



UNIVERSIDAD DE EXTREMADURA
CENTRO UNIVERSITARIO DE MÉRIDA

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO FIN DE GRADO

E^XTREBOOKS: SUPERVISIÓN Y CONTROL DE MATERIALES
CURRICULARES EN LOS DIFERENTES NIVELES
EDUCATIVOS EN EXTREMADURA

AUTOR: ISIDRO REGAÑAS MOLINA

Mérida, Julio de 2015



UNIVERSIDAD DE EXTREMADURA
CENTRO UNIVERSITARIO DE MÉRIDA

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO FIN DE GRADO

EXTREBOOKS: SUPERVISIÓN Y CONTROL DE MATERIALES
CURRICULARES EN LOS DIFERENTES NIVELES
EDUCATIVOS EN EXTREMADURA

Autor: Isidro Regañas Molina
Fdo.:

Director: Luis J. Arévalo Rosado
Fdo.:

Director: D. Francisco Chávez de la O
Fdo.:

Resumen

En la actualidad, el desarrollo de las tecnologías de la información y comunicación, ha dado origen a nuevos enfoques en la concepción del proceso de enseñanza-aprendizaje. En vista de estos nuevos planteamientos, se hace necesario, diseñar una plataforma tecnológica, para la gestión de materiales curriculares en los centros educativos de la Comunidad Autónoma de Extremadura.

El propósito de esta plataforma, es posibilitar una colaboración estrecha y funcional entre el centro, los profesores y las familias, permitiendo reducir las carencias organizativas y didácticas, que supone todo un curso escolar e implicando un diálogo permanente y periódico entre todas las partes. En la actualidad, los centros educativos extremeños basan este diálogo en sistemas tan rudimentarios como las hojas del tablón de anuncios, cartas de correos convencional, notas informativas entregadas a los alumnos... presentando grandes deficiencias, como puede ser: consumo elevado de papel, difícil control por las familias, olvido de los alumnos, periodos vacacionales,

Este proyecto, consiste en aplicar las nuevas tecnologías para terminar con estos métodos informativos toscos y elementales, para lo cual se tendrá como objetivo, el análisis, diseño e implementación de un sistema software de supervisión y control de materiales curriculares, utilizando avances en dispositivos y tecnologías existentes.

En el desarrollo de este proyecto, se evaluaron múltiples patrones de arquitectura Web, resultando finalmente una estructura de MVC (modelo vista controlador), con funciones específicas e independientes entre sí, manteniendo las capas de presentación y acceso a datos separadas.

La implementación fue llevada a cabo mediante el IDE Eclipse y el lenguaje de programación Java. Para la construcción de la capa de presentación, se trabajó con el framework Bootstrap y controladores dinámicos de la librería de jQuery, Ajax, ...La capa de acceso a datos fue construida bajo la conexión con una base de datos MySQL.

Abstract

At present, development of information technology and communication (ICTs) has led to new approaches in the design of the teaching-learning process. From these new approaches, it is necessary to design a technology platform for the management of curriculum materials in schools in the Autonomous Community of Extremadura.

The purpose of this platform is to enable a close and practical cooperation between schools, teachers and families, thereby reducing organizational and educational deficiencies involving an entire school year and involving a permanent and regular dialogue between all of them. Currently, this dialogue in schools of Extremadura is based on a rudimentary systems such as leaves bulletin board, conventional post letters, briefing notes given to the students ... presenting major deficiencies such as: high consumption of paper, hard Control by families, students forget, holiday periods

The objective of this project is to apply new technologies in order to stop these crude methods and basic information, To do it, the analysis, design and implementation of a software system of supervision and control of curriculum materials and devices has been made.

In developing this project, multiple patterns of Web architecture has been evaluated, finally resulting in a structure of MVC (model view controller) with specific functions independent of each other, keeping the presentation layer and data access spread wide apart.

The implementation was carried out by the Eclipse IDE and the Java programming language. For building the presentation layer Bootstrap framework has been chosen as well as the dynamic driver library jQuery and Ajax, ... The data access layer was built under the connection with a MySQL database.

Agradecimientos

En primer lugar, quiero mostrar mi agradecimiento al Centro Universitario de Mérida por la oportunidad que me han brindado de seguir formándome como persona y profesional durante estos últimos cuatro años.

A mis directores de proyecto, D. Luis Jesús Arévalo Rosado y D. Francisco Chávez de la O, quiero agradecerle el apoyo y dedicación que me han dado durante los últimos meses. No sólo me han guiado en la elaboración de este trabajo de fin de grado, también me han enseñado todo lo necesario por si en un futuro quiero realizar trabajos semejantes.

A mis compañeros de aula y de profesión (Eduardo, Adrián, Elena y Jose) porque siempre que los he necesitado he podido contar con ellos. A mis amigos por escucharme cuando he necesitado consejo y ayuda.

Por último, pero no menos importante, me gustaría agradecer este trabajo a mi familia y en especial a mi mujer Ana y a mi hijo Álvaro por transmitirme su entusiasmo y energía por y para la educación, siendo siempre un apoyo para mí.

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Antecedentes	2
1.3. Objetivos de este proyecto fin de grado	4
2. Análisis Previo	7
2.1. Recursos para favorecer el aprendizaje	7
2.2. Portales de materiales curriculares	9
2.2.1. Agrega	9
2.2.2. Contenidos Educativos Digitales	9
2.2.3. Proyecto Medusa	10
2.2.4. Wikisaber	12
2.2.5. Otros portales	12
2.3. Identificación del entorno tecnológico	13
2.3.1. Herramientas utilizadas	16
3. Metodología de Desarrollo	23
3.1. Introducción	23
3.1.1. Análisis	23
3.1.2. Diagrama de caso de uso	24
3.1.3. Diagramas de secuencia	25
3.1.4. Diseño de la interfaz de usuario	25
3.1.4.1. Diseño de entradas y salidas	25
3.1.4.2. Diseño de entradas	25
3.1.4.3. Diseño de salidas	25
3.1.4.4. Diseño de entradas salidas	25
4. Análisis del Sistema	27
4.1. Planificación del sistema	28
4.1.1. Viabilidad técnica	28
4.1.2. Viabilidad operacional	28
4.1.3. Viabilidad económica	29
4.1.4. Viabilidad de la solución	29
4.2. Análisis funcional	30
4.2.1. Casos de uso global	30
4.2.2. Listado de casos de usos	35

4.3.	Diagramas de casos de uso detallados	36
4.3.1.	Casos de uso detallados del actor Administrador	36
4.3.2.	Casos de uso detallados del actor Equipo directivo	42
4.3.3.	Casos de uso detallados del actor Profesor	48
4.3.4.	Casos de uso detallados del actor Alumno	51
4.3.5.	Casos de uso detallados del actor Padre/Madre	51
4.3.6.	Casos de uso detallados del actor Móvil	53
4.4.	Diagramas de secuencia	53
4.4.1.	Diagrama de secuencia de la aplicación web	54
4.4.2.	Diagrama de secuencia de la aplicación móvil	54
5.	Diseño del Sistema	57
5.1.	Arquitectura del sistema	57
5.1.1.	Capa de Presentación	58
5.1.2.	Capa Lógica	59
5.1.3.	Capa de datos	59
5.2.	Diseño del sistema	59
5.2.1.	Diseño de la interfaz	60
5.2.2.	Acceso Web	60
5.2.2.1.	Requisitos mínimos	60
5.2.2.2.	Navegación por el sitio	61
5.2.2.3.	Interfaz sistema web del administrador	63
5.2.2.4.	Interfaz sistema web del equipo directivo	64
5.2.2.5.	Interfaz sistema web del profesor	66
5.2.2.6.	Interfaz sistema web del alumno	67
5.2.2.7.	Interfaz sistema web de la familia	68
5.2.3.	Acceso Móvil	69
5.3.	Base de datos	72
5.3.1.	Servidor de base de datos MySQL	72
5.4.	Diseño de la seguridad	76
5.4.1.	Seguridad en la base de datos	76
5.4.2.	Seguridad en las comunicaciones	76
6.	Implementación	81
6.1.	Análisis y diseño de la Arquitectura	81
6.1.1.	Capa de datos	82
6.1.2.	Capa lógica	84
6.1.2.1.	Entorno de desarrollo	84
6.1.2.2.	Servidor de aplicaciones	84
6.1.2.3.	Arquitectura de la capa lógica	85
6.1.2.4.	Diagrama de secuencia	89
6.1.2.5.	Particularidades de la implementación	89
6.1.3.	Capa de presentación	90
6.2.	Implementación de la seguridad del sistema	92
6.2.1.	Seguridad en la base de datos	92

