polimorfismo, de métodos abstractos— creo que lo que más intento... Domino e intento seguir dominando... Es entender las responsabilidades, la modularización de cada objeto, ¿sí? Que cada objeto tenga funcionalidades bien definidas, que sea un ente —¿no?— bien cerrado, con sus responsabilidades que nadie... que ningún otro objeto debe saber cómo son, y que se tengan que comunicar entre sí mediante mensajes, que es lo que reza —en realidad— el paradigma, antes de meterse —como digo— en cualquier otro concepto. Así que —bueno— eso es lo que más domino —creo— a esta altura. Y bueno, será cuestión de ir viendo después... Metiéndose más de lleno en lo que son patrones de diseño —¿no?— como para tener algunas cosas mejor diagramadas, pero... Pero eso es lo que mejor me... Me queda, por ahora.*

Entrevistador: ¿Pensás que sería posible cursar esta materia utilizando juegos? ¿Cómo? ¿Tomaría mucho tiempo y esfuerzo?

Entrevistado: *Eh... Es algo que... Que a mí me tiene con bastante entusiasmo preguntármelo, porque al ser yo también docente, estar capacitándome también para ello, uno también busca distintas... Eh... Metodologías de... De enseñanza, novedosas, que apliquen a las nuevas tecnologías tradicionales y a los nuevos paradigmas del siglo XXI en cuanto a la tecnología... Así que creo que se podría abordar tranquilamente la enseñanza mediante vi... Juegos, dado que... Eh... La mayoría de los juegos que yo he jugado —obviamente— persiguen —ya al conocer uno el tema— la programación orientada a objetos. Por lo que... Eh... Inclusive... Eh... Es hasta intuitivo. Yo he notado que —al hacer juegos— las cosas... Eh... Se resuelven, se piensan, se ocurren las soluciones a los problemas de manera más intuitiva que —quizás— yendo a algún programa... A algún problema, perdón, más de índole matemático o lógico, buscando solucionar cosas de... No sé, empresariales, ¿sí? Siempre los juegos, como... Como todo el mundo ha tenido infancia, ha jugado a los juegos más típicos... Eh... Las soluciones... Y*

*los juegos justamente son los que más lógica requieren de fondo. Creo que es la mejor manera de abordar la parte lógica del alumno y —encima— trasladarlo a objetos porque —digamos— todo juego consta de objetos, porque inclusive aquellos que son en su mayoría juegos conceptuales, ¿no? Que no son físicos, sino que son juegos más de... ¿Qué se yo? 'Piedra, papel o tijera'. Por así decirlo, es un juego que no tiene objetos —físicamente hablando— pero sabemos muy bien que —en la programación orientada a objetos— no solamente lo físico es un objeto sino también lo abstracto. Entonces, nada. Todo sirve. Creo que los juegos son una excelente herramienta. Hay que —obviamente— estudiar cuál es la mejor manera de implementarlos. Tal vez el tema de la 'gamification'. Que es hacer que una cursada sea dada solamente por juegos, dándoles a los alumnos en vez de notas en un examen tal vez medallas, premios, quitándoles vidas si fallan o si no cumplen con plazos de entrega. Me parece que... Que abordaría la enseñanza desde un punto de vista más dinámico, más ligado a lo que a la gran mayoría de nosotros nos gusta, que es jugar. Y, por supuesto, nunca dejando de aprender, que creo que es el foco de lo que todos —como docentes— buscamos. Así que nada, creo que... Que es una propuesta atractiva para... Para aplicar.*

**TRADI-05 (Programador, 24 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Bueno... Entre los recursos más importantes a lo largo de la cursada, me pareció muy bueno el tema de los programas de ejemplo. Algunos pequeños programas, por ejemplo, para explicar polimorfismo. O la herencia. O sino también con problemas de... Que atravesamos en la vida cotidiana. Programas para administrar los turnos de los hoteles, de los vuelos, o tal vez de transporte. Creo que me pareció muy importante el tema de plantear problemas de la realidad, que sean del día a día, y tratar de resolverlos con... Utilizando las herramientas de programación.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Entre los conceptos aprendidos a través de la cursada de Programación II, creo que reforcé mucho mis conocimientos... Conceptos sobre lo que son los objetos, la herencia, la abstracción. Creo que ref... Pude reforzar mucho mis conocimientos de programación orientada a objetos y de la programación en sí.*

Entrevistador: ¿Pensás que sería posible cursar esta materia utilizando juegos? ¿Cómo? ¿Tomaría mucho tiempo y esfuerzo?

Entrevistado: *Lo veo como algo muy positivo, algo que nos ayudaría mucho a entender temas complejos de una manera más rápida y sencilla, mediante la interacción con algún juego, la visualización de ejemplos. Pienso que ahorraría mucho más tiempo y quizás simplificaría algunos temas.*

#### **TRADI-06 (System engineer / Desarrollador, 25 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Okay. Eh... De los recursos usados, este... Lo más impresionante tal vez fue —a mi parecer— la documentación provista. La verdad que fue muy... Muy pedagógica y el acompañamiento del docente, este... A mí me resultó muy... Muy potable. Fueron —tal vez— dos de los puntos que a mí me... Me empujaron a seguir con la materia, este... Y a estudiar —incluso— durante horarios de trabajo, cuando tenía tiempos libres... Cosa que comúnmente no hago. Este... Digamos, y a poder darle todo a la materia para poder aprenderla. Realmente me llevé... Me llevé mucho conocimiento, gracias a esos dos puntos particulares, realmente.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Hoy día, este... Me llevé mucho, en cuanto a Programación II, dominar la sintaxis de los lenguajes que me enseñaron, este... Las estructuras, este... De datos de esos lenguajes. El cual me permitió, hoy día, este... Programar tal vez*

*un poquito más eficientemente o, tal vez, con mayor conciencia de... De lo que estoy escribiendo —digamos— y que sea eficiente para... Para el requerimiento que me están dando.*

Entrevistador: *¿Pensás que sería posible cursar esta materia utilizando juegos? ¿Cómo? ¿Tomaría mucho tiempo y esfuerzo?*

Entrevistado: *Eh... En mi enfoque personal, creo que sí, sería este... Muy posible, tal vez, dar la materia... Eh... Usando juegos. Creo que es una... Un método que apoyaría mucho porque hoy día, este... Genera más, este... Inmersión de parte del alumno con... Con la materia y con el docente. Entonces ayuda a mantener al alumno —digamos— dentro de la cursada. Este... Evita la frustración, más... Más que nada. Este... Así que creo que sí. No creo... No sé si tomaría poco tiempo, igualmente, o sea, para adoptar una... Una metodología basada alrededor de los juegos, creo que tomaría un... Un tiempo como para diseñarlo bien, como para que sea atractivo para toda la clase de... De público que esté en el aula.*

## **ESTUDIANTES QUE DESARROLLARON VIDEOJUEGOS EN PROGRAMACIÓN II (DESVJ)**

### **DESVJ-01 (Java developer, 19 años, desarrolló *Fighting Questions Club*, hombre)**

Entrevistador: *Mirando hacia atrás, ¿cuál de los proyectos desarrollados en Programación II te impresionó más? ¿Por qué?*

Entrevistado: *Bue... Muchos hablan del videojuego. A mí me gustó. Yo pensé que eso iba a ser lo más impactante, por ahí, del año. Empezar algo desde cero y, encima, algo nuevo... Pero, a mí, en lo particular, me gustó más el proyecto de fin de año que hicimos, que se trató de hacer un sistema completo, funcional, desde cero. Nada, es algo que es difícil pensarlo y, al ser mi primera vez, me gustó poder enfrentarme con eso y que no sea en el trabajo, que sea antes. Estuvo bueno.*

Entrevistador: *Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?*

Entrevistado: *Bue... Vimos un poco de bases de datos, que —bah— vimos bastantes clases. Habremos visto tres o cuatro clases. Para mí, yo me siento más fuerte en eso*

*justamente porque puedo ir mezclando con lo que ya sé de SQL, pero además de eso, algo que estuvo muy bueno verlo es el tema referido a patrones de diseño, que le da cierta agilidad cuando te tenés que enfrentar a un problema. No es lo mismo enfrentarlo como yo lo hacía hace un año que como ahora. Tengo un poco más... Tengo más herramientas para hacerlo.*

Entrevistador: Volvamos al proyecto del videojuego, ¿te gustó desarrollarlo con tu grupo? ¿Le dedicaron mucho tiempo y esfuerzo? ¿Podrías decirme más sobre cómo administraron el proceso? ¿Qué los motivó?

Entrevistado: *Bue... Para empezar, le dediqué muchísimo tiempo. Lo que más me motivó, y supongo que a mi compañero también, fue hacer un juego, ¿no? Nunca me había imaginado en la facu poder llegar a hacerlo. Sí, desde chiquito me fascinaba el tema de los juguitos y siempre quería hacer uno. Cruzarse con esto acá fue bastante bueno. Después... ¿Cómo administré la parte del desarrollo? Fue algo que me costó al principio, pero pude tomar la mejor elección, que fue dividir en partes que estén desacopladas. Por ejemplo, a mí me tocó algo de pelea: una parte bien clara es la gráfica, otra parte bien clara es —yo qué sé— el sistema de puntuación. Cosas donde no tengamos que depender del otro... Nada, también es lo que se viene, se está hablando mucho de ‘microservices’, que tienen que ver con eso de separar en partes las aplicaciones para un desarrollo más ágil.*

### **DESVJ-02 (Desarrollador .NET, 23 años, desarrolló *Road of Destiny*, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los proyectos desarrollados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Bueno, creo que —de los proyectos desarrollados en Programación II— el del videojuego obviamente fue el que más me impresionó, el que más —de hecho— el más novedoso para mí, más allá de que... Bueno, tengo cierta experiencia en lo que es el desarrollo —de programación orientada a objetos y eso— yo, más que nada, trabajo igualmente en C#, así que Java para mí igual era algo que podía aprender y que no tenía un conocimiento sólido. Pero bueno, lo de los videojuegos, además de aprender un poco de Java me mostró cómo se trabaja con*

*'frameworks', basados en desarrollo de videojuegos. Así que creo que elijo ese, ¿no? Como el que más me impresionó. Bueno, ahí respondo un poco también el porqué, ¿no? Porque no había trabajado nunca con un 'framework' para el desarrollo de videojuegos y tampoco... A veces, me había inclinado a hacer algunos juegos muy chiquitos, pero usando... No usando 'frameworks', sino, nada... Usando otro tipo de formas y de maneras. Más... que se me ocurrían, más personales.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Bueno, de los conceptos aprendidos o vistos en Programación II, los que más domino quizás sea lo que es la programación orientada a objetos, el tema de polimorfismos, herencias. También porque es un tema que me apasiona. Ya lo sabía igual de antes, en cierta forma, pero creo que el que más domino es la programación orientada a objetos en sí.*

Entrevistador: Volvamos al proyecto del videojuego, ¿te gustó desarrollarlo con tu grupo? ¿Le dedicaron mucho tiempo y esfuerzo? ¿Podrías decirme más sobre cómo administraron el proceso? ¿Qué los motivó?

Entrevistado: *Creo que estuvo bueno desarrollarlo en grupo. Solo hubiera sido bastante complicado. No hubiera sido lo mismo. Y... Nada, aparte lo hicimos con amigos, con personas que nos llevábamos bien, así que, nada, pudimos hacerlo bien. Y sí, le dedicamos mucho tiempo, mucho esfuerzo. De hecho, un poco, no me recuerdo ahora, pero me acuerdo de haber estado programando —sí, si no me equivoco— el mismo día... el día antes de la entrega, estar desarrollándolo hasta las cinco de la mañana. No porque no le hayamos metido de antes, ¿no? Pero sí porque fue largo, fue extenso, bueno, y a lo último se habían quedado algunos bugs dando vueltas, que lo resolvimos a último día... Y sí, me acuerdo de haber estado ese mismo día hasta muy tarde, así que nos llevó, nos llevó un tiempo largo... Y bueno, obviamente con mucho esfuerzo también. No es que nos tiramos a no hacer nada, pero sí, fue... Costó. Bueno, y después, respecto a cómo administramos el proceso... No, no recuerdo exactamente ahora cómo lo hicimos,*

*pero sí me acuerdo que dividimos en varias clases, entonces pudimos... Como uno se dedicaba más al... Me acuerdo que había una clase... En nuestro caso era un juego de autos de carreras... Bueno, y había una clase que era Ruta, una clase que era Auto, y nada, como que pudimos abocarnos en lo que son las distintas clases de manera más personal, uno se abocó más a la ruta, otro más al auto, y bueno, y después las partes en común sí, las íbamos tocando, íbamos viendo, por ahí lo íbamos probando y si encontrábamos algún bug lo resolvíamos. Pero sí quizás la manera en que nos administramos fue —más que nada— usando clases, ¿no? O sea, cada uno trataba de tocar su clase. Y bueno, las clases eran bastante inteligentes, como para poder dominar bien... para poder hacer bien lo que tenían que hacer y ser bastante específicas. Y bueno, lo que nos motivó a hacer el juego en sí, el juego de carreras... En mi caso, yo había jugado una vez —me acuerdo— en la primaria... Había un juego de carreras que alguien iba... el jugador iba de... por ejemplo, de Buenos Aires a Córdoba, y tenía que ir haciendo kilómetros, y esos kilómetros los hacía respondiendo preguntas. Y bueno, si respondías preguntas... Podías parar en estaciones de servicio y algunas cosas... Tenía muchas cosas... No sé por qué ese juego me gustaba, me marcó como una etapa de mi infancia y bueno, cuando tuvimos la posibilidad de hacer un juego, nuestro juego estaba basado un poco también en eso porque te iba haciendo preguntas y, bueno, nada, esas preguntas iban... Al responder las preguntas ibas ganando plata y después podías parar en gomerías y en distintas cosas, y con la plata que ganabas respondiendo bien las preguntas podías comprar ruedas de auxilio, porque después en el camino podías chocar y podías pinchar la rueda, entonces —nada— ibas comprando. Creo que el juego estaba basado un poco en ese juego que jugué en la primaria, que jugaba en Computación, y nada, me llamó la atención y —nada— quise... quisimos hacer un poco algo parecido a ese juego que algo... algo tenía porque realmente fue algo que marcó una etapa importante de mi primaria. Así que nada, eso.*



**DESVJ-03 (Desarrollo y soporte, 43 años, desarrolló *El Escuerzo Erudito*, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los proyectos desarrollados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Bueno, en realidad, el segundo es el que más... El segundo, el que involucra el segundo parcial, porque es en el que más estuve involucrado y porque, digamos que entendí más los conceptos de orientación a objetos. Para entonces. En el caso del primero, no estuve tan involucrado y teníamos una persona que dominaba más el tema y, entonces, es como que fue un periodo de aprendizaje recién.*

Entrevistador: ¿Y cuál era el asunto del segundo, del que más te impresionó?

Entrevistado: *En realidad, el problema en sí. Estaba muy bueno porque me parece que abarca todo, el planteo —en sí— del problema y hay que desarrollar prácticamente un sistema, si bien chiquito, pero que abarca prácticamente todas las funciones.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Lo que yo manejo mejor es bases de datos. Ahí es donde yo me siento bastante más cómodo. Tiene que ver no solo por lo aprendido ahora sino por —¿cómo es, eh?— un poco los conocimientos previos, ¿no?*

Entrevistador: Volvamos al proyecto del videojuego, ¿te gustó desarrollarlo con tu grupo? ¿Le dedicaron mucho tiempo y esfuerzo? ¿Podrías decirme más sobre cómo administraron el proceso? ¿Qué los motivó?

Entrevistado: *Sí, la verdad es que la idea está muy buena. Si bien el desarrollador principal no fui yo. Prácticamente mi función, más bien, fue de testarlo. Pero fue muy interesante y se aprendió bastante porque, bueno, teníamos a alguien que manejaba el tema, ¿no? Y cómo lo administramos, bueno, en realidad, Gaby lo desarrollaba y nos pasaba y hacíamos las pruebas. Nos manejamos prácticamente por internet. No nos reunimos. Y la motivación era terminarlo y aprobar.*

**DESVJ-04 (Programador PHP, 30 años, desarrolló *Programación Verdura*, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los proyectos desarrollados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *El que más me gustó —no me acuerdo bien el nombre— es el de carreras que habían hecho, y lo que más me gustó es que habían agregado muchos elementos al juego, entonces lo hacían tener en cuenta muchos más factores de los que, por ahí, más linealmente se le ocurren a uno.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Yo creo que el concepto más fuerte que me llevé es el tema de las herencias y cómo trabajar también el tema de los objetos. Mayoritariamente eso.*

Entrevistador: Volvamos al proyecto del videojuego, ¿te gustó desarrollarlo con tu grupo? ¿Le dedicaron mucho tiempo y esfuerzo? ¿Podrías decirme más sobre cómo administraron el proceso? ¿Qué los motivó?

Entrevistado: *Me gustó. A mi grupo también. Nos gustó. Nos gustaría haberle dedicado un poco más de tiempo porque fuimos muy puntuales en cuanto a las necesidades y a las cosas que íbamos teniendo, con la idea de llegar a la fecha límite, a la fecha de entrega. Sí, fue un esfuerzo en el sentido de... desde aprender algo nuevo, organizar los tiempos, organizar... También, bueno, el diseño, sacar una idea de qué hacer, cómo hacer, y a partir de ahí también el tema de cómo diseñar y diagramar todo eso... A lo que yo más — por ahí— le dediqué tiempo fue, sobre todo, a la parte inicial. Me encargué de lo que fue la idea, de pensarla, desarrollarla, esquematizarla y el diseño. Y mi compañero, mientras yo iba haciendo eso, iba aprendiendo cómo usar el 'framework'. ¿Qué es lo que más me motivó durante el proceso? Yo creo que fue el hecho de que es aprender a hacer un juego que es... que creo que va a ser uno de... o es el tema que yo tengo pensado para entregar como trabajo final de la carrera.*

#### **DESVJ-05 (Logística y distribución, 28 años, desarrolló *The Java Mania Game*, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los proyectos desarrollados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Bueno, lo que más me impresionó de los proyectos fue el juego o videojuego que hicimos. ¿Por qué? Porque al comienzo empecé a programar desde cero en Java y*

*cuando me mencionaron el videojuego era como que lo veía imposible... realizarlo. A medida que uno se va proponiendo hacer las cosas —con ayuda del profesor, obviamente— se le va haciendo más fácil el poder programar y llegar a la meta final. Pero, la verdad que me asombró porque —con voluntad y confianza... y el grupo también— se pudo llegar.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Bueno, conceptos que empecé a dominar ahora son: lo básico de la programación orientada a objetos, son cosas que verdaderamente —si lo tenés bien claro todas esas cosas— se te va a hacer más fácil la programación que lo que es más avanzado en Java.*

Entrevistador: Volvamos al proyecto del videojuego, ¿te gustó desarrollarlo con tu grupo? ¿Le dedicaron mucho tiempo y esfuerzo? ¿Podrías decirme más sobre cómo administraron el proceso? ¿Qué los motivó?

Entrevistado: *Bien, la verdad que a todos nos gustó. Al principio, como dije, nos re asombró el poder desarrollar un videojuego. Pero después —poco a poco— nos fuimos animando entre nosotros. Al principio fue duro, ¿no? Como ya venía diciendo... le tuvimos que dedicar muchísimo —pero muchísimo— tiempo. Recuerdo una anécdota que fue juntarnos un sábado y ese sábado crear la primera pantalla del videojuego. Después, al segundo sábado, nos volvimos a juntar y estuvimos casi como cinco horas y no pudimos avanzar nada. Así que volvimos a retomar lo que era leer el libro y, al siguiente sábado —que nos volvimos a juntar— era como que ya las cosas nos iban saliendo un poco mejor. Y eso también nos motivaba, ¿no? A seguir adelante... Como que iba a salir e íbamos a llegar a la meta final. La administración del tiempo... era un poco complicado por el tema de que algunos del grupo trabajaban, y otros —por ahí— tenían más tiempo que otros. Bueno, algo flexible fue lo del tiempo, pero sí... Nos dábamos a cada persona algo para hacer y al siguiente sábado era como que teníamos que traerlo hecho para después sumarlo todo al proyecto y ver si compilaba o no. Y así nos fuimos administrando.*

## ESTUDIANTES QUE USARON VIDEOJUEGOS EN PROGRAMACIÓN II (USOVJ)

### USOVJ-01 (Empleado administrativo, 28 años, hombre)

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *¿Qué me impresionó más y por qué? Me parece interesante lo de los juegos, este... Me parece interesante desde el lado de que los que los desarrollaron habrán aprendido desarrollando y, al jugar, hay cosas que te quedan, sobre todo el tema de la abstracción, cuando vimos el juego de las flechas, este... Me pareció interesante por ese lado.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Lo que estamos viendo ahora... Lo de Modelo-Vista-Controlador, eh... Y bases de datos. Es lo que... Es lo últimos que vimos.*

Entrevistador: Pensando en los videojuegos que jugaste al cursar esta materia, ¿te gustó utilizarlos con tu grupo? ¿Les dedicaron mucho tiempo y esfuerzo?

Entrevistado: *Esfuerzo no. Eran fáciles. Es simple aprenderlos, este... Y sí... Sí, sí, me parece que es una experiencia que está buena y que no se usa, además, bah... No? En tres años que vengo acá, nunca me habían hecho jugar, y menos para... para aprender algo. Me parece que es... que está bueno.*

### USOVJ-02 (Soporte técnico, 23 años, hombre)

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Eh... ¿Puedo pensar? Eh... ¿Recursos de Programación II? Y... Los patrones de diseño... Capaz...*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Mmm... El modelado de objetos...*

Entrevistador: Pensando en los videojuegos que jugaste al cursar esta materia, ¿te gustó utilizarlos con tu grupo? ¿Les dedicaron mucho tiempo y esfuerzo?

Entrevistado: *¿Para resolverlos? Eh... No les dedicamos tanto esfuerzo, pero sí puedo asegurar que, de alguna manera, aprendí, eh... Por ejemplo, tipos de relaciones entre clases, eh... No digo que a la fuerza, pero sí como más subliminalmente.*

### **USOVJ-03 (DBAdmin - Soporte de aplicaciones, 25 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Eh... Me impresionó más, eh... Bueno, el tema de los videojuegos. Eh... El tema de usar un IDE bastante fácil de usar o, por lo menos para mí, fue una curva de aprendizaje bastante buena, no sé cómo se describe exactamente, pero digo, aprendí bastante rápido a usar el NetBeans, cosa que... Una vez cuando quería... había querido usar el Eclipse me pareció más complejo de utilizar para hacer exactamente lo mismo.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Eh... Bueno, buenas prácticas de programación. Eh... Buenas prácticas de programación en sí, polimorfismo, herencia, eh... Etcétera. Eh... Respecto, bueno, quizás no son conceptos en sí, pero son cosas de programación. Yo no había visto antes programación web, sí HTML a secas, pero no... Eh... No había visto JSP, no sabía lo que era un 'servlet', eh... No sabía... Bueno, no sabía bastantes cosas, en realidad, la verdad que me cuesta puntualizar los conceptos, etc. Pero, la verdad que, hasta ahora, estoy aprendiendo mucho y todavía quedan un par de clases, pero no muchas más, pero me llevo —hasta ahora— mucho contenido... aplicable.*

Entrevistador: Pensando en los videojuegos que jugaste al cursar esta materia, ¿te gustó utilizarlos con tu grupo? ¿Les dedicaron mucho tiempo y esfuerzo?

Entrevistado: *Bueno, me gustó utilizarlos. Voy a ser sincero, la verdad que no los utilicé demasiado, pero sí a mis compañeros les gustó y los vi muy eh... —como se dice*

*acá en Argentina— ‘copados’ [entusiasmados] con ellos. Tanto a... Bueno, podría decir los nombres de ellos pero no los voy a decir, pero compañeros cercanos a mí. Sí, me gustó usarlos y me pareció bastante didáctico aprender cosas que, en vez de leerlas por ahí de un papel, que por ahí queda, pero no tanto, que es más... Suena a memorizarlo, en esta forma, al incorporar algo lúdico con... Mientras uno va viendo cuando pasan los ‘frameworks’ y todo eso, es... Eh... Nada, es divertido y a la vez aprende uno, ¿qué se yo?*

#### **USOVJ-04 (Atención al cliente - Docente, 29 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Los juegos que utilizamos en el primer cuatrimestre. El que más me acuerdo es el de las flechitas, que era un juego en el que vos tenías que ir embocando — digamos— disparándoles a las relaciones de las distintas clases en función del significado. Me resultó mnemotécnico. Me lo acuerdo porque me resultó útil como mnemotecnia.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Eh... No necesariamente esas clases, sino... Creo que... Armar el proyecto... Como intentar pensarlo... Como previo a este manejo de clases... Intentar pensarlo. Después plantear las clases, quizás me... Me mezclo un poco más. Pero... Pero el planteo, creo que estoy más fuerte.*

Entrevistador: Pensando en los videojuegos que jugaste al cursar esta materia, ¿te gustó utilizarlos con tu grupo? ¿Les dedicaron mucho tiempo y esfuerzo?

Entrevistado: *Eh... Sí, me gustó. Y... Le dedicamos tiempo, sí, porque... Más que nada, el de las flechitas, me acuerdo, por el tema de que —al principio— no recordábamos bien cuáles eran... Intentábamos avanzar y no avanzábamos. Pero... Creo que después de practicarlo bastante, algunas cosas fueron quedando.*

**USOVJ-05 (Administrativa, 32 años, mujer)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Bueno, la herencia es uno de los recursos que se utilizan en Programación II y que más me sorprendió, porque se pueden aprovechar los miembros de otras clases y —a la vez— se ahorra mucho tiempo de codificación. También, si lo hacemos bien, nos aseguramos de que el programa tenga menos errores. Es como una medida más de seguridad. Me pareció bastante práctico este recurso.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Bueno, creo que los conceptos que más me quedaron grabados fueron las diferentes relaciones que puede haber entre las clases. Eh... Tuvimos varias horas dedicadas a eso. Además... Eh... Los videojuegos que jugamos eran algunos con respecto a eso también, así que me quedó bastante fijo... Ese... Me quedaron bastante fijos esos conceptos.*

Entrevistador: Pensando en los videojuegos que jugaste al cursar esta materia, ¿te gustó utilizarlos con tu grupo? ¿Les dedicaron mucho tiempo y esfuerzo?

Entrevistado: *Estuvo bueno jugar a los videojuegos en grupo. Porque... Como había que recurrir a los conocimientos aprendidos en clase, por ahí, lo que no se acordaba uno, lo decía el otro. Y de esa forma íbamos recordando y estudiando a la vez. Eh... Se hizo como más llevadero repasar todos esos conceptos. Eh... Te llevaban un tiempito, porque había que aprender a jugarlos, o sea, qué teclas usaban, algunos eran por tiempo, en otros había cosas que uno —por ahí— eh... no se acordaba, así que tenía que agarrar la carpeta y empezar a revisar, así que sí, hubo que dedicarle tiempo y esfuerzo.*

## ESTUDIANTES QUE CURSARON LA ASIGNATURA PROGRAMACIÓN II LUDIFICADA (LUDIF)

### LUDIF-01 (Empleado administrativo, 23 años, hombre)

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Bueno, lo que más me impresionó fue la herramienta GitLab... Realmente es más interactivo de lo que yo esperaba. Si bien sabía que nos iban a preparar en cuanto a... no solo lo que conlleva programar, sino además de programar en grupo, ya que hoy en día las empresas o cualquier negocio que necesite un sistema en sí, armado, no necesariamente tiene que ser un solo programador el que lo haga, generalmente son grupos de programadores que trabajan entre sí, y bueno, me impresionó la herramienta GitLab, la interactividad que tiene con respecto a esto porque se puede trabajar de a varias personas y cada uno desde su propia casa, no necesariamente tienen que estar todos juntos y... ¿Se ve aplicado esto? Eh... La verdad que me impresionó más que nada esa herramienta.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Bueno, considero que mejor domino... No sé si es una unidad en particular, pero... sería cómo encabezar el proyecto, es decir, si nos dieran un programa para armar desde cero la parte que mejor domino sería la organización de las clases, es decir, cómo va a estar organizado un proyecto por distintas clases, que cada clase tenga sus atributos y sus métodos y las relaciones que van a tener entre ellas para que el programa arroje los resultados esperados.*

Entrevistador: Pensando en la primera parte de la materia que cursaste con una dinámica de juego, ¿te gustó cursarla así? ¿Opinás que te aportó algo?