7. Manual de Usuario	95
7.1. Aplicación Web	95
7.1.1. Usuario Administrador	95
7.1.2. Usuario Equipo directivo	96
7.1.3. Usuario Profesor	98
7.1.4. Usuario Alumno	99
8. Conclusiones y Trabajos Futuros	101
8.1. Conclusiones	101
8.2. Trabajos futuros	101
8.3. Planificación estimada vs Planificación Real	103
Bibliografía	105
9. Anexo I. Importación de la base de datos	107

Índice de figuras

1.1. Hoja informativa. IES Siberia Extremeña	3
2.1. Proyecto Agrega	10
2.2. Contenidos Educativos Digitales	11
2.3. Proyecto Medusa	11
2.4. Wikisaber.es	12
2.5. Arquitectura de un sistema de aplicaciones web	14
2.6. Diagrama Modelo-Vista-Controlador.[11]	15
3.1. Esquema de la metodología	24
4.1. Caso de uso del actor administrador	31
4.2. Caso de uso del actor equipo directivo	32
4.3. Caso de uso del actor profesor	33
4.4. Caso de uso del actor alumno	33
4.5. Caso de uso del actor padre/madre	34
4.6. Caso de uso del actor móvil	34
4.7. Diagrama de secuencia de la aplicación web	54
4.8. Diagrama de secuencia Android	55
5.1. Arquitectura de la aplicación web	58
5.2. Esquema web del sistema	61
5.3. Diseño de la interfaz de inicio de sesión	62
5.4. Diseño de la interfaz de la pantalla principal	62
5.5. Diseño de la interfaz de información	63
5.6. Diseño de la interfaz de alta del centro	63
5.7. Diseño de la interfaz de asignar equipo directivo	64
5.8. Diseño de la interfaz de listar tipo de materiales	64
5.9. Diseño de la interfaz de nuevo curso	65
5.10. Diseño de la interfaz de listar profesores	65
5.11. Diseño de la interfaz de editar asignaturas	66
5.12. Diseño de la interfaz de listar materiales	66
5.13. Diseño de la interfaz de alta alumno	67
5.14. Diseño de la interfaz de alta de familia	67
5.15. Diseño de interfaz de listar cursos académicos	68
5.16. Diseño de interfaz de listar asignaturas	68
5.17. Diseño de la interfaz de listar materiales	69
5.18. Diseño de la interfaz de listar asignaturas	69

5.19. Diseño de la interfaz de listar materiales	70
5.20. Diseño interfaz de listar materiales	70
5.21. Diseño de la interfaz inicio de sesión	71
5.22. Diseño de la interfaz de profesores y asignaturas	71
5.23. Diseño de la interfaz del centro	72
5.24. Diagrama entidad-relación de la base de datos	73
5.25. Modelo del sistema eXtreBooks	77
5.26. Seguridad ofrecida por SSL [16]	79
6.1. Arquitectura del sistema	82
6.2. Implementación de la base de datos	83
6.3. Arquitectura del sistema	85
6.4. Diagrama de secuencia aplicación web	89
6.5. Esquema de rejillas del servicio web	91
7.1. Página principal del usuario administrador	96
7.2. Página principal del usuario equipo directivo	98
7.3. Página principal del usuario equipo directivo	98
7.4. Página principal del usuario profesor	99
7.5. Página principal del usuario alumno	100
8.1. Diagrama de Gantt	104

Índice de tablas

4.1. Análisis del coste del desarrollo	30
4.3. Iniciar sesión	36
4.4. Cerrar sesión	36
4.5. Crear usuario de Equipo directivo	37
4.6. Modificar usuario	37
4.7. Listar usuarios	38
4.8. Eliminar usuario	38
4.9. Crear centro	38
4.10. Modificar centro	39
4.11. Listar centros	39
4.12. Eliminar centro	39
4.13. Crear material didáctico	40
4.14. Modificar material	40
4.15. Eliminar material	41
4.16. Listar materiales	41
4.17. Consultar materiales por tipo	41
4.18. Consultar materiales por centro	42
4.19. Crear curso	42
4.20. Modificar curso	43
4.21. Eliminar curso	43
4.22. Listar cursos	43
4.23. Crear asignatura	44
4.24. Modificar asignatura	44
4.25. Eliminar asignatura	45
4.26. Listar asignaturas	45
4.27. Asignar un curso a un profesor	45
4.28. Asignar una asignatura a profesor	46
4.29. Asignar un material a profesor	46
4.30. Asignar un material a un curso	47
4.31. Listar materiales por profesor	47
4.32. Listar materiales por asignatura	48
4.33. Listar materiales por curso	48
4.34. Listar cursos asignados	49
4.35. Listar asignaturas asignadas	49
4.36. Asignar material a una asignatura	49
4.37. Gestionar recomendación	50
4.38. Mostrar datos personales	50

4.39. Modificar datos personales	50
4.40. Listar profesores asociados	51
4.41. Recomendar material	51
4.42. Listar materiales por centro	52
4.43. Mostrar datos de alumno	52
4.44. Modificar datos del alumno	53
4.45. Login sistema	53
8.1. Planificación real	103

Algoritmos

6.1. Código de ejemplo del controlador Action.java	87
6.2. Código de ejemplo del patrón DAO	88
6.6. Sentencia SQL para insertar una clave cifrada	92
6.7. Sentencia SQL para descifrar una clave cifrada	92
6.3. Código ejemplo del modelo Material	93
6.4. Fichero context.xml	94
6.5. Esquema eXtreBooks template	94

Capítulo 1

Introducción

Contenidos

1.1. Introducción	1
1.2. Antecedentes	2
1.3. Objetivos de este proyecto fin de grado	4

Este capítulo se utiliza como introducción a este trabajo fin de grado. En él se abordan tres aspectos: introducción, antecedentes y los objetivos que se han perseguido.

En la sección 1.1 se introduce brevemente cómo se está realizando actualmente el control de materiales curriculares.

En la sección 1.2 se describen los antecedentes existentes en los Centros Educativos Extremeños.

En la sección 1.3 se especifican los objetivos de este trabajo fin de grado.

1.1. Introducción

Es en los centros educativos donde la amplia mayoría de los docentes y en general de la Comunidad Educativa (equipos directivos, personal laboral, padres y madres, ...) constatan frecuentemente las carencias organizativas y didácticas que supone no solo los inicios de cursos escolares, sino también la gran brecha de desinformación que los padres en particular denotan durante todo el curso escolar. Por ello, surge la necesidad de cambiar, porque en el siglo XXI, con Internet, con una sociedad dinámica como la nuestra, con la cantidad de materiales digitales existentes en múltiples plataformas y web sites, las innovaciones tecnológicas en el ámbito educativo, deben proporcionar a los docentes en general y a los padres en particular, nuevos canales de comunicación.

Para introducir al lector en este trabajo, debemos reflexionar sobre la siguiente pregunta: ¿Por qué nos resulta de interés pasar a informatizar el control de los libros de textos y materiales digitales?. La relación entre la familia y el centro educativo es fundamental para realizar una labor conjunta y eficaz, que implique un diálogo permanente y periódico entre ambas partes. Sin embargo, con respecto a los materiales curriculares que se utilizan en el día a día en el aula, incluidos los libros de textos, su selección la realiza cada departamento didáctico de cada centro y aparecen recogidos exclusivamente en las programaciones didácticas, incluidas

a su vez, en el Plan General Anual de Centro (PGA). Como el lector podrá discurrir, esta información permanece oculta a gran parte de la Comunidad Educativa.

La relación de libros de texto y demás materiales curriculares, se deben exponer en el tablón de anuncios de cada centro educativo antes del 30 de junio del curso anterior. Se debe, así mismo informar, a las asociaciones de padres y madres acerca de los libros de texto y los materiales didácticos adoptados por el centro. Sin embargo, ni se cumple en el tiempo, ni se cumple en las formas.

Este documento pretende materializar el trabajo realizado durante el desarrollo del trabajo de fin de grado denominado “eXtreBooks: APP para la supervisión y control de libros de textos en los diferentes niveles educativos en extremadura”. Se ha realizado la documentación conforme el modelo más realista posible, así como la descripción adecuada a cada una de las fases, por las que generalmente pasa un proyecto de estas características.

1.2. Antecedentes

En la actualidad, y durante muchos años, los centros educativos de nuestra comunidad utilizan un sistema tan rudimentario como son las hojas informativas o los tabloneros de anuncios colocados en el propio centro, para dar a conocer los materiales curriculares obligatorios para el curso escolar presente. Por ello, cuando los padres y/o madres de los alumnos necesitan conocer de antemano los materiales curriculares, obligatoriamente necesitan desplazarse al centro educativo para ser informados, puesto que el centro no proporciona esta información por ningún medio ordinario, como teléfono, email o correo tradicional. Además, determinados centros educativos con el fin de integrar las Tecnologías de la Información y la Comunicación, cuelgan en la página web del centro educativo un mero documento de texto con la información más relevante, que los padres y/o madres necesitan para la adquisición del material didáctico. Este modelo de relación de libros de textos, es un simple documento dividido en los siguientes campos:

- Curso. Determina el curso escolar de las asignaturas que se imparten.
- Asignatura. Nombre de la asignatura impartida por un profesor.
- Título. Nombre correspondiente al libro de texto.
- Editorial. Nombre de la compañía que publica el libro de texto.
- ISBN. Referencia numérica única para cada libro de texto.
- Otros campos. Cita determinados aspectos como préstamo, obligatoriedad, material digital, ...