Entrevistado: *Me gustó cursarla así porque pienso que aportó el tema de la competitividad en dos puntos, principalmente, porque prácticamente me obligó a estar al día en la materia, eh... no perder estabilidad en cuanto a las clases, esto significa que, por cada semana, además de que aprendí algo nuevo, sí o sí tenía que reflexionar en lo que habíamos visto la clase anterior para poder estar al día con el desafío y no*



*quedar atrás... Y segundo que... Que pienso que está el tema la competencia, es algo... un aspecto muy realista, porque cuando salgamos de la carrera, digamos, cuando estemos recibidos y vayamos al campo laboral, vamos a ver que se ve así tal cual en un ambiente competitivo en donde cada persona o cada grupo de desarrolladores va a tener que... Que estar juntos, eh... cada uno con sus conocimientos para poder sacar adelante su proyecto a comparación de otros.*

### **LUDIF-02 (Soporte técnico, 27 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Me gustó más lo que es GitLab porque es muy útil para todo lo que es la parte corporativo o empresarial.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Y, por ahora... La parte de... de herencia y polimorfismo.*

Entrevistador: Pensando en la primera parte de la materia que cursaste con una dinámica de juego, ¿te gustó cursarla así? ¿Opinás que te aportó algo?

Entrevistado: *Eh... La verdad es que no me convenció. Eh... Porque, bueno... Como en todos lados, hay personas que están más capacitadas que otras, y otras que no. Y capaz que se ven, eh... en cierta forma, desmotivadas por esa competencia que no es, en cierta forma, tan... justa.*

### **LUDIF-03 (Desarrolladora de software, 30 años, mujer)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Eh... Teniendo en cuenta recursos no como parte de la información de la materia sino como herramientas para utilizar dentro de la materia, eh... creo que lo que encontré más útil... el ... la utilización de GitHub y GitLab porque agiliza mucho y además prepara a los alumnos para un ambiente más de trabajo colaborativo y para un ambiente de trabajo fuera de la facultad.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Creo que... Uy, ¡qué difícil de contestar! La verdad que dentro de lo que es Programación II... no creo que domine demasiado uno específicamente. Pero... la... lo que es herencia y polimorfismo, digamos, es lo que... de lo que tengo un poco más de conocimiento —por ahí— dentro de lo que es... el lenguaje que se dio y la Programación Orientada a Objetos y... Sí, sería... sería... más o menos esa ... esa... esa parte, digamos. Sé que tengo que aumentar mis conocimientos en ese área también. Pero es con lo que me siento —por ahí— un poco más cómoda.*

Entrevistador: Pensando en la primera parte de la materia que cursaste con una dinámica de juego, ¿te gustó cursarla así? ¿Opinás que te aportó algo?

Entrevistado: *Eh... Sí... Al principio... debo decir que me era difícil, por el hecho de que... tengo cierto rechazo a la cuestión competitiva... eh... porque tengo esa idea de... el otro que, por ser competitivo, trata de hacerte el mal y no de compartir la competitividad en un buen sentido. Pero, la verdad que en el grupo que se formó, por lo menos este año, no vi esa situación, sino más bien una cuestión de juego bien tomado y... y risa, digamos, de poder competir sanamente y le agregó un... Para mí funcionó porque le agregó un 'plus' a venir a cursar un viernes a la noche, que era muy importante y muy difícil a veces... Pero... me parece muy bien que se haya... que se haya agregado esta situación competitiva, digamos, que era corto, era relativamente fácil... no era muy difícil de... de generar, y agregaba bastantes conocimientos a situaciones que se podían dar en el código.*

#### **LUDIF-04 (Soporte IT, 28 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Eh... Por ejemplo, a diferencia de Programación I donde te enseñaban C... la versatilidad y la facilidad —por ahí— por ejemplo, para... tener... eh... no sé, cuando vos creás usuarios, tener una facilidad para poder acceder o crear objetos*

*a través de... me pareció mucho más simplificado que en lo que fue en C, que era totalmente mucho más complejo, lejos.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Eh... todo lo referido a herencias lo estuve practicando un montón y estuve haciéndome unos machetes y cosas al respecto. Eh... Particularmente con eso, fue lo que me impresionó, me interesó porque... Para poder armar bien las estructuras y demás... es lo básico, totalmente.*

Entrevistador: Pensando en la primera parte de la materia que cursaste con una dinámica de juego, ¿te gustó cursarla así? ¿Opinás que te aportó algo?

Entrevistado: *Eh... Sí, me gustó cursarla así y... Eso me incentivó un poquito más para poder... eh... dedicarle más porque soy eh... —mal que mal— soy una persona que me gusta competir, me gusta jugar y... hacerlo de esta manera se me hizo una forma más... más dinámica y más motivadora, de alguna forma.*

#### **LUDIF-05 (Desarrollador, 25 años, hombre)**

Entrevistador: Mirando hacia atrás, ¿cuál de los recursos utilizados en Programación II te impresionó más? ¿Por qué?

Entrevistado: *Eh... de recursos: groups.io. Me pareció bastante piola. Está bueno porque podemos usar... comunicarnos entre los mismos alumnos... nos mantiene al día y a Usted, para saber si... si hay clases, el contenido, todo, está bastante bueno.*

Entrevistador: Reflexionando sobre todos los conceptos aprendidos en Programación II, ¿cuáles son los que dominás más?

Entrevistado: *Eh... Creo que polimorfismo y herencia.*

Entrevistador: Pensando en la primera parte de la materia que cursaste con una dinámica de juego, ¿te gustó cursarla así? ¿Opinás que te aportó algo?

Entrevistado: *Sí, sí, sí me gustó. Eh... Me parece que aporta competitividad, un poco, y motivación para que... que no se duerman los chicos, que traten de practicar para ver si... si pueden elegir un buen tema de parcial.*



# ÍNDICE

---



## 1

*1000 Miles (videojuego)*, 117

*1942 (videojuego)*, 171

## 3

*3-D Tic-Tac-Toe (videojuego)*, 117

*3x3 Puzzle (juego)*, 165

## 4

*4WC (Four Week Challenge)*, 191

## 6

*6D (marco de diseño)*, 183, 204

## A

*A-10 Attack (videojuego)*, 117

*Aarseth, E.*, 118, 119

*Abelson, H.*, 23, 27, 44

*About Ludology (ponencia)*, 126

*Abstracción (pilar de la programación orientada a objetos)*, 50

*Abt, C. C.*, 140, 141

*Accredible BadgeBuilder (herramienta para ludificación)*, 188

*ACM (Association for Computing Machinery)*, 10, 13, 43

*ACM Digital Library (base de datos online)*, 150, 188

*ACM-ICPC – International Collegiate Programming Contest (competición)*, 16, 190

*Acti-Logo (lenguaje)*, 26

*ActionScript (lenguaje)*, 163

*Ada (lenguaje)*, 30

*Adobe Flash (plataforma de software)*, 163

*ADVA (Asociación de Desarrolladores de Videojuegos Argentina)*, 129

*Advanced Placement Computer Science (examen)*, 23, 36, 38

*Advanced Technology Internet Group*, 37

*Aerobiz (videojuego)*, 117

*Affordance (ofrecimiento)*, 176

*Agapito, J. L.*, 192

*Age of Empires (videojuego)*, 119

*AgentSheets (entorno de desarrollo)*, 33, 36, 44, 162, 163

*Agôn (principio que rige algunos juegos)*, 110

*Agre, G.*, 188

- Agregación (relación entre clases)*, 51
- Aguirre, J.*, 11
- AHA (American Heart Association)*, 176
- Ahorcado (juego)*, 164, 165
- AI Fleet Commander (videojuego)*, 117
- Akl, R. G.*, 166
- Alcorn, A.*, 115
- Alea (principio que rige algunos juegos)*, 110
- Alemania*, 15
- Alexander, R. C.*, 18
- Alfa de Cronbach*, 98
- Alfabetización computacional*, 24, 37, 55
- Alga-Blaster (videojuego)*, 136
- ALGOL (lenguaje)*, 12, 15, 16
- Algoritmo*, 22, 30, 41, 47, 73, 145, 147, 169, 178
- Alice (entorno de desarrollo)*, 35, 39, 163
- Alice99 (entorno de desarrollo)*, 35
- Almacenamiento en caché*, 54
- AlphaWorks (comunidad web)*, 37
- Alto (computadora)*, 18
- Alvarez, J.*, 142, 143
- Ambach, J.*, 33
- Ambigüedad*, 67
- American Football (videojuego)*, 117
- Análisis sistémico*, 159
- Anarchy Online (videojuego)*, 119
- Andalucía*, 25
- Anderson, J. R.*, 24
- Anderson, L. W.*, 59
- Anderson, R.*, 38
- Android (sistema operativo)*, 42, 44, 121, 147, 163, 193
- Andy Capps (bar)*, 115
- Angelova, G.*, 188
- Angry Birds (videojuego)*, 147
- Anita Borg Institute*, 43
- Annan, K.*, 56
- ANSI C (lenguaje)*, 33
- AntMe! (videojuego)*, 150
- API (Application Programming Interface)*, 168, 194
- App Inventor (entorno de desarrollo)*, 44, 191
- Apple Computer, Inc. (empresa)*, 21, 27, 28, 34
- Apple II (computadora)*, 18, 28
- Apple IIc Plus (computadora)*, 20
- Apple Inc. (empresa)*, 42
- Apple Lisa (computadora)*, 27
- Apple Macintosh (computadora)*, 21, 27
- Aprendizaje basado en juegos digitales*, xxix, 2, 133, 134, 136, 141, 189
- Aprendizaje basado en problemas*, 136, 146, 167
- Aprendizaje centrado en el estudiante*, 68
- Aprendizaje en línea (e-learning)*, 179, 188
- Aprendizaje multimodal*, 158
- Aprendizaje piagetiano*, 152
- Aprendizaje profundo*, 68, 70
- Aprendizaje sobre el aprendizaje*, 159



- Aprendizaje superficial*, 70
- Aptitud para programar*, 2, 68, 70, 71, 72
- AquaZone* (videojuego), 117
- Árbol binario de búsqueda* (estructura de datos), 145
- Arduino* (plataforma de prototipado electrónico), 38, 46
- Arenero* (sandbox), 148
- Argentina*, xxix, 1, 3, 9, 11, 13, 20, 22, 24, 28, 49, 54, 91, 93, 127, 128, 129, 130
- Argumento analógico*, 65
- Aristóteles*, 103
- Arkanoid* (videojuego), 117
- Armstrong, P.*, 149
- Arnedo-Moreno, J.*, 183
- Arrastrar y soltar* (acción con el ratón), 33, 36, 39, 41, 47, 162, 163
- Arreglo* (estructura de datos), 169
- Arruga i Valeri, A.*, 75, 76, 77, 81, 82, 83, 84, 85
- ARS* (Análisis de Redes Sociales), 75
- Arsgames* (asociación sin fines de lucro), 127
- Artes animadas*, 122, 123
- ASEIV* (Asociación Española de Investigadores de Videojuegos), 127
- Asesino/Killer* (tipo de jugador), 124
- Asociación* (relación entre clases), 51
- Asociacionismo*, 127, 129
- Aspray, W.*, 43
- Assembler* (lenguaje ensamblador), 14
- Assembly* (lenguaje ensamblador), 148
- Atari 800* (computadora), 20
- Atari Baseball* (videojuego), 117
- Atari Inc.* (empresa), 21, 115
- Atari Institute for Educational Action Research*, 21
- Atari ST* (computadora), 21
- Ataxx* (videojuego), 117
- Atkinson, B.*, 27
- ATLAS.ti* (software para el análisis de datos cualitativos), 216
- Atributo* (estado de un objeto), 50, 51
- Aumont, J.*, 122
- Australia*, 36, 40
- Austria*, 48
- Autocode* (lenguaje), 11
- Auwinen, T.*, 193
- Avatar* (elemento del diseño de juegos), 149, 179, 180, 181, 182, 188
- Avengers* (videojuego), 117
- Avverbuj, G.*, 129
- AVuSG* (Algorithm Visualization using Serious Games), 145
- ## B
- Babini, N.*, 11
- Backgammon* (videojuego), 117
- Backtracking* (algoritmo), 54
- Badger, M.*, 41
- Baer, R.*, 115

- Baffle Ball (juego)*, 112
- Baker, A.*, 144
- Baldur's Gate (videojuego)*, 119
- Banerji, N.*, 96
- Banzi, M.*, 38
- Baptista Lucio, P.*, 197
- Barcelona Code School (campo de entrenamiento de codificación)*, 45
- Barclaycard (empresa)*, 177
- Barclaycard Ring (tarjeta de crédito)*, 177
- Bar-On, R.*, 88, 99, 100
- Barragán, H.*, 38
- Bartha, P.*, 65
- Bartle, R.*, 124, 125
- Barton, M.*, 20
- Base de datos*, 32, 64, 188
- BASIC (lenguaje)*, 12, 13, 14, 15, 19, 22, 28, 117
- Bastin, G.*, 76, 77, 78, 79, 81, 85, 86
- Baszuro, P.*, 192
- Batalla naval (juego)*, 165
- Bates, B.*, 205
- Bateson, G.*, 105
- Battletech (videojuego)*, 117
- Battlezone (videojuego)*, 117
- Bau, D.*, 47
- Bauer, A.*, 150
- BBC Micro (computadora)*, 21
- BBN - Bolt, Beranek and Newman, Inc. (empresa)*, 14, 15
- BeagleBoard.org Foundation*, 46
- BeagleBone Black (computadora de placa única)*, 46
- Beatmania (videojuego)*, 117
- Beck, K.*, 34
- Bélgica*, 45, 160
- Bell Labs (empresa)*, 27
- Bellucci, M.*, 54
- Berners-Lee, T.*, 31
- Bertie the Brain (juego)*, 113
- Biblioteca de funciones*, 29, 168, 169, 170, 171, 172
- Big Bird's Egg Catch (videojuego)*, 117
- Binary Search (videojuego)*, 145
- Binary Search Tree (videojuego)*, 145
- Bisquerra Alzina, R.*, 87, 99, 101
- Bisson, G.*, 21
- Bit (robot)*, 145
- Blackjack (juego)*, 164
- Blanco, S.*, 149
- Blas, M. J.*, 167
- Bloc (plataforma para aprender a programar online)*, 45
- Blohm, I.*, 179
- Bloom, B. S.*, 59, 61
- Bloom, M.*, 14
- Blue (lenguaje)*, 33, 35
- BlueJ (entorno de desarrollo)*, 35, 40
- Bobrow, D.*, 14
- Body Slam (videojuego)*, 117