El modelo del documento puede verse en la figura 1.1.

RELACIÓN LIBROS DE TEXTO PARA EL CURSO 2014/2015

CURSO	ASIGNATURA	TÍTULO	EDITORIAL	ISBN	Préstamo	
1º ESO	Ciencias de la Naturaleza	Ciencias de la Naturaleza 1º ESO Proyecto La Casa del Saber	Santillana	978-84-294-0959-8	*	
	Ed. Plástica y Visual	Ed. Plástica y Visual I	SM	978-84-675-0552-8	*	
	Inglés	My Life 1	Students' Book Pack	Pearson Longman	978-84-9837-355-4	*
			Workbook Pack (Castellano)	Pearson Longman	978-84-9837-354-7	
	Lengua Cast. y Literatura	Lengua Cast. y Literatura 1º ESO (Libro + monografía)	Oxford	978-84-673-3225-4	*	
	Matemáticas	Matemáticas 1º ESO	Anaya	978-84-667-5875-8	*	
	Religión Católica	Kairos	Magisterio Casals	84-218-3185-2	*	
	Ciencias Sociales	Ciencias Sociales 1º ESO (Pr. Anfora) (Libro + monografía)	Oxford	978-84-673-3084-7	*	
	Música	Tempo XXI – Música I	Casals	978-84-218-3717-7	*	
	Destrezas de Lengua	Juglar. Lengua C. y Literatura 1º ESO Libreta de competencias básicas	Vicens Vives	978-84-316-1547-5		
	Ciencias Sociales-Bilingüe	Social Sciences, Core Concepts, ESO 1	Oxford Educación	9788467351972	*	
	Inglés – Bilingüe	Voices 1	Students' Book	Macmillan	9780230533080	*
			Workbook Pack (Castellano)	Macmillan	9780230034037	
	Portugués	Portugués XXI Nova edição (Livro do Aluno + Caderno de Exercícios)	LIDEL	978-972-757-934-1	*	
	Francés	Adosphere 1	Livre de l'élève	SM	978-84-675-4576-0	*
Cahier d'activités			SM	978-84-675-4580-7		

(*) Posiblemente, los libros marcados serán prestados por el centro a todos los alumnos solicitantes de beca de libros. Se recomienda no comprarlos.

Figura 1.1: Hoja informativa. IES Siberia Extremeña

Como puede apreciarse, es un sistema bastante obsoleto, más aún en la sociedad tecnológica que actualmente nos encontramos. Entre los problemas detectados del sistema actual podemos establecer:

- El coste económico y energético que supone trabajar con papel para los centros educativos.
- El tiempo consumido para la creación y manipulación de las hojas informativas, así como para la impresión de las mismas.
- A lo largo de los semestres es frecuente, que los equipos directivos o los propios profesores realicen cambios, por lo tanto, es impensable volver a realizar procesos de impresión.
- Los padres y/o madres quedan desvinculados de los centros educativos al comienzo del curso, con el funcionamiento normal de los mismos, quedando al margen de las nuevas informaciones generadas.
- Los centros educativos disponen de periodos no lectivos, donde las consultas acerca de los materiales didácticos no podrán ser resueltas, provocando posibles errores en la adquisición de los materiales.

1.3. Objetivos de este proyecto fin de grado

El proyecto que se plantea como Trabajo Fin de Grado consiste en terminar con este rudimentario método informativo, para lo cual se desarrollará un sistema de información de materiales curriculares para la Comunidad de Extremadura, basado en software, usando herramientas de código abierto.

Para ello, se realizará el análisis, diseño e implementación de un sistema de información web donde los responsables de los centros podrán generar conocimiento. El resultado de este trabajo serán dos aplicaciones: aplicación de Gestión Educativa y aplicación Móvil Educativa.

Aplicación de Gestión Educativa. Entorno web que supondrá una filosofía de trabajo para profesores y centros educativos que gestiona, actualiza y comunica integralmente las informaciones y datos académicos dentro de una misma plataforma.

Adicionalmente, los **objetivos generales** marcados son:

- Diseñar una aplicación web multiusuario concurrente.
- Implementar un sistema fácilmente ampliable, añadiendo nuevas funcionalidades de forma simple.
- Crear una arquitectura de servidor escalable ,que permita la evolución para desarrollar nuevos servicios, como incorporar otras comunidades al sistema.
- Diseñar una interfaz vía web usable y siguiendo los estándares web.
- Recurrir al uso de una arquitectura Modelo-Vista-Controlador, para proporcionar un núcleo estructural de base de datos centralizado, con la información básica común y reutilizable.

Por tanto, los **objetivos específicos** serán los siguientes:

- Optimización del tiempo empleado en la gestión de inventario de los materiales didácticos.
- Acceso seguro y limitado a la información según el perfil de usuario (profesor/equipo directivo/padres y madres,...)
- Evitar duplicar la adquisición de nuevos materiales.
- Acceso a la ficha de cada material curricular del centro.
- Acceso a los préstamos e intercambios de libros de textos.

Aplicación Móvil Educativa. Otro de los objetivos del proyecto, es la elaboración de un sistema software que pueda ofrecer servicios en el móvil similares a los gestionados por la aplicación web de gestión educativa. Para ello, será necesario la creación de una aplicación móvil con sistema operativo Android y de un servidor nuevo, que dé soporte a la aplicación, ya que el sistema anterior no permite la integración de determinados servicios con estas características.

Por tanto, los objetivos específicos serán los siguientes:

- Crear una aplicación móvil que permita a los usuarios gestionar determinados servicios como, información de centros educativos, listado de materiales curriculares, detalles de profesores, ...

- Crear un servidor que dé servicio a esta aplicación móvil, el cual deberá implementar una interfaz REST.
- Proteger la comunicación entre la aplicación móvil y el servidor, dada la protección legal que requiere dicha información.
- Diseñar un sistema de seguridad en el servidor, que soporte distintos perfiles de usuario, niveles de seguridad y proteja los servicios web en base a dichos perfiles.
- Diseñar una interfaz gráfica en la aplicación móvil adaptada a distintos formatos de pantalla, de fácil lectura y uso.

Capítulo 2

Análisis Previo

Contenidos

2.1. Recursos para favorecer el aprendizaje	7
2.2. Portales de materiales curriculares	9
2.2.1. Agrega	9
2.2.2. Contenidos Educativos Digitales	9
2.2.3. Proyecto Medusa	10
2.2.4. Wikisaber	12
2.2.5. Otros portales	12
2.3. Identificación del entorno tecnológico	13
2.3.1. Herramientas utilizadas	16

En este capítulo explicaré la situación actual de la comunidad educativa, donde se hace necesario utilizar herramientas útiles de contenidos educativos, para que las Tecnologías de la Información y la Comunicación sean fácilmente integrables en el aula. Este capítulo se dividirá en un apartado con diferentes subapartados:

1. Recursos para favorecer el aprendizaje. En este punto se realizará un estudio de las tecnologías educativas implantadas en Extremadura en los últimos diez años.
2. Portales de Recursos digitales. Los recursos didácticos engloban todo el material didáctico al servicio de la enseñanza y son elementos esenciales en el proceso de transmisión de conocimientos del profesor al alumno. El modo de presentar la información es fundamental para su asimilación por el receptor. En este punto citaré una serie de portales web eficaces para dicho propósito.
3. Identificación del entorno tecnológico. En este punto determinaré el entorno tecnológico en que se implementará el sistema, basándome en los diferentes requisitos exigidos.

2.1. Recursos para favorecer el aprendizaje

En la última década, la Consejería de Educación y Cultura del Gobierno de Extremadura ha realizado un importante esfuerzo para conseguir el uso real y cotidiano de las TIC, tanto

en las aulas como en los hogares extrememos. Este proyecto tan innovador comienza en el año 2003, donde se inicia un ambicioso despliegue tecnológico que ha permitido que todos los centros educativos conformen una gran extranet.

En 2005, completada la primera fase de la Red Tecnológica Educativa Extremeña (RTEE) y desarrollado el software básico para su funcionamiento (GnuLinex), se constató la necesidad de extender este modelo de TIC hacia el resto de miembros de la Comunidad Educativa (docentes, alumnos, familias, administración,...) de manera que las innumerables posibilidades de los sistemas tecnológicos actuales se traduzcan en una mayor calidad educativa.