- Bogost, I.*, 3, 187
- Booch, G.*, 34, 36
- Boolean (clase de envoltorio)*, 53
- boolean (tipo primitivo)*, 53
- Boot Camp (videojuego)*, 117
- Borchers, D.*, 113
- Borches, C.*, 11
- Borland Inc. (empresa)*, 23, 31
- Botero Tabares, R. de J.*, 145
- Boulder Dash (videojuego)*, 117
- Bowen, B. D.*, 32
- Boxer (entorno de desarrollo)*, 27
- Boyatzis, R.*, 100
- Breakout (videojuego)*, 117, 165
- Brecha comercial*, 56
- Brecha de contenido*, 56
- Brecha de género*, 56, 161, 162
- Brecha digital*, 56
- Brecha racial*, 162
- Brecha tecnológica*, 56
- Breen, B.*, 34
- Brookhaven National Laboratory*, 113
- Bruner, J.*, 105
- Bubble Sort (videojuego)*, 145
- Bubinski, R.*, 44
- Bulgaria*, 28
- Burbuja punto com*, 133
- Burguera, A.*, 166
- Burke, B.*, 176, 177
- Burke, Q.*, 9, 30, 153, 154, 155, 156, 157, 158, 159, 161, 162, 163
- Bushnell, N.*, 114
- Búsqueda binaria (algoritmo)*, 145
- Butler, E.*, 150
- Buytendijk, F.*, 104
- Byte (clase de envoltorio)*, 53
- Byte (revista)*, 18, 20
- byte (tipo primitivo)*, 53
- ## C
- C (lenguaje)*, 3, 27, 30, 32, 189
- C# (lenguaje)*, 147, 150, 170, 171, 189
- C++ (lenguaje)*, 27, 30, 32, 33, 35, 36, 38, 144, 164, 166, 167, 169, 170, 171, 189, 192
- C3 (proyecto Chrysler Comprehensive Compensation)*, 34
- Caballero, R. Y.*, 93
- CAI (computer assisted instruction)*, 22
- Cailliau, R.*, 31
- Caillois, R.*, 104, 109, 110, 142, 143, 148
- Calero, A.*, 91
- Cámara de la Industria Argentina del Software*, 54
- Camilo Hernandez, C.*, 171
- Campaña, J. R.*, 194
- Campbell, D. T.*, 201
- Campo de entrenamiento de codificación*, 45
- Canarias*, 25

- Cañeque, H.*, 131, 132
- Capacidad cognitiva*, 88
- Capacidad de procesamiento*, 64, 65
- Capacidad matemática*, 64
- Capcom Co., Ltd. (empresa)*, 171
- Capital Blue Cross (empresa)*, 177
- Capital Children's Museum*, 20
- Cara o cruz (juego)*, 165
- Carmen Sandiego (videojuego)*, 135
- Carnival (videojuego)*, 117
- Carnota, R.*, 11, 12, 14, 16
- Caruso, D. R.*, 93, 99
- Casano, J. D.*, 192
- Casino (videojuego)*, 117
- Castaño Collado, C.*, 56
- Castellaro, M.*, 167
- Castro, C.*, 131, 132
- Cataluña*, 25
- Cat-A-Pult (videojuego)*, 165
- Cathode Ray Tube Amusement Device*, 113
- Caton, H.*, 190
- Catrobat (lenguaje)*, 48
- CBS Interactive Inc. (empresa)*, 121
- CDGM (Collaborative Digital Game Making)*, 160
- CeeBot (videojuego)*, 146
- CeeBot3 (videojuego)*, 147
- CeeBot4 (videojuego)*, 147
- CeeBot-A (videojuego)*, 147
- CeeBot-Teen (videojuego)*, 146
- Cellan-Jones, R.*, 9
- Centipede (videojuego)*, 117
- Centro de Epistemología Genética*, 152
- Centro Universitario de Tecnología y Arte Digital*, 127
- CERN (Conseil Européen pour la Recherche Nucléaire)*, 31
- Cervo, D.*, 38
- Championship Manager 4 (videojuego)*, 120
- Changing Minds: Computers, Learning, and Literacy (libro)*, 36
- char (tipo primitivo)*, 53
- Character (clase de envoltorio)*, 53
- Charbonneau, E.*, 149
- Chase HQ (videojuego)*, 117
- Chasteen, M.*, 190
- Checkers (videojuego)*, 117
- Chen, J. X.*, 145
- Chen, M.*, 19
- Chen, S.*, 141
- Chess (videojuego)*, 117, 119
- ChildProgramming (proceso de enseñanza/aprendizaje)*, 191
- ChildProgramming-G (proceso de enseñanza/aprendizaje)*, 191
- Chrysler Corporation (empresa)*, 34
- Ciclo de sobreexpectación (hype cycle)*, 173
- C-jump (juego)*, 144
- Claparède, É.*, 104
- Clark, W.*, 17

- Clase (plantilla de código)*, 51, 145
- Clase de envoltorio (wrapper class)*, 53
- Classcraft (herramienta para ludificación)*, 188
- ClassDojo (herramienta para ludificación)*, 188
- Classroom Social Distance Scale (instrumento sociométrico)*, 79, 84
- Clementina (computadora)*, 11, 12
- Clúa, O.*, 166
- CMU (Carnegie–Mellon University)*, 24, 35
- CNE (Canadian National Exhibition)*, 113
- COBOL (lenguaje)*, 12, 22
- Cociente intelectual*, 87
- Cocos2d (framework)*, 170
- Cocos2d-JS (framework)*, 170
- Cocos2d-Python (framework)*, 170
- Cocos2D-Swift (framework)*, 170
- Cocos2d-x (framework)*, 170
- Cocos2d-XNA (framework)*, 170
- Code Academy (campo de entrenamiento de codificación)*, 45
- Code Avengers (plataforma para aprender a programar online)*, 45
- Code Quiz (juego)*, 167
- Code School (plataforma para aprender a programar online)*, 45
- Code Warriors: Hakitzu Battles (videojuego)*, 147
- Code.org (organización sin fines de lucro)*, 47, 147
- Codecademy (plataforma para aprender a programar online)*, 44
- CodeHS (plataforma para aprender a programar online)*, 45
- Código abierto*, 38, 47, 48, 163
- Código de máquina*, 13
- Coffeescript (lenguaje)*, 47
- Cohn, D.*, 144
- Coincidencia de patrones*, 70
- Cola (estructura de datos)*, 169
- Colegio Universitario IES (Instituto de Estudios Superiores)*, 128
- Collazos, C. A.*, 191
- College Board (organización sin fines de lucro)*, 23, 36, 38
- Colmerauer, A.*, 16
- CoLoBoT (videojuego)*, 146
- Comanche 3 (videojuego)*, 117
- Combat (videojuego)*, 117
- Comercio electrónico*, 56
- Comisión Europea*, 47, 160
- Comisión Nacional de Informática*, 24
- Commodore 128 (computadora)*, 20
- Commodore Amiga (computadora)*, 21
- Commodore PET (computadora)*, 18
- Communications of the ACM (periódico)*, 41
- Compaine, B. M.*, 56
- Comparative Media Studies (carrera)*, 122

- Competencia emocional*, 101
- Competencia emocional y social*, 100
- Competencia social*, 99
- Compilación just-in-time*, 35
- Compilador (software)*, 12, 23
- Comportamiento (componente de un objeto)*, 50
- Composición (relación entre clases)*, 51
- Comprobación de la realidad*, 99
- Computación personal*, 17, 18
- Computadora de placa única*, 46
- Computadoras para el resto de nosotros*, 39
- Computational literacy*, 37
- Computational Thinking (artículo)*, 41
- Computational Thinking in K-12: A Review of the State of the Field (artículo)*, 46
- Computer literacy*, 37
- Computer Science For All (iniciativa gubernamental)*, 10
- Computer Space (videojuego)*, 114
- Computers and the World of the Future (simposio)*, 12
- Computing in the Core*, 47
- Computing in the Core (coalición)*, 43
- Computing Research Association*, 43
- Comunidad Valenciana*, 25
- Conectar Igualdad (programa gubernamental)*, 1
- Conseguidor/Achiever (tipo de jugador)*, 124
- Construccionismo*, xxix, 2, 3, 5, 29, 30, 151, 152, 153
- Constructivismo*, 29, 154, 155
- Conundra Ltd. (empresa)*, 174
- Coonradt, C. A.*, 180
- Cooper, S. C.*, 39
- Coplien, J.*, 34
- CoquitoDobleO (videojuego)*, 145
- Corbató, F. J.*, 12
- Cortés, C. D.*, 12
- Counter-Strike (videojuego)*, 119
- Coursera (plataforma para aprender online)*, 45
- Craps (juego de pase inglés)*, 164, 165
- Cratilo*, 1
- Create@School (entorno de desarrollo)*, 48
- Creative Skillset (organización sin fines de lucro)*, 162
- Creatures (videojuego)*, 117
- Crespi, M. C.*, 93
- Creswell, J. W.*, 198
- Critter caretaker (juego)*, 164
- Crowdsourcing (colaboración masiva)*, 178
- CRT (Cathode Ray Tube)*, 109
- CSEdWeek (Computer Science Education Week)*, 43, 47
- Csikszentmihályi, M.*, 103, 105, 106, 107
- CSS (Cascading Style Sheets)*, 44
- CSTA (Computer Science Teachers Association)*, 43

- CSUA (*Computer Science Undergraduate Association*), 148
- CTSS (*Compatible Time Sharing System*), 12
- Cuadrado, A., 127
- Cuartielles, D., 38
- Cuestionario de autoinforme, 88, 89, 96, 97
- Cumbre Mundial sobre la Sociedad de la Información, 56
- Cunningham, C., 175
- Cunningham, R., 79, 82, 84
- Cunningham, W., 34
- Cvijikj, I. P., 38
- D**
- d de Cohen, 249, 315
- Dabney, T., 114
- Dagiene, V., 39
- Daley, R. C., 12
- Dance Dance Revolution (*videojuego*), 117
- Danino, N., 191
- Dann, W., 39
- Dartmouth College, 12, 114
- Data-driven development (*metodología de desarrollo de software*), 170
- Dawson, M., 164
- Daytona U.S.A. (*videojuego*), 117
- de Halleux, J. P., 149
- Dealer Demo (*Bally Astrocade*), 117
- DEC - *Digital Equipment Corporation* (*empresa*), 17, 18
- DEC PDP-1 (*computadora*), 114
- Decreto 459/10 (*legislación argentina*), 1
- Delphi (*entorno de desarrollo*), 31
- Dependencia (*relación entre clases*), 51
- Depuración (*fase de un proceso*), 55, 157
- Desafío Bebras (*competición*), 39
- Desarrollador de información (*tipo de usuario*), 28
- Desarrollo de videojuegos sin escribir código, 162
- Desarrollo de videojuegos usando lenguajes de propósito general, 165
- Desarrollo dirigido por datos (*metodología de desarrollo de software*), 170
- Descent (*videojuego*), 117
- Design Patterns: *Elements of Reusable Object-Oriented Software* (*libro*), 33
- Designatum (*componente de la semiosis*), 57
- Detección de colisiones, 170
- Deterding, S., 175, 179
- Dev Bootcamp (*campo de entrenamiento de codificación*), 45
- DeVries, R., 131
- Dewey, J., 104
- DGBL - *Digital Game-based Learning* (*estrategia de aprendizaje*), 134
- Dhillon, G., 58, 59
- Diablo (*videojuego*), 117

- Diagnostic Cartridge FDS100144 del Atari 5200*, 117
- Diagrama de clases (UML)*, 51
- Diagrama de secuencia (UML)*, 49, 50
- Díaz Matarranz, J. J.*, 101
- Dichev, C.*, 188
- Dicheva, D.*, 188
- Diferencia entre el constructivismo y el construccionismo*, 152
- Dig Dug (videojuego)*, 117
- Digital Diner (blog)*, 46
- Digital Game-Based Learning (libro)*, 134
- Digital Research, Inc. (empresa)*, 27
- Digitalización*, 1, 10, 53
- Dijkstra, E. W.*, 15
- Dillon, R.*, 18, 113, 115
- Dinamarca*, 126
- Direct3D (API)*, 168
- DirectTV, LLC (empresa)*, 176
- DirectX (SDK)*, 168
- Diseño de juegos*, 3, 151, 158, 163, 167, 175, 178, 179, 180, 181, 182, 185, 186, 190, 199, 239
- diSessa, A. A.*, 19, 27, 36, 37
- Distancia social del grupo (valor sociométrico)*, 84
- Distancia social media (índice sociométrico)*, 84
- Distancia social propia (valor sociométrico)*, 84
- Diversión seria (enfoque pedagógico)*, 43
- Dixon, D.*, 175
- DIY - Do-it-yourself (cultura del bricolaje)*, 154
- Djaouti, D.*, 142, 143
- Dodge 'Em (videojuego)*, 117
- Doménech Betoret, F.*, 62
- Dominio muestral (conjunto de dimensiones de un modelo)*, 94, 95, 96, 100
- Dong, Z.*, 190
- Donkey Kong (videojuego)*, 117
- Don't Bother Me Mom — I'm Learning! (libro)*, 135
- Doom (videojuego)*, 117, 119
- Dopplick, R.*, 10
- Double (clase de envoltorio)*, 53
- double (tipo primitivo)*, 53
- Douglas, A.*, 113
- Drag and drop (arrastrar y soltar)*, 33, 36, 39, 41, 47, 162, 163
- Dragon Architect (videojuego)*, 150
- Draughts (juego de damas)*, 113
- Drill and practice (ejercicio y práctica)*, 136
- DTSS (Dartmouth Time-Sharing System)*, 12
- Du Boulay, B.*, 73
- Dungeons & Dragons (videojuego)*, 117, 119
- Durán, W. O.*, 12
- DWP (Department for Work and Pensions)*, 177



*Dynabook (dispositivo)*, 17, 18, 19

## E

*E.T. The Extraterrestrial (videojuego)*, 117

*Earp, J.*, 151

*EDSAC (computadora)*, 113

*educ.ar (portal educativo argentino)*, 1

*Educación Primaria*, 3, 9, 10, 21, 22, 24, 26, 30, 47, 155

*Educación Secundaria*, 3, 9, 10, 22, 23, 26, 30, 40, 47, 155

*Educación Superior*, xxix, 2, 3, 4, 9, 16, 22, 23, 32

*Edutainment (contenido educativo entretenido)*, 135, 141

*edX (plataforma para aprender online)*, 45

*Efecto Kingmaker (característica de algunos juegos)*, 139, 140

*Effacer (videojuego)*, 117

*Egenfeldt-Nielsen, S.*, 120

*Ejercicio y práctica (tipo de juego)*, 136

*Ejercitación frecuente (requisito para aprender a programar)*, 72

*e-learning (aprendizaje en línea)*, 179, 188

*Elium Academy (campo de entrenamiento de codificación)*, 45

*Elkonin, D.*, 105

*else Heart.Break()* (videojuego), 150

*Elzi, A.*, 76

*Emotional competence: How emotions and relationships become integrated (ponencia)*, 101

*Encapsulamiento (pilar de la programación orientada a objetos)*, 50, 51

*Engelhart, M. D.*, 59

*Enigma (dispositivo criptográfico)*, 11

*Enki (IDE para MOOC)*, 194

*Ensmenger, N.*, 70, 71

*Enti (Escuela de Nuevas Tecnologías Interactivas)*, 127

*Entorno de desarrollo integrado (IDE)*, 23, 31, 42, 59, 194

*Epistemology and Learning Group (MIT)*, 29

*Epsitec SA (empresa)*, 146

*EQ-i (Emotional Quotient Inventory)*, 99, 100

*ERIC - Education Resources Information Center (base de datos online)*, 188

*Erikson, E.*, 104

*Escala de la Distancia Social en el Aula (instrumento sociométrico)*, 79, 80, 82, 83, 84, 213

*Escalera semiótica (modelo didáctico)*, 57

*ESCI (Emotional and Social Competency Inventory)*, 100

*Escritura de código*, 2, 35, 44, 72, 159, 162

*Escuela Da Vinci (institución terciaria argentina)*, 128

*Esnaola Horacek, G. A.*, 111, 129, 133  
*ESO (Educación Secundaria Obligatoria)*, 9  
*España*, vii, 9, 24, 30, 45, 48, 91, 93, 127  
*Esquire (revista)*, 122  
*Estado (componente de un objeto)*, 50  
*Estados Unidos*, 10, 11, 21, 23, 36, 45, 47, 54, 57, 114, 115, 123, 161, 177  
*Estatus social*, 76, 179, 181, 184, 192  
*Estatus sociométrico*, 76, 77  
*Estilo de aprendizaje*, 68, 69  
*Estrategia lúdica*, xxix, 2, 3, 4  
*ETD (Educational Technology Department)*, 14  
*ETH (Eidgenössische Technische Hochschule)*, 15  
*Etoys (entorno de desarrollo)*, 34, 36  
*EU Code Week (Semana Europea de la Programación)*, 47  
*Evaluación 360º*, 89  
*Evaluación basada en el desempeño*, 61, 62  
*Evaluación del aprendizaje de la programación*, 61  
*Evaluación formativa*, 62, 63  
*Evaluación objetiva*, 61  
*Evaluación sumativa*, 62  
*Evans, N.*, 21  
*Evo (robot)*, 145  
*Evolve, Inc. (empresa)*, 144  
*Examen de respuestas cortas*, 61, 63

*Exploitationware (medio de explotación)*, 186, 187  
*Explorador/Explorer (tipo de jugador)*, 124  
*extends (palabra reservada)*, 52  
*Extreme Pinball (videojuego)*, 117  
*Extreme Programming (metodología de desarrollo de software)*, 34  
*Extremera Pacheco, N.*, 89, 91, 93, 94

## F

*f de Cohen*, 245, 266  
*Faceta emocional*, 94, 95, 96, 97, 98  
*Factor emocional*, 95, 96, 98  
*Factorovich, P.*, 11, 14  
*Farrell, M.*, 76  
*Fax (videojuego)*, 117  
*Feldgen, M.*, 166  
*Fernández Collado, C.*, 197, 198  
*Fernández Prados, J. S.*, 77, 78, 79, 81, 82, 83, 86  
*Fernández-Berrocal, P.*, 89, 91, 93, 94  
*Ferranti Ltd. (empresa)*, 12  
*Ferranti Mark 1 (computadora)*, 113  
*Ferranti Mercury II (computadora)*, 11  
*Festival of Britain*, 113  
*Feurzeig, W.*, 14, 15  
*Fiabilidad test-retest*, 98  
*FIFA 2004 (videojuego)*, 120  
*Filogénesis (concepto de la teoría de la recapitulación)*, 104

- Final Test Cartridge (Coleco ADAM)*, 117
- Finlandia*, 160
- Fischetti, M.*, 31
- Flight Unlimited (videojuego)*, 117
- Flipper (paleta de una máquina de petacos)*,  
112
- Flipper Game (videojuego)*, 117
- Float (clase de envoltorio)*, 53
- float (tipo primitivo)*, 53
- Flow: The Psychology of Optimal Experience (libro)*, 105
- Flujo (estado de)*, 5, 105, 106
- Flujo de control*, 65, 66, 67
- Fonseca, F.*, 167
- Fontela, C.*, 49, 50
- Foot, S.*, 44
- FORTTRAN (lenguaje)*, 11, 12, 14, 16, 22,  
30
- Framework (marco de trabajo)*, 168, 169,  
170, 171
- Frasca, G.*, 111, 126, 127, 133
- Fraser, S.*, 34
- Frazer, J. G.*, 104
- freeCodeCamp (plataforma para aprender a programar online)*, 45
- Freeman, L. C.*, 75
- Freeway (videojuego)*, 117
- Freiberger, P.*, 20, 23, 27
- Freud, S.*, 104
- Friedman, M.*, 44
- Fröbel, F.*, 104
- Frobenius, L.*, 104
- Frogger (videojuego)*, 117
- Frozen (película)*, 147
- Fry, B.*, 38
- Funciones esenciales del juego*, 132
- Fundación Sadosky (institución público-privada argentina)*, 1
- Fundamentos de la teoría de los signos (libro)*,  
57
- FUNDAV (Fundación Argentina de Videojuegos)*, 129
- Fundéu BBVA (Fundación del Español Urgente)*, 175
- Furnham, A.*, 88, 94, 95, 96, 97, 98, 99
- Furst, E. J.*, 59
- FUSE - Future Social Experiences (laboratorios de Microsoft Research)*, 42
- Future Center (ambiente en el Capital Children's Museum)*, 20
- ## G
- G\*Power (software para el cálculo de la potencia estadística)*, 216
- Gadget (videojuego)*, 117
- Galaga (videojuego)*, 117
- Galicia*, 25
- Galli, M. G.*, 133
- Gamarekian, B.*, 20
- Game Design Project*, 151, 157

- Game studies (estudios del juego)*, 127
- Gamefroot (entorno de desarrollo)*, 163
- GameMaker (entorno de desarrollo)*, 162, 163, 172
- Games for Change (organización sin fines de lucro)*, 140
- Games, I. A.*, 156, 157, 161
- Games, the new lively art (artículo)*, 123
- GameSalad (entorno de desarrollo)*, 162, 163
- GameSpot (sitio web de reseñas)*, 121
- Gamestar Mechanic (entorno de desarrollo)*, 163
- Gamification (ludificación)*, xxix, 2, 5, 173, 175, 176, 179, 186, 188
- Gamification in education - Learn computer programming with fun (artículo)*, 189
- Gamma, E.*, 33
- García Camarero, E.*, 11
- García, A. A.*, 191
- García, N. S.*, 12
- García-Mundo, L.*, 150
- Garlick, R.*, 166
- Gartner, Inc. (empresa)*, 173, 176
- Garzón, G.*, 131
- GCC (GNU Compiler Collection)*, 30
- Gee, E. R.*, 156
- Gee, J. P.*, 135, 136
- GEM (interfaz gráfica de usuario)*, 27
- Generación de Internet*, 134
- Generación gamer*, 111
- General Assembly (campo de entrenamiento de codificación)*, 45
- Generala (juego)*, 165
- Generalización (relación entre clases)*, 51
- Genero, M.*, 150
- Géneros de videojuegos según su interactividad*, 117
- Géneros de videojuegos según sus requerimientos para el éxito*, 120
- George Mason University*, 145
- Georgia Institute of Technology*, 36, 37, 187
- Ghinea, G.*, 2, 72
- Globalización*, 1, 10
- Globaloria (plataforma para aprender a programar online)*, 40, 161
- Gloss (biblioteca de funciones)*, 171
- GML (lenguaje)*, 163, 172
- GNU/Linux (sistema operativo)*, 29, 30
- Go (lenguaje)*, 30
- Goldberg, A.*, 20
- Goldman, S. L.*, 91
- Goldsmith, T.*, 113
- Goleman, D.*, 87, 94, 99, 100, 101
- Golf (videojuego)*, 119
- Gómez Tarín, F. J.*, 127
- Gómez, S.*, 127
- González González, C. S.*, 183, 192
- González Urbano, E.*, 101
- Google LLC (empresa)*, 43, 44, 47, 121
- Google Play (tienda online)*, 121

*Google Scholar (base de datos online)*, 183,  
 188  
*Google+* (red social), 193  
*Gopher (videojuego)*, 117  
*Gordin, D. N.*, 160  
*goto (instrucción)*, 15  
*Gottlieb, D.*, 112  
*Gran Depresión*, 112  
*Grand Theft Auto III (videojuego)*, 119  
*Grant, R.*, 14  
*GREATERP (entorno de desarrollo)*, 24  
*Greco, H.*, 131, 132  
*Green, D.*, 22  
*Greenfoot (entorno de desarrollo)*, 40, 43  
*Greenhill, D.*, 190  
*Grilla 2D*, 40  
*Groos, K.*, 104  
*Groups.io (plataforma de grupos de correo electrónico)*, 204  
*Group-social-distance (valor sociométrico)*, 84  
*Grover, S.*, 46  
*Guerra Antequera, J.*, 111, 136, 137  
*Guerrero, J.*, 166  
*Guess my number (juego)*, 164, 165  
*GUI (Graphical User Interface)*, 27, 31, 39  
*Guitar Freaks (videojuego)*, 117  
*Guzdial, M.*, 10, 12, 37, 53, 54, 55, 56, 57

## H

*Hack Reactor (campo de entrenamiento de codificación)*, 45  
*Hackaton (encuentro de programadores)*, 148  
*Haeckel, E.*, 104  
*Hafner, K.*, 11  
*Hakitzu (videojuego)*, 147  
*Hakulinen, L.*, 193  
*Half-Life 2 (videojuego)*, 120  
*Hall, G. S.*, 104  
*Hall, J. A.*, 76  
*Halloween/Noche de brujas (celebración)*, 192  
*Halma, A.*, 40  
*Hamari, J.*, 176  
*Hangman from the 25th Century (videojuego)*, 117  
*Hardware abierto*, 38, 46  
*Harel, I.*, 29, 32, 40, 153  
*Harrison, P.*, 122  
*Haskell (lenguaje)*, 29, 171  
*Hauque, F.*, 167  
*Haxe (lenguaje)*, 163  
*Hay Group (empresa)*, 100  
*Hayes, E. R.*, 156, 157, 161  
*Healthcare University (plataforma online)*,  
 177  
*HeartChase (competición)*, 176  
*Helm, R.*, 33  
*Hemmendinger, D.*, 13  
*Henriksen, P.*, 40

- Heráclito*, 1
- Herencia (pilar de la programación orientada a objetos)*, 50, 52, 145
- Herman, T.*, 168, 170
- Hernández Sampieri, R.*, 197, 198
- Heroes III (videojuego)*, 119
- Heurística*, 55
- Higinbotham, W.*, 113
- Hill, W. H.*, 59
- Hiltzik, M. A.*, 17
- Historia general de la programación y su aprendizaje*, 10
- Hoadley, C. M.*, 160
- Hole Hunter (videojuego)*, 117
- Home Finance (software)*, 117
- Homeostasis*, 104
- Homo ludens (libro)*, 109
- Homocedasticidad*, 239
- Hope, A.*, 9
- Hora del Código (desafío online)*, 47
- Horace Mann–Lincoln Institute of School Experimentation*, 75
- Horizon 2020 (programa de la Unión Europea)*, 48
- Howes, D.*, 46
- HTML (Hypertext Markup Language)*, 44, 163, 164
- Hughes, J.*, 71
- Huizinga, J.*, 103, 104, 109, 110
- Human Cannonball (videojuego)*, 117
- Human Resource Machine (videojuego)*, 148
- Humpty Dumpty (juego)*, 112
- Hunicke, R.*, 178
- Hunt, A.*, 149
- Hunter, D.*, 175, 182, 183, 184, 185, 186, 187, 199, 200
- Huotari, K.*, 176
- Hurtado, J. A.*, 191
- HyperCard (entorno de desarrollo)*, 27, 28, 34
- HyperTalk (lenguaje)*, 28
- I**
- IBM 360 (computadora)*, 14
- IBM AT (computadora)*, 28
- IBM Corporation (empresa)*, 14, 16, 20, 27, 37, 71
- IBM PAT (Programmer Aptitude Test)*, 71
- IBM PC (computadora)*, 20, 23, 25, 27, 28
- IBM SPSS Statistics (software para el análisis de datos cuantitativos)*, 216
- IBM XT (computadora)*, 28
- IDE (Integrated Development Environment)*, 23, 31, 42, 59, 194
- Idea Street (plataforma online)*, 177
- Identidad (componente de un objeto)*, 50
- iDREAMS (Integrative Design-based Reform-oriented Educational Approach for Motivating Students)*, 42
- IEEE Computer Society*, 43

- IEEE Digital Library (base de datos online)*, 150
- IEEE Xplore (base de datos online)*, 188
- Ifenthaler, D.*, 141
- Ignite Leadership (juego)*, 177
- Igoe, T.*, 38, 39
- Ilinx (principio que rige algunos juegos)*, 110
- Image Campus (institución terciaria argentina)*, 128
- Impossible Mission (videojuego)*, 119
- Impresora láser*, 18
- Impresora matricial*, 25
- Índice de inteligencia emocional percibida o autoinformada*, 89
- Índice sociométrico*, 5, 83, 84, 85
- Índice sociométrico global*, 85
- Índice sociométrico individual*, 85
- Industria basada en el conocimiento*, 49
- Infogenius French Language Translator (software)*, 117
- Informática física*, 38, 46
- Informe sociométrico*, 86
- Ingalls, D.*, 34, 49
- Ingeniería de software*, 45, 144
- Insertion Sort (videojuego)*, 145
- Insignia (elemento del diseño de juegos)*, 179, 181, 182, 185, 189, 194
- Instituto Nacional Superior del Profesorado Técnico*, vii, xxix, 3, 209
- Instituto Privado de la Cámara Argentina de Comercio y Servicios*, 128
- Instruccionismo*, 151, 154
- int (tipo primitivo)*, 53
- Integer (clase de envoltorio)*, 53
- Inteligencia artificial*, 114, 169, 170
- Inteligencia emocional*, 4, 5, 87, 88, 89, 90, 92, 93, 94, 95, 96, 97, 99, 100, 101
- Inteligencia emocional capacidad*, 88, 94
- Inteligencia emocional global*, 95
- Inteligencia emocional rasgo*, 88, 94, 97
- Inteligencia emocional-social*, 88, 100
- Inteligencia general*, 64, 65
- Inteligencia personal*, 94
- Inteligencia psicométrica*, 88
- Inteligencia social*, 87, 88, 94
- Inteligencia socioemocional*, 88
- Interaction Design Institute Ivrea*, 38
- Interfaz gráfica de usuario (GUI)*, 27, 31, 39
- Interpretante (componente de la semiosis)*, 57
- Intérprete (componente de la semiosis)*, 57
- Introduction to programming with Greenfoot: object-oriented programming in Java with games and simulations (libro)*, 43
- Inversión del control (característica de los frameworks)*, 170
- Ioannidou, A.*, 42
- IOI - International Olympiad in Informatics (competición)*, 28
- iOS (sistema operativo)*, 147, 163, 193