En 2006, con la puesta en marcha de la LOE (Ley Orgánica de Educación), el Ministerio de Educación lanza el programa "Escuela TIC 2.0", cuyo objetivo era poner en marcha las aulas digitales del siglo XXI, aulas dotadas de infraestructura tecnológica y de conectividad.

Este proyecto estaba basado en los siguientes ejes :

- **Aulas digitales.** Dotar de recursos TIC a los centros: ordenadores portátiles para alumnado y profesorado y aulas digitales con dotación eficaz estandarizada.
- **Garantizar la conectividad a Internet** y la interconectividad dentro del aula para todos los equipos.
- **Promover la formación del profesorado.**
- **Generar y facilitar el acceso a materiales digitales educativos** ajustados a los diseños curriculares, tanto para el profesorado como para el alumnado y sus familias.
- **Implicar a alumno/as y a las familias** en la adquisición, custodia y uso de estos recursos.

Por otra parte, en 2006, la Consejería de Educación se planteó dotar a la Comunidad Educativa Extremeña de distintas plataformas integradas que, a través de Internet, permitieran establecer un entorno colaborativo para los diferentes miembros de la comunidad educativa, acercando el centro a la familia y potenciando la participación de todos en el desarrollo y progreso de la Educación. Podemos destacar el entorno Rayuela.

Paralelamente, durante estos años, estamos asistiendo a un aumento muy importante en el número y calidad de los materiales curriculares , libres o de pago, producidos por la administración educativa, por las editoriales o por autores independientes (profesorado, alumnado, ...). Los recursos didácticos han pasado a ser parte muy importantes en el proceso de enseñanza-aprendizaje desarrollado en nuestras aulas.

Llegados a este punto, se hace necesario dar un paso más en la formación con recursos digitales, y poner a disposición de la Comunidad Educativa, una herramienta mucho más potente, funcional y que suponga una revolución en el proceso de enseñanza-aprendizaje utilizando medios digitales, y, en especial, los libros de texto digitales, y el posterior seguimiento de la relación que dichos contenidos establecerán con los alumnos y docentes. Además, se busca implicar de un modo masivo al profesorado y alumnado de todos los niveles, por lo que se hace necesaria una herramienta con un curva de aprendizaje mínima, que utilice las últimas tecnologías webs y que permita la adaptación al alumnado infantil además de a las familias con pocos conocimientos en la utilización de este tipo de herramientas.

2.2. Portales de materiales curriculares

En esta sección se presentarán los sistemas de gestión educativa identificados durante la investigación acompañados por sus principales características.

2.2.1. Agrega

El proyecto Agrega 2 es una plataforma tecnológica implantada en las administraciones educativas españolas, a través de nodos autonómicos interconectados entre sí (federación de repositorios de objetos digitales educativos Agrega). Esta red de servidores aloja miles de recursos educativos, adecuadamente etiquetados y empaquetados.

Los objetos educativos y secuencias didácticas alojados en Agrega se pueden descargar y ser utilizados directamente por profesores y alumnos como recursos didácticos y también ser combinados para confeccionar con ellos nuevos materiales educativos [31]

El sistema presenta las siguientes funcionalidades :

- Los recursos didácticos pueden ser utilizados directamente en el aula sin conexión a Internet.
- Los recursos didácticos admiten múltiples formatos de exportación (LIGHT, SCORM, ...)
- Permite adaptar los contenidos a nuestros intereses así como crear nuevos recursos.
- Posibilidad de crear itinerarios de aprendizaje donde aparecerán todas los recursos asociados y los usuarios que forman parte de él.
- Multiplataforma.

En la figura 2.1 se puede observar el portal del proyecto.

2.2.2. Contenidos Educativos Digitales

El Gobierno de Extremadura comparte con la comunidad educativa un portal denominado *Contenidos Educativos Digitales* muy atractivos para su uso en clase, especialmente en PDI. Abarcan todas las etapas, todos los niveles y todas las materias. Los contenidos se pueden visualizar online, pero también permite su descarga.

El sistema presenta las siguientes funcionalidades [24]:

- Los recursos didácticos pueden ser utilizados sin requerimiento de conexión a Internet.
- Los recursos pueden ser manipulados por el alumnado, y experimentar con conceptos y procedimientos para mejorar su comprensión y adquisición.
- Incorpora un sistema de ayuda fácil e intuitivo sobre la utilización de los recursos.
- Dispone de un sistema de calificación del recurso, de modo que la comunidad educativa puede evaluar su uso dentro del aula.
- Gran variedad de recursos en los diferentes niveles educativos.

En la figura 2.2 se puede observar el portal del proyecto.

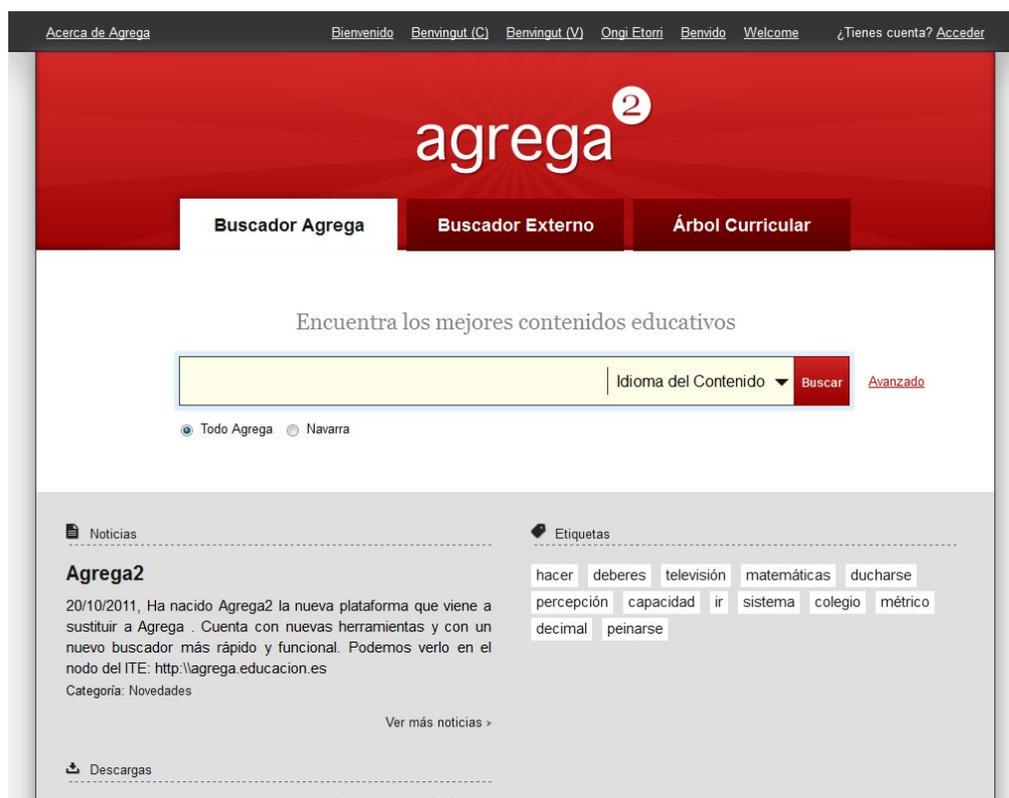


Figura 2.1: Proyecto Agregate

2.2.3. Proyecto Medusa

El proyecto Medusa es un portal educativo desarrollado por el Gobierno de Canarias como un servicio más para la comunidad educativa, cuyo objetivo es la implantación de las TIC, para el logro de la competencia digital y su uso como herramienta de trabajo intelectual, objetivos y fines de aprendizaje [21] .

El profesorado encontrará recursos y contenidos educativos para el aula, así como otros espacios para compartir proyectos y experiencias educativas realizadas de forma individual o colectiva.

El alumnado tendrá un espacio en el que encontrará enlaces de interés y recursos, con fines de aprendizaje, trabajo y ocio.

El sistema presenta las siguientes funcionalidades:

- Los recursos didácticos pueden ser utilizados sin requerimiento de conexión a Internet.
- Gran soporte técnico para el desarrollo de recursos.
- Gran variedad de recursos en los diferentes niveles educativos.
- Multiplataforma.

En la figura 2.3 se puede observar el portal del proyecto.



Figura 2.2: Contenidos Educativos Digitales



Figura 2.3: Proyecto Medusa
Grado en Ingeniería Informática en Tecnologías de la Información

2.2.4. Wikisaber

Wikisaber es un portal educativo fundamentado en la enseñanza y el aprendizaje colaborativo. Gratuito y pensado para todos los cursos de enseñanza obligatoria. Wikisaber quiere ser el marco para que padres y tutores, profesores y alumnos puedan compartir conocimiento y se comuniquen entre sí, además de aportarles recursos y contenidos de calidad .

El sistema presenta las siguientes funcionalidades [17]:

- Gran cantidad de materiales, herramientas y contenidos digitales a disposición del alumnado, profesorado y familias.
- Solo es necesario rellenar un simple formulario de registro para poder utilizar los recursos bajo licencia creative commons.
- Permite crear espacios para compartir y colaborar.
- Dispone de servicio técnico para facilitar el uso de los recursos didácticos y herramientas educativas.

En la figura 2.4 se puede observar el portal del proyecto.



Figura 2.4: Wikisaber.es

2.2.5. Otros portales

- EDUC@CONTIC <http://www.educacontic.es/>

Portal web educativo impulsado por el Plan Avanza y Red.es del Gobierno de España, destinado a docentes interesados en las aplicaciones de las TIC en las escuelas. Tiene numerosos recursos, blogs, noticias, e informaciones sobre software educativo.

- INSTITUTO DE TECNOLOGÍAS EDUCATIVAS <http://www.ite.educacion.es/>

Este es el sitio web del Ministerio de Educación destinado a potenciar el uso de los recursos digitales y la formación del profesorado en esta temática. Es quizás, el portal educativo en lengua española, más completo donde se pueden encontrar de forma gratuita numerosos materiales multimedia didácticos clasificados bien por niveles educativos, bien por áreas temáticas, así como información diversa sobre las TIC, su uso educativo y la formación del profesorado en este campo.
- MAESTROTECA <http://www.maestroteca.com>

Uno de los portales pioneros en español sobre webs educativos, creado en 1998. Es un directorio con muchos enlaces (casi 3.000 en lengua española) clasificados temáticamente en 56 categorías de diverso tipo: blogs educativos, redes sociales, aplicaciones didácticas, periódicos y prensa escolar, editoriales educativas, enciclopedias, etc. Tiene un buscador propio.
- PROFES.NET <http://www.profes.net/>

Es el portal web de una editorial privada de libros de texto, SM, pero está abierta a cualquier docente. Tiene numerosas secciones de acceso libre: Comunidades Virtuales para Educación Infantil, Primaria y Secundaria; banco de recursos; Revista Digital; Biblioteca, etc.
- AULABLOG <http://www.aulablog.com/>

Es un blog de edublogs que publica los post o noticias que se publican en los blogs de otros docentes adscriptos al mismo. Además permite acceder a esos blogs educativos, crear nuevos blogs o suscribirse a canales RSS de noticias educativas. Tiene secciones como una biblioteca de blogs, una wiki sobre los mismos, tutoriales para crear blogs, etc.
- BUSCADOR INFANTIL <http://www.buscadorinfantil.com/>

Es el primer buscador específicamente creado en lengua española para niños. Se basa en la tecnología de búsquedas de Google. Es muy sencillo de usar y ofrece una selección de enlaces filtrados y adaptados para el público infantil y escolar.
- EDUCARED <http://www.educared.net/>

Es un portal para docentes creado por la Fundación Telefónica en colaboración con otras asociaciones e instituciones de diverso tipo (sindicatos, asociaciones de profesores, ...). Tiene diversas secciones y recursos. Ofrece noticias, formación, acceso a una sección de profesores innovadores, comunicación con escuelas latinoamericanas, materiales educativos, foros de debate, ...

2.3. Identificación del entorno tecnológico

Las aplicaciones Web (web-based application) son aplicaciones distribuidas que usan las tecnologías Web como su infraestructura. Usan un navegador Web como cliente, el protocolo HTTP (Hypertext Transfer Protocol) para comunicarse entre cliente y servidor, y diversos lenguajes Web para expresar el contenido transmitido entre clientes y servidores.

Hoy en día, existen muy diversas tecnologías para desarrollar aplicaciones Web. El modo de generar páginas dinámicas ha evolucionado, desde la utilización del Common Gateway Interface (CGI), hasta los servlets, pasando por tecnologías tipo JavaServer Pages (JSP). Todas estas tecnologías se encuadran dentro de aquellas conocidas como Server Side, ya que se ejecutan en el servidor web.

Por tal motivo, las aplicaciones Web son sistemas complejos que utilizan una combinación de plataformas tecnológicas para su funcionamiento y se caracterizan por tener un navegador Web como interfaz.

A nivel arquitectural, las aplicaciones Web se componen de tres capas: la capa de presentación, la de lógica de negocio y la capa de persistencia.

- La *capa de presentación* se encarga de mostrar todos los componentes de la interfaz que permite a los usuarios interactuar con la aplicación Web. Renderizan y formatean la información que se quiere presentar, así como capturan y validan las entradas de datos.
- La *capa de lógica de negocio* contiene la lógica principal de procesamiento de datos dentro de nuestra aplicación Web. Se comunica con la capa de presentación para obtener las entradas del usuario y presentar la información resultante, así como la capa de acceso a datos o directamente con servicios para realizar sus operaciones.
- La *capa de persistencia* ofrece los servicios de persistencia y recuperación de información a las capas superiores.

En la figura 2.5 se muestra la descomposición de las capas de una aplicación Web.

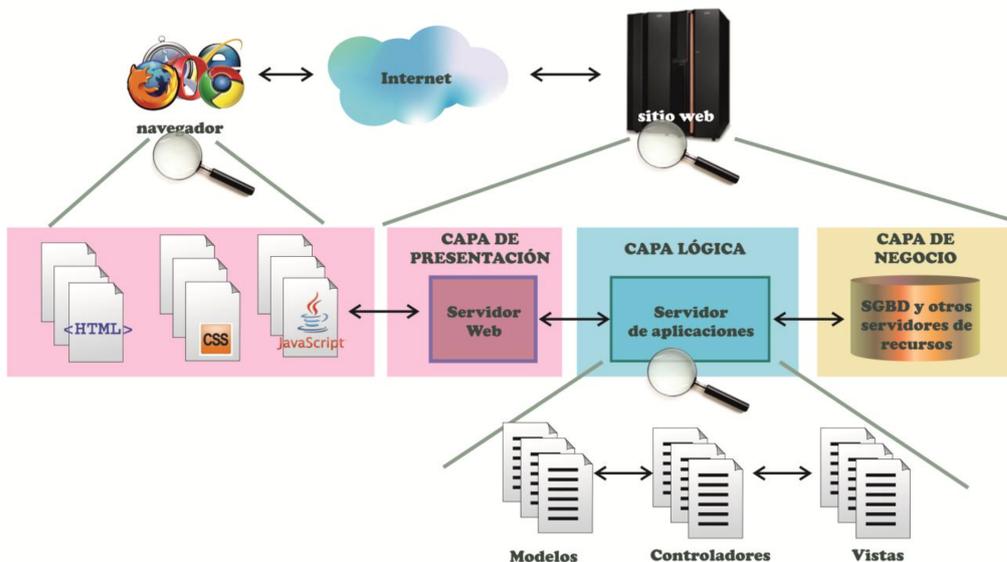


Figura 2.5: Arquitectura de un sistema de aplicaciones web

Dentro de la vista lógica, los patrones de diseño constituyen la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y también para otros ámbitos referentes al diseño de interfaces. Uno de los patrones de arquitectura que se utilizan para el desarrollo de aplicaciones Web es el *Modelo-Vista-Controlador* (MVC). MVC es un patrón que fue utilizado para construir las interfaces de usuario en el lenguaje Smalltalk-80, siendo la aportación más importante de este patrón la separación de los componentes relacionados con los datos de la aplicación de los componentes de la interfaz de usuario.

La separación de las capas permite tener, a nivel de desarrollo, un código más claro, flexible y reusable.

El patrón MVC descompone la aplicación en capas, permitiendo tener una separación entre la lógica de negocio de la aplicación, la representación y la persistencia. El patrón MVC identifica tres capas que son importantes para cualquier aplicación, las cuales son:

- Modelo: encapsula los datos de la aplicación y la lógica para interactuar con ellos.
- Vista: maneja la interacción con el usuario y la representación del modelo.
- Controlador: es el intermediario entre el modelo y la vista ante las peticiones generadas por el cliente en la vista. El controlador se encarga de seleccionar el modelo solicitado por el usuario y la vista adecuada para representarlo.

En la figura 2.6 se puede apreciar una arquitectura simple para una aplicación Web utilizando el patrón MVC como pilar para la definición de la arquitectura.