- IP (dirección de red)*, 149
- iPad (tablet de Apple)*, 42
- Iparraguirre, A.*, 129
- iPhone (teléfono móvil de Apple)*, 42
- iPod (reproductor de medios de Apple)*, 176
- Israel*, 36
- Istituto per le Tecnologie Didattiche*,  
*Consiglio Nazionale delle Ricerche*, 160
- Italia*, 38, 160
- Ítem (componente de un test)*, 78, 90, 91, 93,  
94, 97, 98, 99
- ITiCSE - Innovation and Technology in  
Computer Science Education (conferencia)*,  
36
- Ivory, J. D.*, 113, 114
- ## J
- Jacobs, G. M.*, 68
- Jacobson, I.*, 34, 36
- Jacovkis, P. M.*, 11
- James, M.*, 34, 62
- Java (lenguaje)*, 3, 32, 33, 35, 37, 38, 39,  
49, 51, 52, 53, 144, 146, 147, 149, 163,  
165, 167, 170, 171, 189
- Java 3D (API)*, 168
- JavaScript (lenguaje)*, 44, 147, 148, 149,  
163, 164, 170
- Jeffries, R.*, 34
- Jenkins, H.*, 122, 123, 124
- Jenkins, T.*, 2, 69, 71, 72, 73
- Jeopardy (videojuego)*, 117, 165
- Jessel, J.-P.*, 142, 143
- Jigsaw (videojuego)*, 117
- jMonkeyEngine (motor de videojuego)*, 171
- Jobs, S.*, 39
- Johnson, L.*, 174
- Johnson, R.*, 33
- Juego*, 103
- Juego complejo*, 135, 136
- Juego con componentes digitales y no digitales*,  
144
- Juego de apuestas*, 110, 117
- Juego de azar*, 109, 110
- Juego de computadora*, 108
- Juego de damas*, 113
- Juego de programación*, 117
- Juego digital*, 108
- Juego electrónico*, 108, 110
- Juego móvil*, 108
- Juego no digital*, 144
- Juego portátil*, 108
- Juegos serios (serious games)*, xxix, 2, 133,  
140, 141, 142, 145, 150, 154, 189, 202
- Jungle Hunt (videojuego)*, 117
- Juul, J.*, 110, 126
- ## K
- Kaasbøll, J. J.*, 57, 58, 61
- Kaczmarczyk, L.*, 10
- Kaehler, T.*, 34



- Kafai, Y. B.*, 9, 30, 151, 153, 154, 155, 156,  
157, 158, 159, 161, 162, 163
- Kahn, P.*, 23
- Kahoot! (herramienta para ludificación)*, 188
- Kalinichenko, E.*, 28
- Kamii, C.*, 131
- Kant, I.*, 103
- Kapp, K. M.*, 187
- Karel The Robot (videojuego)*, 22, 40
- Karel The Robot: A Gentle Introduction to the  
Art of Programming (libro)*, 22
- Karsenty, R.*, 64
- Kates, J.*, 113
- Katholieke Universiteit Leuven*, 160
- Kay, A. C.*, 17, 18, 19, 34
- Kelleher, C.*, 35, 40
- Kemeny, J. G.*, 12, 13
- Kenderov, P. S.*, 28
- Kent, S. L.*, 34, 40, 112, 113, 114, 115
- Keystone Kapers (videojuego)*, 117
- Khaled, R.*, 175
- Khan Academy (plataforma para aprender  
online)*, 45, 177
- Kholodov, I.*, 144
- Khurana, P.*, 189
- King Coder (juego)*, 167
- Knowledge Munchers (videojuego)*, 136
- Kodu (entorno de desarrollo)*, 42, 163
- Kölling, M.*, 31, 33, 35, 40, 43
- Korhonen, A.*, 193
- Kotlin (lenguaje)*, 149
- Kramer, W.*, 139, 140
- Krathwohl, D. R.*, 59, 60
- Krogue, K.*, 180
- Kroll, J.*, 122, 124
- Kuato Studios (empresa)*, 147
- Kumar, B.*, 189
- Kurland, D. M.*, 55, 64, 65, 66, 67
- Kurtz, T. E.*, 12, 13
- ## L
- La Era de Hielo (película)*, 147
- Laberintos (juego)*, 164
- Laboratorios Bell (empresa)*, 27
- Lacasa, P.*, 127
- Lambert, K. A.*, 30
- LaViola Jr, J. J.*, 149
- Lazarus, M.*, 104
- Leal, J. P.*, 194
- Learning to Program with Alice (libro)*, 39
- LeBlanc, M.*, 178
- LEGO (juguete de construcción con bloques)*,  
26, 29
- Lego A/S (empresa)*, 26, 34
- LEGO Mindstorms (kit de construcción de  
objetos programables)*, 34, 35
- LEGO/Logo (kit de construcción de objetos  
programables)*, 26
- Leimeister, J. M.*, 179
- Lemmings (videojuego)*, 119

- Lenguaje de juguete*, 33
- Lenguaje de scripting*, 170
- Lenguaje ensamblador*, 11, 14, 148, 166
- Leuner, B.*, 87
- Lew, L. S.*, 12
- Lewis, P. H.*, 28
- Ley 25.922 (legislación argentina)*, 1
- Li, C.*, 190
- Li, S.*, 37
- LibGDX (framework)*, 170, 202
- Lifelong Kindergarten (MIT)*, 41, 48
- Lifelong Learning Programme (programa de la Comisión Europea)*, 160
- LifePlay (revista)*, 127
- LightBot (videojuego)*, 41, 147, 171
- Likert (escala)*, 89, 97, 98
- LINC - Laboratory INstrument Computer (computadora)*, 17
- Linked List (videojuego)*, 145
- LISP (lenguaje)*, 12, 23, 114
- Lista enlazada (estructura de datos)*, 145, 169
- Little Big Planet (videojuego)*, 156
- Lituania*, 39
- Livingstone, I.*, 9
- LMDG (Learning-by-Making-Digital-Games)*, 151
- LMS (learning management system)*, 188, 192
- Lode Runner (videojuego)*, 117
- Logo (lenguaje)*, 14, 22, 26, 27, 28, 29, 34, 47, 151, 152, 153
- Logo Laboratory (MIT)*, 14
- Logo SB (lenguaje)*, 26
- Loguidice, B.*, 20
- Loh, C. S.*, 141
- Lohr, S.*, 45
- London Psychometric Laboratory*, 98, 212
- Long (clase de envoltorio)*, 53
- long (tipo primitivo)*, 53
- Looney, L.*, 44
- Lost fortune (juego)*, 164
- Lost Luggage (videojuego)*, 117
- LÖVE (framework)*, 170
- LRG - Learning Research Group (Xerox PARC)*, 17
- LSL - Linden Scripting Language (lenguaje)*, 150
- LTi - Learning Tool Interoperability (API)*, 194
- Lua (lenguaje)*, 147, 149, 170
- Ludificación*, xxix, 2, 3, 5, 173, 174, 175, 182, 189
- Ludificación BLAP*, 184, 185
- Ludificación empresarial*, 184
- Ludificación externa*, 184
- Ludificación interna*, 184
- Ludificación significativa*, 184, 185, 186, 199, 206, 207, 239
- Ludificación social*, 184, 185

- Ludología*, 5, 125, 126, 127, 130
- Ludología y Ludoterapia (artículo)*, 125
- Ludólogo*, 125
- Ludoterapeuta*, 125
- Ludus (forma de juego)*, 142, 143, 148
- Lunazzi, J. M.*, 125
- LWJGL - Lightweight Java Game Library (biblioteca de funciones)*, 169
- Lynda.com (plataforma para aprender online)*, 45
- Lyon, M.*, 11
- M**
- M.U.L.E. (videojuego)*, 117
- macOS (sistema operativo)*, 163
- MAGICAL (Making Games in Collaboration for Learning)*, 160, 161
- Magnavox (empresa)*, 115
- Maigaard, P.*, 126
- MakeCode for Minecraft (editor de código)*, 149
- Malinowski, B.*, 104
- Malone, T. W.*, 68
- Maloney, J.*, 34
- MaMaMedia.com (plataforma para aprender online)*, 32
- Manchester Metropolitan University*, 160
- Manejo emocional*, 89, 94
- Maneva, N. M.*, 28
- Maniac Mansion (videojuego)*, 120
- Mann, E. R.*, 113
- Máquina de Turing (dispositivo)*, 22
- Máquina física*, 73
- Máquina nocional*, 73
- Marcano Lárez, B. E.*, 133
- Marco de trabajo (framework)*, 168, 169, 170, 171
- Mario Kart 64 (videojuego)*, 117
- Mario Teaches Typing (videojuego)*, 117
- Martin, N. L.*, 167
- Martinez, J. C.*, 192
- Martino, G.*, 38
- Mason, B.*, 104
- Math Blaster (videojuego)*, 135
- Matriz sociométrica*, 82, 85
- May, J.*, 58, 59
- May, L.*, 191
- Mayer, J. D.*, 87, 90, 91, 92, 93, 94, 99
- Maze Craze (videojuego)*, 117
- Mazmorra (escenario de algunos juegos)*, 121, 124, 148
- McCarthy, J.*, 114
- McCormick, R.*, 62
- McCracken, M.*, 36, 37, 61, 62, 63
- McDougall, W.*, 104
- McLuhan, H. M.*, 105
- McNamara, W.*, 71
- MDA (Mechanics, Dynamics, and Aesthetics)*, 178
- Mead, G.*, 104

- Means, B. M.*, 160
- MediaComp - Media Computation (enfoque pedagógico)*, 37
- Medialab-Prado (centro cultural)*, 127
- Meike, R.*, 46
- Mellis, D.*, 38
- Membrey, P.*, 46
- Memotest (juego)*, 164
- Mentalidad de diseñador*, 156
- Merit Inc. (empresa)*, 31
- Merwin-Daggett, M.*, 12
- Metacognición*, 159
- Metacritic (sitio web de reseñas)*, 121
- Método (comportamiento de un objeto)*, 50, 51
- Métodos de investigación mixtos*, 198
- Meyer, B.*, 28
- Meyer, J.*, 164
- Michael, D. R.*, 141
- Michabelles, F.*, 38
- Microcomputadora*, 18, 19, 20, 21
- Microcontrolador*, 46
- Micromundo*, 29, 136
- Microsoft Corporation (empresa)*, 21, 23, 27, 31, 43, 149, 168
- Microsoft Excel (planilla de cálculo)*, 98, 188, 204, 216
- Microsoft Research (división de Microsoft Corporation)*, 42
- Mikulic, I. M.*, 93
- Mimicry (principio que rige algunos juegos)*, 110
- Minds in Play (libro)*, 151, 153, 157
- Mindstorms: children, computers, and powerful ideas (libro)*, 66, 153
- Minecraft (videojuego)*, 148, 149
- Minijuego*, 135, 136, 145, 146, 164, 165
- Miniproyecto de laboratorio*, 61, 63
- MissionMaker (entorno de desarrollo)*, 163
- MissionPlayer (entorno de ejecución)*, 163
- MIT (Massachusetts Institute of Technology)*, 11, 12, 14, 17, 22, 23, 27, 32, 47, 114, 122
- MIT Media Lab*, 26, 29, 38, 41, 48
- MIT OpenCourseware (plataforma para aprender online)*, 45
- Mitchell, E.*, 104
- Mitchell, N.*, 191
- Modding (acción de modificar)*, 149, 154, 156
- Modelado 3D*, 35, 47
- Modelado de rendimiento*, 54
- Modelo en cascada (metodología de desarrollo de software)*, 16
- Modelo G/P/S (taxonomía de juegos serios)*, 142
- Modelos de inteligencia emocional mixtos*, 99, 100
- Modelos didácticos para la programación*, 57
- Mojang AB (empresa)*, 148

- Molnar, C.*, 17
- Mondolfo, R.*, 1
- Monitor orientado a gráficos*, 18
- MonoGame (framework)*, 170, 171
- Montana (videojuego)*, 117
- Montaño, E.*, 147
- MOOC (Massive Open Online Course)*, 194
- Moodle (sistema de gestión del aprendizaje)*,  
188, 192, 194
- Moon, J.*, 45
- Mora Carreño, A.*, 183, 192
- Morales Moras, J.*, 142
- Morán, R.*, 149
- Moreno, E.*, 146, 194
- Moreno, J.*, 147
- Moreno, J. L.*, 75, 85
- Morris, B. A.*, 131
- Morris, C. W.*, 57
- Morris, D.*, 41
- Morrison, B.*, 10
- Mortal Kombat (videojuego)*, 117
- Moss, F.*, 26
- Motivación*, xxix, 68, 137, 167
- Motivación extrínseca*, 68
- Motivación intrínseca*, 69, 185, 186
- Motor de videojuego (videogame engine)*,  
166, 168, 170, 171
- Motorola 68000 (microprocesador)*, 21
- Mouse (ratón)*, 18, 25, 37, 153, 162
- Mousetrap (videojuego)*, 117
- MSCEIT (Mayer-Salovey-Caruso  
Emotional Intelligence Test)*, 93, 94
- MS-DOS (sistema operativo)*, 20, 25
- MSX (estándar de microcomputadoras)*, 21
- MUA - Multi-User Adventure (aventura  
multiusuario)*, 124
- MUD - Multi-User Dungeon (mazmorra  
multiusuario)*, 124
- Multi-Health Systems Inc. (empresa)*, 93,  
100
- MultiLogo (lenguaje)*, 29
- Multiplication game (juego)*, 165
- Multi-User Entertainment Ltd. (empresa)*,  
124
- Mumuki (plataforma argentina para  
aprender a programar online)*, 49
- Mundo abierto*, 121, 148, 149
- Mundo virtual 3D*, 149, 163
- Munné, F.*, 103, 104
- Music Box Demo (Coleco ADAM)*, 117
- Myst (videojuego)*, 117
- ## N
- Nacke, L.*, 175
- Nagle, J.*, 39
- Name That Tune (videojuego)*, 117
- Narratología*, 126, 130
- NASA (National Aeronautics and Space  
Administration)*, 135
- National Children's Museum*, 20