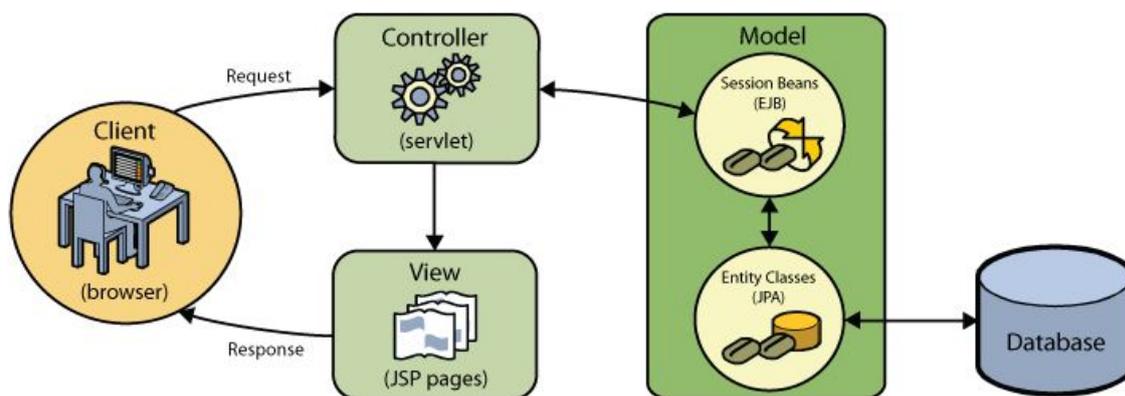


Figura 2.6: Diagrama Modelo-Vista-Controlador.[11]

2.3.1. Herramientas utilizadas

■ HTML Y JAVASCRIPT

HTML es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<, >). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

El diseño en HTML, aparte de cumplir con las especificaciones propias del lenguaje debe respetar unos criterios de accesibilidad web, siguiendo unas pautas, o las normativas y leyes vigentes en los países donde se regule dicho concepto. Se encuentra disponible y desarrollado por el W3C a través de las Pautas de Accesibilidad al Contenido Web 1.0 WCAG. [30]

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de herencia, si bien ésta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM. [9]

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

■ CSS (Cascading Style Sheets)

Las hojas de estilo aparecieron poco después que el lenguaje SGML (Standard Generalize Markup Language) alrededor de la década de los setenta. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El organismo W3C (World Wide Web Consortium), encargado de crear todo los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilo específico para el lenguaje HTML (HyperText Markup Language). Las dos propuestas que aparecieron sobre los años 90 fueron: CHSS (Cascading HTML Style Sheets) y SSP (Stream-based Style Sheet Proposal). [7]

La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets). [7]

CSS es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. El World Wide Web Consortium, es el encargado

de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información del estilo (color, fuente, alineación, tamaño, ...) puede ser definida en un documento separado o en el mismo documento HTML.

El W3C trabaja continuamente para mejorar el lenguaje CSS, corrigiendo errores e incorporando nuevas funcionalidades. Antes de llegar a una especificación se trabajan numerosos borradores que son sometidos a revisión y corrección. Cuando se alcanza un relativo grado de acuerdo entre los miembros del W3C se libera lo que se denomina una recomendación oficial de CSS ó versión a modo de propuesta para su uso y aplicación por todas las empresas, instituciones y personas. [7]

Las versiones de CSS a lo largo de la historia han sido: CSS1(publicada en 1996) , CSS2 (publicada en en 1998), CSS2.1 (publicada en 2004), CSS3(publicada en 2011) y el futuro CSS4 (se estima que puede ser especificación oficial en 2019).

■ JQUERY

jQuery es una librería de Javascript que ha irrumpido con fuerza desde el año 2006. Su autor original fue John Resig, aunque como sucede con todas las librerías exitosas, actualmente recibe contribuciones de decenas de programadores. Este framework Javascript permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones,...

jQuery es un software libre y de código abierto, que ha sido programado de forma muy eficiente y su versión comprimida apenas ocupa 20KB, por lo que no retrasará mucho la carga de las páginas. Además, el servidor lo enviará al cliente la primera vez que visite una página del sitio. En siguientes páginas, el cliente ya tendrá el archivo del framework, por lo que no necesitará transferirlo y lo tomará de la caché. Con lo que la carga de la página sólo se verá afectada por el peso de este framework una vez por usuario.

■ BOOTSTRAP

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y Javascript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Esta técnica de diseño y desarrollo se conoce como diseño adaptativo (responsive design).

El framework trae varios elementos con estilos predefinidos fáciles de configurar: botones, menús desplegados, formularios, etiquetas,...y otra serie de elementos que permiten ser fácilmente integrables con JQuery para ofrecer ventanas y tooltips dinámicos. Es compatible con la mayoría de los navegadores web, e incluye soporte para HTML5 y CSS3.

■ JSP (Java Server Pages)

JSP es un lenguaje desarrollado por Sun Microsystems para la creación de sitios web dinámicos, que presenta funcionalidades muy similares a otras tecnologías web como ASP.NET. Es pues, una tecnología orientada a desarrollar páginas web en Java. Creado para ejecutarse del lado del servidor. [32]

La tecnología JSP es multiplataforma permitiendo crear aplicaciones web que se ejecuten en varios servidores web, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuesta de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java.

El motor de las páginas JSP está basado en los servlets de Java (programas en Java destinados a ejecutarse en el servidor), aunque su curva de aprendizaje es menor. Generamos archivos con extensión *.jsp* que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor. Antes de que sean funcionales los archivos, el motor JSP lleva a cabo una fase de traducción de esa página en un servlet, implementado en un archivo class (Byte codes de Java). Esta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de la página *.jsp*, aunque existe la opción de precompilar en código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

■ **J2EE (Java Platform Enterprise Edition)**

Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura distribuida de n niveles, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación.

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc. y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets (siguiendo la especificación de Portlets Java), JSPs y varias tecnologías de servicios web. Esto permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, la escalabilidad, la concurrencia y la gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel. [19]

La especificación original J2EE fue desarrollada por Sun Microsystems.

Las APIs de Java EE incluyen varias tecnologías que extienden la funcionalidad de las APIs base de Java SE.

javax.ejb

La API Enterprise JavaBeans define un conjunto de APIs que un contenedor de objetos distribuidos soportará para suministrar persistencia, RPCs (usando RMI o RMI-IIOP), control de concurrencia, transacciones y control de acceso para objetos distribuidos. [19]

javax.naming

javax.naming.spi definen la API de JNDI. La Interfaz de Nombrado y Directorio Java es una Interfaz de Programación de Aplicaciones para servicios de directorio. Esto permite a los clientes descubrir y buscar objetos y nombres a través de un nombre. Adicionalmente, especifica un SPI que permite que las implementaciones del servicio de directorio sean enchufadas en el framework. Las implementaciones pueden hacer uso de un servidor, un fichero, o una base de datos. [19]

java.sql

Los paquetes `java.sql` y `javax.sql` definen la API de JDBC que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema de operación donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice. [19]

java.transaction

Estos paquetes definen la API JTA. Este API para transacciones en Java establece una serie de Interfaces java entre el manejador de transacciones y las partes involucradas en el sistema de transacciones distribuidas: el servidor de aplicaciones, el manejador de recursos y las aplicaciones transaccionales. [19]

javax.xml

Estos paquetes definen la API JAXP que sirve para la manipulación y el tratamiento de archivos XML. [19]

javax.jms

Estos paquetes definen la API JMS. La API de Servicios de Mensajería de Java es la solución para el uso de colas de mensajes. Éste es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma de Java 2 crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona. [19]

■ **TOMCAT**

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JSP de Sun Microsystems.

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

■ **MYSQL**

MySQL es un sistema de gestión de base de datos relacional, multi-hilo y multiusuario. MySQL AB (subsidiaria de Sun Microsystems) desarrolla MySQL como software libre en un esquema de licencia dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privados pueden comprar a la empresa una licencia específica que les permita este uso. [12]

MySQL es muy utilizado en aplicaciones web. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

■ SDK ANDROID

SDK responde a las siglas de *Software Development Kit*, lo que viene a ser un kit de desarrollo de software, que permitirá desarrollar aplicaciones basadas en el lenguaje Java.

El SDK de Android, incluye un conjunto de herramientas de desarrollo, que comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU (es un emulador de procesadores basado en la traducción dinámica de binarios (conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped), documentación, ejemplos de código y tutoriales. [18]

Las plataformas de desarrollo soportadas incluyen cualquier distribución de Linux, Mac OS X y Windows. También puede utilizarse el propio sistema Android para desarrollos, utilizando las aplicaciones AIDE - IDE for Android Java C++ y el editor de java.

Las plataformas integrales de desarrollo (IDE - Integrated Development Environment) soportadas oficialmente son:

- *Eclipse*. Es una plataforma de integración de herramientas de desarrollo diseñada para ser extendida de forma indefinida a través de plugins (complemento, extensión o programa adicional). Inicialmente desarrollado por IBM, y actualmente esta bajo el desarrollo de la Fundación Eclipse.

Para su funcionamiento con Android debe integrarse con el complemento ADT (Android Development Tools plugin).

- *IntelliJ IDEA*. Se autodefine como un entorno inteligente para desarrollar aplicaciones java, cliente y servidor del que existen dos distribuciones: Community Edition (open source) y Ultimate (comercial). Posee además, un avanzado editor de código, compatible con multitud de tecnologías (AJAX, JSP, EJB, ..). Soporta completamente el desarrollo en Android utilizando plugin J2ME.

- *Android Studio*. Es un entorno de desarrollo integrado (IDE), basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0. Está programado en Java y es multiplataforma.

Fue presentado por Google en mayo del 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción.

■ SERVICIO WEB

El consorcio W3C (World Web Consortium) define los Servicios Web como sistemas software identificados por una URI (Uniform Resource Identifier), cuyas interfaces se pueden definir y descubrir mediante documentos XML. Los servicios web hacen posible la interacción entre agentes software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet. [14]

La definición de Servicios Web propuesta alberga muchos tipos diferentes de sistemas, pero el caso común de uso se refiere a clientes y servidores que se comunican mediante mensajes XML que siguen el estándar SOAP (Simple Object Access Protocol).

En los últimos años se ha popularizado un estilo de arquitectura Software conocido como **REST** (Representational State Transfer). Es una técnica de arquitectura software compuesta por un conjunto coordinado de restricciones arquitectónicas aplicadas a componentes, conectores y elementos de datos.

La idea tras la que subyace REST es la de que, en lugar de utilizar mecanismos complejos como CORBA, RPC o SOAP para realizar la comunicación entre máquinas, simplemente utilizar el protocolo HTTP para realizar las llamadas entre ellas.

Para ello, REST se basa en un protocolo con las siguientes características: [27]

- *Arquitectura cliente-servidor*: tras este principio se encuentra la separación de contenidos; una separación de funcionalidades apropiada simplificará el sistema servidor y mejorará la escalabilidad del sistema.
- *Sin estado*: cada petición del cliente al servidor deberá contener toda la información necesaria para que este último pueda comprender la solicitud, imposibilitando el almacenamiento de cualquier tipo de contexto en el servidor. De este modo, será el dispositivo cliente el encargado de mantener el estado de la sesión.
- *Caché*: con el fin de optimizar en lo posible la eficiencia del sistema es posible añadir restricciones de caché en las que se especifique si una determinada respuesta es o no cacheable y, por tanto, reutilizable en solicitudes similares.
- *Sistema por capas*: un sistema por capas impide que los distintos elementos que lo componen vean más allá de la capa inmediata con la que interactúan, minimizando de este modo la complejidad global del sistema y mejorando su escalabilidad. Sin embargo, un sistema basado en capas aumentará la carga y latencia del sistema al realizar el procesamiento de los datos, reduciendo el rendimiento del sistema.
- *Code on demand (opcional)*: REST permite definir la funcionalidad de los dispositivos cliente y hacer que el funcionamiento de estos se base únicamente en pequeños applets o scripts, reduciendo de este modo su complejidad.

Capítulo 3

Metodología de Desarrollo

Contenidos

3.1. Introducción	23
3.1.1. Análisis	23
3.1.2. Diagrama de caso de uso	24
3.1.3. Diagramas de secuencia	25
3.1.4. Diseño de la interfaz de usuario	25

En este capítulo describiremos la metodología utilizada en el desarrollo de nuestro proyecto. Para la estructuración, planificación y control del proyecto se aplicará un proceso de desarrollo software iterativo e incremental que permita controlar y gestionar los riesgos de forma periódica.

En el siguiente esquema podremos ver cómo estará organizada y tras ello describiremos en qué consiste y cómo se van a realizar cada una de las fases que la forman.

3.1. Introducción

En este apartado explicaremos la metodología que se utilizará en el desarrollo de nuestro proyecto. Se realizará con una metodología orientada a objetos [29]. A continuación mostraremos un esquema de como estará organizada y tras ello describiremos en que consiste y cómo se va a realizar cada una de las fases. Fig:3.1

3.1.1. Análisis

El objetivo de este proceso es analizar el problema, llegando hasta una especificación completa y minuciosa del comportamiento del mismo y consistencia, en la que se especifican las características funcionales y operacionales del mismo.

Este proceso debe dar información de las necesidades del sistema, cuáles van a ser las funciones del sistema (no como se va a implementar), debe mostrar una visión completa del sistema y de lo que este debe hacer.

El fin del análisis orientado a objetos es desarrollar una serie de modelos que describan el software para satisfacer un conjunto de requisitos definidos por el usuario.

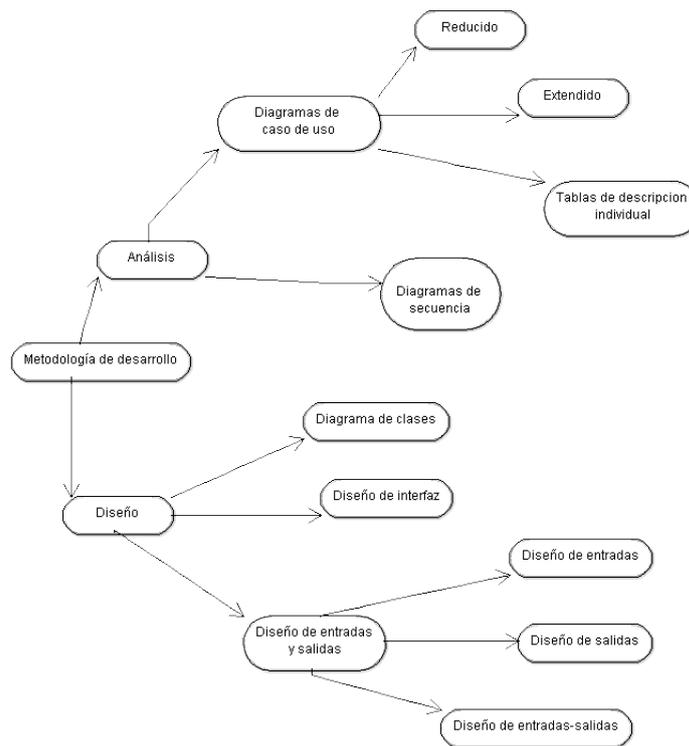


Figura 3.1: Esquema de la metodología

3.1.2. Diagrama de caso de uso

La función de un diagrama de caso de uso es documentar el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede realizar. [29]

Su principal ventaja es la facilidad que tienen para interpretarlo, hacen que la comunicación con el cliente sea mas fácil.

En este apartado mostraremos los casos de uso del sistema de forma reducida, después de forma extendida, y tras ello los describiremos individualmente mediante tablas.

Elementos básicos del caso de uso de forma reducida

- Actores. Representan un tipo de usuario del sistema. Un usuario es cualquier cosa externa que interactúa con el sistema. Se representa con una figura de un muñeco.
- Caso de uso. Son las acciones o tareas que se realizar tras una orden de uno o varios actores en el sistema con un objetivo determinado.
- Asociaciones. Son las relaciones entre los actores y los casos de uso cuando el usuario interactúa con el sistema.

Formato de la tabla que describe los casos de uso específicos.

Código del caso de uso	Código al que corresponde el caso de uso de la lista
Descripción	Nombre del caso de uso
Actores	Actores que realizan la acción
Precondiciones	Condiciones que deben cumplirse para poder realizar la acción.
Flujo normal	Etapas que se realizan para el desarrollo de la función.
Flujo alternativo	Camino alternativo para realizar la función
Postcondiciones	Resultado que debe cumplirse tras la ejecución de la función.

3.1.3. Diagramas de secuencia

Muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo. En particular, muestra los objetos participantes en la interacción y los mensajes (llamadas a métodos) que intercambian, ordenados según su secuencia en el tiempo. Se realizan diagramas de secuencia para definir acciones que se pueden realizar en la aplicación en cuestión.

En este apartado encontraremos los diagramas de secuencia del sistema, aquellos que sean similares se describirán con un sólo diagrama y se explicarán qué mensajes intercambian los demás diagramas agrupados en él.

3.1.4. Diseño de la interfaz de usuario

La interfaz de usuario es lo que une al sistema informático con el usuario. Define como interactúan usuario y ordenador, juega un papel importante en la aceptación del sistema.

En esta sección se describirán los distintos diseños de cada pantalla que forman la interfaz de usuario. Estos diseños deberán seguir un formato que las haga: sencillas, accesibles, atractivas y cómodas para el usuario.

3.1.4.1. Diseño de entradas y salidas

En este apartado se describen las entradas, salidas y entradas-salidas al sistema.

3.1.4.2. Diseño de entradas

En este apartado se describen las distintas pantallas que el usuario se encontrará cuando va a introducir datos en el sistema. Deben ser claras, sencillas y fáciles de entender.

3.1.4.3. Diseño de salidas

En esta sección se describirán las distintas pantallas que visualizará el usuario cuando se realizan confirmaciones.

3.1.4.4. Diseño de entradas salidas

Algunas partes de la aplicación contienen pantallas que aportan datos del sistema al usuario y al mismo tiempo permiten introducir datos nuevos al mismo.