*National Council of Teachers of Mathematics*,  
43

*National Geographic Society (organización  
sin fines de lucro)*, 135

*National Science Foundation (agencia  
gubernamental)*, 31, 153, 161

*National Science Teachers Association*, 43

*Nativo digital*, 134

*Naturaleza del juego*, 103

*Navarro, E. O.*, 144

*Navegador web*, 42, 47

*NCWIT (National Center for Women &  
Information Technology)*, 43

*Needle, D.*, 21

*Nelson, M.*, 37

*Nesta - National Endowment for Science,  
Technology and the Arts (organización  
beneficiante)*, 9

*Newsweek (revista)*, 122

*Next Gen (informe)*, 9

*Next Generation (revista)*, 138, 139

*Nichols, R.*, 108

*Nicholson, S.*, 184, 185, 186

*Nickerson, R. S.*, 66

*Nieves, J.*, 25, 30, 31

*Night Driver (videojuego)*, 117

*Nike+ (aplicación que estimula a correr)*, 176

*Nim (juego)*, 113, 165

*Nimrod (computadora)*, 113

*Nisnevich, A.*, 148

*Nivel (elemento del diseño de juegos)*, 179,  
180, 181, 182, 185, 189, 190, 193, 194

*No One Left Behind/Nadie se queda atrás  
(proyecto de la Unión Europea)*, 48

*Norman, M. D.*, 149

*Notación*, 12, 73

*NTT Data Corporation (empresa)*, 177

*Nuevo latín*, 9

*Nutting Associates (empresa)*, 114

## O

*Objective-C (lenguaje)*, 30, 170

*Object-Oriented Software Construction  
(libro)*, 28

*Objetivos de la ludificación*, 176

*Objeto (instancia de una clase)*, 49, 145

*Observatorio Permanente Software y  
Servicios Informáticos (iniciativa de la  
Cámara de la Industria Argentina del  
Software)*, 54

*Obstáculos para el aprendizaje de la  
programación*, 68

*Obstáculos para el aprendizaje mediante el  
desarrollo de videojuegos*, 153

*OCDE (Organización para la Cooperación y  
el Desarrollo Económicos)*, 25, 26

*Ocko, S.*, 26

*Odin (servicio de ludificación online)*, 194

*Odyssey (consola de videojuegos)*, 115

- Oficina de Estadísticas Laborales de Estados Unidos*, 54
- OLPC (One Laptop Per Child)*, 34
- Olsen, K.*, 18
- Ontogénesis (concepto de la teoría de la recapitulación)*, 104
- Ordenamiento por burbujeo (algoritmo)*, 145
- Ordenamiento por inserción (algoritmo)*, 145
- Ordenamiento por selección (algoritmo)*, 145
- Orejuela, H. F.*, 191
- Osborne, M.*, 30
- Osorio, J.*, 25, 30, 31
- OTAN (Organización del Tratado del Atlántico Norte)*, 15
- Oulton, M. L.*, 129
- Oxford Dictionaries*, 174
- OXO (juego)*, 113
- Ozobot (robot)*, 144
- P**
- Pac-Man (videojuego)*, 116, 117, 119, 120, 171, 172
- Paidia (forma de juego)*, 142, 143, 148
- País Vasco*, 25
- Paisley, W.*, 19, 20
- Paiva, J. C.*, 194
- Pajek (software para ARS)*, 75
- Palfai, T. P.*, 91
- Paperboy (videojuego)*, 119
- Papert, S.*, 14, 19, 22, 26, 29, 41, 66, 151, 152, 153, 154
- Paralelismo*, 29
- Parker, J. R.*, 165
- Pascal (lenguaje)*, 15, 16, 22, 23, 28, 31, 33, 35, 36, 40, 166
- Patella-Rey, P. J.*, 187
- Patrones de diseño*, 33, 192, 193
- Pattis, R. E.*, 22
- Pausch, R.*, 35, 39, 40
- PBL (Points, Badges, Leaderboards)*, 180, 181
- PC (computadora)*, 42, 163
- Pea, R. D.*, 46, 55, 64, 65, 66, 67, 160
- PeerSpace (entorno de aprendizaje online)*, 190, 191
- Peixoto de Queirós, R. A.*, 194
- Pellas, N.*, 150
- Pelling, N.*, 174
- Pencil Code (plataforma para aprender a programar online)*, 47
- Pensamiento computacional*, 2, 3, 10, 22, 41, 42, 46, 48, 54, 55, 154, 159
- Pensamiento cuasiprocedimental*, 67
- Pensamiento de diseño*, 156
- Pensamiento procedimental*, 64, 66
- Pensamiento sistémico*, 159, 163
- Percepción emocional*, 89, 93
- Pérez Cerrato, L.*, 149
- Pérez Escoda, N.*, 101

- Pérez, M. O.*, 12, 14, 101, 149
- Pérez, O.*, 127
- Pérez-González, J. C.*, 94, 98
- Perfil psicológico*, 70
- Perlis, A.*, 12
- Peroutseas, E.*, 150
- Perron, B.*, 109
- Petaco (pinball)*, 112, 117
- Petrides, K. V.*, 88, 94, 95, 96, 97, 98, 99, 212
- Phaser (framework)*, 170
- PHP (lenguaje)*, 44
- Phylum (concepto de la teoría de la recapitulación)*, 104
- Piaget, J.*, 14, 65, 104, 152
- Piattini, M.*, 150
- Pidgin (idioma)*, 14
- Piedra, papel o tijera (juego)*, 164, 165
- Pig-2 (lenguaje)*, 11
- Pinball (máquina de petacos)*, 112
- Pinball Fantasies (videojuego)*, 117
- Pitfall! (videojuego)*, 117
- PL/I (lenguaje)*, 16
- Plagio*, 63, 64, 192
- Plan 111 Mil (programa de capacitación en programación)*, 49
- Planells de la Maza, A. J.*, 127
- Planetfall (videojuego)*, 117
- PLANIED (Plan Nacional Integral de Educación Digital)*, 2
- Platón*, 1, 103
- Playful Invention Company (empresa)*, 48
- PlayStation (consola de videojuegos)*, 122, 163
- PNIDE (Plan Nacional de Inclusión Digital Educativa)*, 1
- Pocket Billiards! (videojuego)*, 117
- Pocket Code (entorno de desarrollo)*, 48, 163
- Pointsification (puntificación)*, 186, 187
- Pole Position (videojuego)*, 117
- Polimorfismo (pilar de la programación orientada a objetos)*, 50, 52, 145, 146
- Polonia*, 36
- Pong (videojuego)*, 115, 116, 117
- Popović, Z.*, 150
- Popplewell, C.*, 11
- Portuondo, J. A.*, 75, 79, 84, 86
- Power, M.*, 68
- Pragmática*, 57, 58, 64, 73
- Prefetching (búsqueda anticipada)*, 54
- Prensky, M.*, 134, 135, 136
- Primera revolución electrónica*, 19
- Primo o no primo (juego)*, 165
- private (calificador de acceso)*, 51
- Pro Evolution Soccer (videojuego)*, 119
- Problems and Programmers (juego)*, 144
- ProBot (videojuego)*, 147
- Procesamiento de textos*, 18, 64
- Proceso ágil (metodología de desarrollo de software)*, 34



- Proceso Unificado de Desarrollo Software*  
(metodología de desarrollo de software), 36
- Processing* (lenguaje), 38
- Productividad*, 55, 56, 180, 184
- Program.AR* (iniciativa argentina para llevar la programación a las aulas), 1
- Programa Codo a Codo* (programa de capacitación en programación), 49
- Programa para Equidad de Género en Ciencia, Matemáticas, Ingeniería y Tecnología*, 161
- Programación* (concepto), 10
- Programación colaborativa*, 47, 190
- Programación como un oficio*, 45
- Programación conversacional*, 44
- Programación declarativa*, 16
- Programación estructurada*, 147
- Programación funcional*, 29, 171
- Programación lógica*, 16
- Programación orientada a objetos*, 18, 20, 27, 28, 29, 32, 49, 145, 163, 171
- Programación táctil*, 33
- Programación y Comunicaciones* (asignatura), 9
- Prolog* (lenguaje), 16
- Prop Cycle* (videojuego), 117
- protected* (calificador de acceso), 51
- Protocolo o ficha técnica* (parte del test sociométrico), 77, 78
- Proyecto Atenea*, 24, 25, 26
- Prueba ANCOVA* (análisis de covarianza), 238
- Prueba ANOVA* (análisis de varianza), 239
- Prueba de aptitud*, 70, 71
- Prueba de ensayo*, 61, 62
- Prueba de independencia*  $\chi^2$ , 223
- Prueba de rangos con signo de Wilcoxon para muestras relacionadas*, 247, 310, 315
- Prueba de Shapiro-Wilk*, 241
- Prueba objetiva*, 61, 62
- Prueba t para muestras relacionadas*, 247, 310
- public* (calificador de acceso), 51
- Puccini, G.*, 122
- Puntaje* (elemento del diseño de juegos), 179, 180, 181, 182, 185
- Puntificación* (pointsification), 186, 187
- Pygame* (biblioteca de funciones), 169, 172
- Python* (lenguaje), 30, 33, 35, 37, 39, 44, 147, 149, 165, 169, 172
- ## Q
- Q\*bert* (videojuego), 117
- Quake Arena* (videojuego), 119
- Queensland University of Technology*, 140
- Quirky* (empresa), 177
- ## R
- r* (índice estadístico), 249, 315
- RadioShack* (empresa), 19

- Radusky, P. D.*, 93
- Railroad Tycoon (videojuego)*, 117
- Ramírez Cogollor, J. L.*, 179, 180, 181
- Ramos, N.*, 91
- Randel, J. M.*, 131
- Ranking (elemento del diseño de juegos)*, 179, 180, 181, 182, 185, 188, 192, 194
- RARS (Robot Auto Racing Simulator)*, 117
- Raspberry Pi (computadora de placa única)*, 46
- Raspberry Pi Foundation*, 46
- Ratón (mouse)*, 18, 25, 37, 153, 162
- Razonamiento abstracto*, 70
- Razonamiento analógico*, 64, 65
- Razonamiento condicional*, 64, 66
- Razonamiento temporal*, 64, 67
- Razones para la eficacia del Aprendizaje Basado en Juegos Digitales*, 136
- Razones para la importancia de aprender programación*, 53
- RCX - Robotic Command Explorer (dispositivo)*, 35
- Re, S.*, 167
- Reader Rabbit (videojuego)*, 135, 136
- Realización (relación entre clases)*, 51
- Reas, C.*, 38
- Recapitulación (ley biogenética)*, 104
- RECIPE (receta para la ludificación significativa)*, 186
- Reclutamiento de personal*, 70
- Red de área local*, 18
- Reggini, H. C.*, 20, 22
- Reglas del ajedrez*, 23
- Reino Unido*, 9, 11, 21, 36, 40, 46, 48, 160, 162, 177
- Reiser, B. J.*, 24
- Relación (entre dos clases)*, 51
- Relaciones interpersonales*, 4, 5, 75, 76, 85, 86, 101
- Remezclado de software*, 157
- Renandya, W. A.*, 68
- Repenning, A.*, 33, 42, 44
- Requisitos cognitivos para el aprendizaje de la programación*, 64
- Resistencia interna*, 81
- Resnick, M.*, 26, 29, 41
- Resolución 263/15 del Consejo Federal de Educación (legislación argentina)*, 1, 9
- Resolución 441/2000 del Ministerio de Educación de la Nación (legislación argentina)*, 1
- Resolución de problemas*, 55, 57, 61, 65, 99, 137, 142, 159, 160, 166, 194
- Reversi (juego)*, 165
- Revilla, M. A.*, 33
- Revuelta Domínguez, F. I.*, vii, 111, 136, 137
- Richardson, C.*, 149
- Riera, D.*, 183

- rise.global Scorecard (herramienta para ludificación)*, 188
- Ritchie, D. M.*, 27
- Ritmo del curso*, 68, 72
- RML 380Z (computadora)*, 21
- RML 480Z (computadora)*, 21
- RML Nimbus (computadora)*, 21
- Roberts, M.*, 76
- Robertson, D.*, 34
- Robertson, M.*, 187
- Robins, A.*, 68
- Robo (lenguaje)*, 40
- Robocode (videojuego)*, 37, 146
- RoboMind (videojuego)*, 40
- Robot Battle (videojuego)*, 117
- Robótica*, 26
- Rodríguez, F.*, 180
- Rodríguez, R. A.*, 146, 194
- Rogers Majestic Corporation Ltd. (empresa)*, 113
- Roschelle, J. M.*, 160
- Rosenberg, J.*, 33
- Ross, J.*, 20
- Rountree, J.*, 68
- Rountree, N.*, 68
- Roussel, P.*, 16
- Roy, S. R.*, 87
- Royce, W. W.*, 16
- RPG Maker (entorno de desarrollo)*, 163
- RSDE (Research Software Design Engineer)*, 149
- Ruby (lenguaje)*, 44
- Rumbaugh, J.*, 34, 36, 51
- Russell, J. P.*, 32
- Russell, S.*, 114
- ## S
- S4SL - Scratch for Second Life (lenguaje)*, 150
- Saarni, C.*, 101
- Sadosky, M.*, 11
- Salen, K.*, 110
- Salovey, P.*, 87, 90, 91, 92, 93, 94, 99
- Samsung (empresa)*, 42
- Sandbox (arenero)*, 148
- Sanders Associates (empresa)*, 115
- Santiago, R.*, 180
- Sapora, A.*, 104
- SAS Institute Inc. (empresa)*, 43
- Scalable Game Design (enfoque pedagógico)*, 42
- Schaller, J.*, 104
- Scheller Teacher Education Program (MIT)*, 47
- Scheme (lenguaje)*, 23, 163
- Schiller, F.*, 104
- ScienceDirect (base de datos online)*, 150, 188
- Scopus (base de datos online)*, 150, 183, 188
- Scott, M. J.*, 2, 72

- Scratch* (entorno de desarrollo), 41, 48, 162, 163
- ScratchJr* (entorno de desarrollo), 48
- Scriven, M.*, 62
- SDC - System Development Corporation* (empresa), 70
- SDK (Software Development Kit)*, 168
- SDL - Simple DirectMedia Layer* (biblioteca de funciones), 169
- Sebesta, R. W.*, 12, 15, 18
- Second Life* (videojuego), 150
- Secuencialidad*, 67
- Segunda revolución de las computadoras en la educación*, 19
- Segunda revolución electrónica*, 19
- Seldes, G.*, 122, 123
- Selection Sort* (videojuego), 145
- Self-social-distance* (valor sociométrico), 84
- Semana Europea de la Programación (EU Code Week)*, 47
- Semántica*, 30, 44, 57, 58
- Semiosis* (lingüística), 57
- Serious games* (juegos serios), xxix, xxxi, 2, 5, 140, 142, 175
- Serious Games Initiative*, 140
- Serrano, M. I.*, 9
- SFML - Simple and Fast Multimedia Library* (biblioteca de funciones), 169
- Shabanah, S.*, 145
- Shapiro, N.*, 21
- Sheikh, R.*, 162
- Sheng, Y.*, 141
- Sherrod, A.*, 168, 170
- Shooting Gallery* (videojuego), 117
- Short* (clase de envoltorio), 53
- short* (tipo primitivo), 53
- Shuflin, G.*, 148
- Sid Meier's Civilization* (videojuego), 117
- Sierra, K.*, 205
- SimCity* (videojuego), 117, 120
- Simões, A.*, 169, 171
- Sims 2* (videojuego), 119
- Sims, Z.*, 44
- Simulación*, 40, 47, 113, 163
- Sinclair QL* (computadora), 21
- Sinclair Spectrum* (computadora), 21
- Sinclair ZX81* (computadora), 21
- Sintaxis*, 30, 33, 57, 58, 62, 70, 167
- Sistema de servidores múltiples*, 54
- Slany, W.*, 48
- Slawson, S.*, 104
- Slot Machine* (videojuego), 117
- Smalltalk* (lenguaje), 18, 20, 34, 149
- Smart TV*, 42
- Smedstad, S. M.*, 118, 119
- Smith, D. K.*, 18
- Smith, J. H.*, 120
- Smith, M.*, 10
- Snap!* (entorno de desarrollo), 163
- Soares, A.*, 167

- Sobrecarga*, 53, 145
- Sobrescritura (de métodos)*, 52
- Socializador/Socialiser (tipo de jugador)*, 124
- Sociograma*, 85
- Sociomatriz*, 82, 83, 85
- Sociometría*, 75, 76, 78, 81, 85, 86
- Software comercial*, 163
- Software gratuito*, 163
- Software libre*, 29, 48
- Sokoban (videojuego)*, 117, 171
- Solaris (videojuego)*, 117
- Solomon, C.*, 14
- Sonoma State University*, 101
- Space Ace (videojuego)*, 117
- Space Invaders (videojuego)*, 117
- Spacewar! (videojuego)*, 114
- Spaceward Ho! (videojuego)*, 117
- Spencer, H.*, 104
- Spertus, E.*, 44
- Sploder (entorno de desarrollo)*, 163
- Sprak (lenguaje)*, 150
- Spring Joint Computer Conference*, 12
- Springer Link (base de datos online)*, 150, 188
- Squeak (entorno de desarrollo)*, 34
- Squire, K.*, 136
- Stake, R. E.*, 62
- Stallings, W.*, 30
- Stampede (videojuego)*, 117
- Stamper, R.*, 58
- Stanchev, L.*, 165
- Stanford University*, 46
- Stanley, J. C.*, 201
- Star Wars (película)*, 147
- StarCraft (videojuego)*, 118, 119
- StarLogo (lenguaje)*, 29, 34
- StarLogo Nova (entorno de desarrollo)*, 47
- Starmaster (videojuego)*, 117
- Steblik, M.*, 10
- Stellar Track (videojuego)*, 117
- STEM (Science, Technology, Engineering, and Mathematics)*, 10
- STEM Education Act (legislación estadounidense)*, 10
- Stencyl (entorno de desarrollo)*, 163
- Stephenson, C.*, 10
- Sticks (juego)*, 165
- STL - Standard Template Library (biblioteca de funciones)*, 169
- Strachey, C.*, 113
- Stroustrup, B.*, 27
- Structure and Interpretation of Computer Programs (libro)*, 23
- Stupuriene, G.*, 39
- Sudol, L. A.*, 10
- Summer Games (videojuego)*, 117
- Sun Microsystems (empresa)*, 32
- Sunnanå, L.*, 118, 119
- Super Mario Bros. (videojuego)*, 117
- Sussman, G. J.*, 23

Sussman, J., 23

Sutton-Smith, B., 105

Swacha, J., 192

Swaine, M., 20, 23, 27

## T

Tabla hash (estructura de datos), 169

Table Tennis (videojuego), 115

Tampere University of Technology, 160

Tandy TRS-80 (computadora), 18

Tandy TRS-80 Color Computer 3  
(computadora), 20

Tanenbaum, A. S., 12

TankWar (videojuego), 166

Tate, L., 96

Taxonomía de Bartle, 125

Taxonomía de entornos y lenguajes, 40

Taxonomía de géneros de los videojuegos, 116,  
118, 120, 121

Taxonomía de juegos serios, 142

Taxonomía de jugadores, 124

Taxonomía de objetivos cognitivos, 57, 59,  
60, 61

Technische Universität Graz, 48

Tecnológico de Antioquía, 145

TEIQue (Trait Emotional Intelligence  
Questionnaire), 95, 96, 97, 98, 99

TEIQue 360°, 99

TEIQue 360°-SF (Short Form), 99

TEIQue-AF (Adolescent Form), 99

TEIQue-ASF (Adolescent Short Form), 99

TEIQue-CF (Child Form), 99

TEIQue-CSF (Child Short Form), 99

TEIQue-SF (Short Form), 99

Teixes, F., 124, 125

Tennis for Two (juego), 113

Teoría literaria, 126, 130

Teragni, M., 149

Test de ejecución típica, 89

Test de rendimiento máximo, 88, 89, 96

Test sociométrico, 5, 76, 77, 78, 81, 82, 83,  
86

Tetris (videojuego), 117, 119, 165, 171

Texas A&M University, 16

Texas Instruments Inc. (empresa), 20

The 7 Lively Arts (libro), 122

The Game of Work (empresa), 180

The Game of Work: How to enjoy Work as  
much as Play (libro), 180

The Hitchhiker's Guide to the Galaxy  
(videojuego), 117

The Little Computer People (videojuego), 117

The Oregon Trail (videojuego), 141

The Price is Right (videojuego), 117

The Starter League (campo de entrenamiento  
de codificación), 45

The Unified Software Development Process  
(libro), 36

Think of a number (juego), 165

Thomas International Ltd. (empresa), 95, 98

- TI-99/4A (computadora), 20
- TIC (Tecnologías de la Información y la Comunicación), 31, 160
- Tienda en línea, 32
- Tilting (acción de inclinar), 112
- Tipo primitivo, 53
- Tiro de cañón (juego), 164
- TIS-100 (videojuego), 148
- TMMS (Trait Meta-Mood Scale), 90, 91
- TMMS-21 (TMMS de 21 ítems), 91
- TMMS-24 (TMMS de 24 ítems), 91
- Toma de decisiones, 137, 142, 159
- Tomorrow Corporation (empresa), 148
- ToonTalk (entorno de desarrollo), 163
- ToonTalk Reborn (entorno de desarrollo), 163
- TopView (interfaz gráfica de usuario), 27
- Tortuga (dispositivo), 14, 47
- Tosca, S. P., 120
- Trabajo para hacer en casa, 61, 62, 63
- Trading Game (juego), 165
- TRAKLA2 (sistema de gestión del aprendizaje), 193
- Tran, K. M., 156
- Treehouse (plataforma para aprender a programar online), 45
- Trefftz Gómez, H., 145
- Tres en raya (juego), 113, 164, 165
- Trinidad de 1977, 18, 20
- Tufts University, 48
- Turbo Pascal (entorno de desarrollo), 23, 26
- Turvey, C., 91
- Tutor de programación, 24
- Tynker (plataforma para aprender a programar online), 45, 163
- TypeScript (lenguaje), 170
- Typing game (juego), 165
- ## U
- Ubuntu (sistema operativo), 163
- Udacity (plataforma para aprender online), 45
- Udemy (plataforma para aprender online), 45
- Ultima (videojuego), 117
- UML (Unified Modeling Language), 34, 36, 49, 51
- UNED (Universidad Nacional de Educación a Distancia), 98
- UNESCO (United Nations Educational, Scientific and Cultural Organization), 24, 28
- UNICEF - United Nations International Children's Emergency Fund, 135
- Unión Europea, 48, 56
- Unión Soviética, 28
- Unity (motor de videojuego), 171
- Universidad Abierta Interamericana, 128
- Universidad Argentina de la Empresa, 128
- Universidad de Buenos Aires, 11, 14, 16
- Universidad de Castilla-La Mancha, 150
- Universidad de La Punta, 128

- Universidad de Mendoza*, 128
- Universidad de Palermo*, 128
- Universidad de Sevilla*, 127
- Universidad de Valladolid*, 33
- Universidad EAFIT*, 145
- Universidad Nacional de Colombia*, 147
- Universidad Nacional de Educación a Distancia*, 98
- Universidad Nacional de José C. Paz*, 128
- Universidad Nacional de La Matanza*, 146
- Universidad Nacional de Rafaela*, 128
- Universidad Nacional del Litoral*, 128
- Universidad Tecnológica Nacional*, vii, xxix, 3, 209
- Universitat Pompeu Fabra*, 127
- Université de Aix-Marseille*, 16
- University College London*, 94, 98, 212
- University of California, Berkeley*, 27, 148
- University of California, Irvine*, 75, 144
- University of California, Los Angeles*, 151
- University of Cambridge*, 113
- University of Chicago*, 59
- University of Colorado*, 33
- University of Kent*, 40
- University of Manchester*, 11, 113
- University of New Hampshire*, 90
- University of Pennsylvania*, 182
- University of Sydney*, 33
- University of Vilnius*, 39
- UNIX (sistema operativo)*, 32
- Unreal Engine (motor de videojuego)*, 171
- Untch, R. H.*, 190
- Untrusted: The Continuing Adventures of Dr. Eval (videojuego)*, 148
- Uppal, G.*, 41
- Upton, E.*, 46
- UVa Online Judge (juez online que corrige automáticamente problemas de programación competitiva)*, 33
- ## V
- V de Cramér*, 224
- Valles, F. E.*, 194
- Valor sociométrico*, 83, 84, 85, 86
- Van Der Hoek, A.*, 144
- Van Eck, R.*, 133
- Vardalas, J. N.*, 113
- Vargas-Enríquez, J.*, 150
- Vázquez, M. C.*, 194
- Vee, A.*, 10
- Vegso, J.*, 39
- Vehículo sígnico (componente de la semiosis)*, 57
- Vera, P. M.*, 146, 194
- Verry, T.*, 46
- Vidal Duarte, E.*, 172
- Video Checkers (videojuego)*, 117
- Video Chess (videojuego)*, 117
- Video Poker (videojuego)*, 117



- Videojuego*, xxix, 2, 107, 133, 136, 141, 142, 162
- Videotexto (servicio de información online)*, 19
- Vilensky, C.*, 11
- Vinikiene, L.*, 39
- Virtual Pool (videojuego)*, 117
- VisiCalc (planilla de cálculo)*, 27
- VisiCorp (empresa)*, 27
- VisiOn (interfaz gráfica de usuario)*, 27
- Visual AgenTalk (entorno de desarrollo)*, 33
- Visual Basic (entorno de desarrollo)*, 31, 35, 150
- Vlissides, J.*, 33
- Voki (herramienta para ludificación)*, 188
- Vygotski, L.*, 105, 111, 160
- ## W
- Wallace, S.*, 34
- Walt Disney Imagineering Research & Development, Inc. (empresa)*, 34
- Web (sistema de documentos accesibles en Internet a través de navegadores)*, 31, 32, 33, 34, 40, 42, 47, 163, 166, 174, 188, 193, 194
- Web of Knowledge (base de datos online)*, 183
- Webb, D.*, 42
- Welchman, G.*, 11
- Wells, D.*, 41
- Werbach, K.*, 175, 182, 183, 184, 185, 186, 187
- Wetzel, C. D.*, 131
- What makes a game good? (artículo)*, 139
- What video games have to teach us about learning and literacy (libro)*, 135
- Whatsapp (aplicación de mensajería)*, 188
- Whitehill, B. V.*, 131
- Wiki (comunidad virtual cuyas páginas pueden ser editadas directamente desde el navegador web)*, 179
- Wiley InterScience (base de datos online)*, 150
- Wilkinson, B.*, 149
- Williams, N.*, 149
- Wilson, C.*, 10
- Windows (sistema operativo)*, 27, 163, 164, 168
- Wing, J. M.*, 2, 41, 46, 54
- Wingrave, C. A.*, 149
- Winnicott, D.*, 105
- Wiring (plataforma de prototipado electrónico)*, 38
- Wirth, N.*, 15
- Wolber, D.*, 44
- Wolf, M. J. P.*, 108, 109, 116, 117
- Woodrow Wilson International Center for Scholars*, 140
- Word Jumble (juego)*, 164
- World Future Society*, 18
- World Wide Workshop Foundation*, 40

*Wrapper class (clase de envoltorio)*, 53

*Wriston, W. B.*, 18

## X

*X3DOM (biblioteca de funciones)*, 171

*Xbox (consola de videojuegos)*, 42, 163, 168

*Xerox Corporation (empresa)*, 17, 18

*Xerox PARC (Palo Alto Research Center)*,

17, 18

*Xiphias (sistema de ludificación para evaluar programación)*, 192

*XO (computadora)*, 34

## Y

*Yale University*, 90

*Yaroslavski, D.*, 41

## Z

*Zabala, G.*, 149

*Zachtronics (empresa)*, 148

*Zakauskas, R.*, 39

*Zichermann, G.*, 175

*Zimmerman, E.*, 110, 158

*Zoltan, C.*, 12

*Zork (videojuego)*, 117

*Zorn, C.*, 149

*Zubek, R.*, 178

*Zyda, M.*, 141