Capítulo 4

Análisis del Sistema

Contenidos

4.1. Planificación del sistema	28
4.1.1. Viabilidad técnica	28
4.1.2. Viabilidad operacional	28
4.1.3. Viabilidad económica	29
4.1.4. Viabilidad de la solución	29
4.2. Análisis funcional	30
4.2.1. Casos de uso global	30
4.2.2. Listado de casos de usos	35
4.3. Diagramas de casos de uso detallados	36
4.3.1. Casos de uso detallados del actor Administrador	36
4.3.2. Casos de uso detallados del actor Equipo directivo	42
4.3.3. Casos de uso detallados del actor Profesor	48
4.3.4. Casos de uso detallados del actor Alumno	51
4.3.5. Casos de uso detallados del actor Padre/Madre	51
4.3.6. Casos de uso detallados del actor Móvil	53
4.4. Diagramas de secuencia	53
4.4.1. Diagrama de secuencia de la aplicación web	54
4.4.2. Diagrama de secuencia de la aplicación móvil	54

Este capítulo describe el análisis del sistema propuesto, llegando hasta una especificación completa y minuciosa del comportamiento del mismo, consistencia y viabilidad, en la que se especifican las características funcionales y operacionales del mismo.

En la sección 4.1 se realiza la planificación del sistema.

En la sección 4.2 se presenta el análisis funcional del sistema, analizando pormenorizada-mente cada de las operaciones deseables.

En la sección 4.3 se detallan los diagramas de caso de usos.

En la sección 4.4 se muestran los diagramas del secuencia generados.

4.1. Planificación del sistema

El desarrollo de cualquier sistema software no consiste únicamente en la codificación del programa, aunque sea una de las fases más importante de cualquier ciclo de vida de un proyecto software, sino que existen determinadas tareas que deben ser desarrolladas paralelamente si se desea un sistema robusto y de fácil mantenimiento y con garantías de finalización en el tiempo y coste estimados. Para determinar las tareas que deben ser llevadas a cabo es necesario seguir un proceso de desarrollo software, adaptándolo según se estime conveniente al proyecto en el cual se aplique.

En el punto tres del capítulo uno, de la introducción a este Trabajo de Fin de Grado, se mostró de forma generalizada las funciones generales que debería aportar el sistema. Además en el capítulo dos, se analizaron distintas tecnologías que cumplían los objetivos planteados.

Llegado a este punto es necesario analizar la viabilidad del sistema propuesto así como las operaciones que se desean desarrollar, estableciendo los actores que realizarán las operaciones, los parámetros de entrada y salida, los posibles errores a controlar, etc.

Tal y como se expresó se desarrollarán dos sistemas:

- **Aplicación de Gestión Educativa.** Entorno web que supondrá una filosofía de trabajo para profesores y centros educativos que gestiona, actualiza y comunica integralmente las informaciones y datos académicos dentro una misma plataforma.
- **Aplicación Móvil Educativa.** Entorno móvil desarrollado para ofrecer determinados servicios gestionados por la aplicación web de gestión educativa.

El objetivo del estudio de viabilidad es definir, de forma clara y no ambigua, las características y cualidades del sistema software a desarrollar, siendo el punto de partida analizar las necesidades para proponer una solución a corto plazo, que tenga en cuenta las restricciones económicas y operativas. Por ello, es necesario identificar los requisitos que se han de satisfacer y se estudiará, si procede, el marco de trabajo actual.

4.1.1. Viabilidad técnica

La viabilidad técnica determina si la tecnología existente en el mercado, permite hacer realidad el proyecto y también si es conveniente hacerlo. Debe proveer información sobre las diversas formas de materializar el proyecto o los diferentes procesos que pueden utilizarse para producir un bien o servicio. El estudio deberá contener una estimación de los requerimientos de capital, mano de obra y recursos materiales, tanto para la puesta en marcha, como para el estado de operación del proyecto. [28]

En mi proyecto, estudiaremos la viabilidad técnica para el control de materiales curriculares en los centros educativos extremeños. Dicho sistema estará gestionado por un servidor que gestionará y controlará todo lo relacionado con los elementos didácticos, gestión de centros educativos, profesorado y alumnado. Esta gestión se realizará a través de un entorno web y móvil, por lo que se determina que este requerimiento es viable, pues se trata de tecnología utilizada durante años.

4.1.2. Viabilidad operacional

La viabilidad operacional es el proceso de examinar la concordancia entre los resultados del proyecto y los objetivos marcados. Es decir, la viabilidad operacional estudia si las necesidades

de los usuarios finales han sido satisfechas con el nuevo sistema de información. [23]

Esta viabilidad puede desglosarse en dos partes:

1. El estudio sobre si merece la pena resolver un problema o si funcionará la solución propuesta.
2. La opinión de los usuarios sobre el problema y sobre la solución propuesta.

Se ha visto mediante investigaciones realizadas, que las dificultades que se van a remediar mediante este sistema de gestión web son muy ventajosas, debido a que los procesos son realizados diariamente y de manera no equitativa. Por este motivo, la creación de un sistema que automatice dicho problema será una de las mejores soluciones, con lo que se creará un sistema que cumpla todas las expectativas y necesidades.

En cuanto a la información, control y eficacia se tiene como objetivo trabajar paralelamente con los responsables del centro educativo, profesores y padres/madres, para que ellos vayan describiendo cuales serán los requerimientos del sistema y de esta manera desarrollar un sistema donde se tendrá la completa satisfacción de los que lo van a utilizar.

Cuando el sistema esté finalizado, solo será necesario que los futuros usuarios tengan acceso a internet desde cualquier dispositivo (ordenador, portátil, tablet, móvil,...) para tener acceso tanto a la aplicación web como a la aplicación móvil.

4.1.3. Viabilidad económica

El estudio de la viabilidad económica determina el presupuesto de coste de los recursos técnicos, humanos y materiales, tanto para desarrollo como para la implantación del sistema [23]. Además, nos ayudará a realizar el análisis costo beneficio de nuestro sistema, el mismo que nos permitirá determinar si es factible desarrollar económicamente el proyecto.

4.1.4. Viabilidad de la solución

El resultado de los puntos anteriores que conforman este proyecto de viabilidad suelen ser un grupo de problemas junto con sus soluciones, candidatos a convertirse en proyectos reales. Para antes de que el proyecto sea marcado como futuro posible, le ha face falta también un estudio de la viabilidad de las soluciones propuestas, acompañado de una primera estimación de objetivos, costes y resultados esperados.

Se observa con las informaciones aportadas que técnicamente, operacionalmente y económicamente el proyecto es catalogado como viable. Los requisitos técnicos son asequibles con las tecnologías actuales y el coste de desarrollo no es elevado con respecto a los beneficios que se pretenden obtener.

PRESUPUESTO				
	Concepto	Horas	€/Hora	Total(€)
Proyecto	Planificación y análisis	150	40	6000
	Seguimiento y control	860	40	34400
	Cierre	50	40	2000
	Concepto	Horas	€/Hora	Total(€)
Sitio Web	Análisis	90	30	2700
	Diseño	1200	20	24000
	Implementación	48	20	900
	Concepto	Total(€)		
	Software	0		
	Hardware	0		
	Dominio	90/año		
	Concepto	Horas	€/Hora	Total(€)
Sitio Móvil	Análisis	100	25	2500
	Diseño	600	25	15000
	Implementación	30	25	750
	Concepto	Total(€)		
	Software	0		
	Hardware	0		
Total				88250

Tabla 4.1: Análisis del coste del desarrollo

4.2. Análisis funcional

En esta sección se realiza un estudio de las funcionalidades del sistema, utilizando las distintas metodologías expuestas en los puntos del capítulo anterior. Al partir de las informaciones aportadas, se observa la necesidad de cinco actores:

- **Administrador**. Encargado de la gestión y control del sistema: materiales, usuarios del equipo directivo, cursos, asignaturas.
- **Equipo directivo**. Les permitirá realizar operaciones de mantenimiento sobre profesores, materiales, cursos, asignaturas.
- **Profesor**. Será el encargado de asignar materiales a sus cursos así como a sus asignaturas.
- **Padres/Madres**. Les permitirá realizar distintas operaciones de consultas sobre profesores, alumnos y materiales.
- **Alumno, Móvil**. Les permitirá realizar operaciones de consultas sobre materiales por profesor, curso o asignaturas.

4.2.1. Casos de uso global

En la figura 4.1 se muestran los casos de usos del actor Administrador del sistema. A groso modo, el Administrador podrá realizar las siguientes funcionalidades:

- Operaciones de añadir, eliminar, modificar o listar usuarios del sistema.
- Operaciones de añadir, eliminar, modificar o listar centros educativos.
- Operaciones de añadir, eliminar, modificar o listar materiales curriculares.
- Operaciones de consultas sobre materiales curriculares por tipo o centro.
- Inicio y fin de sesión.

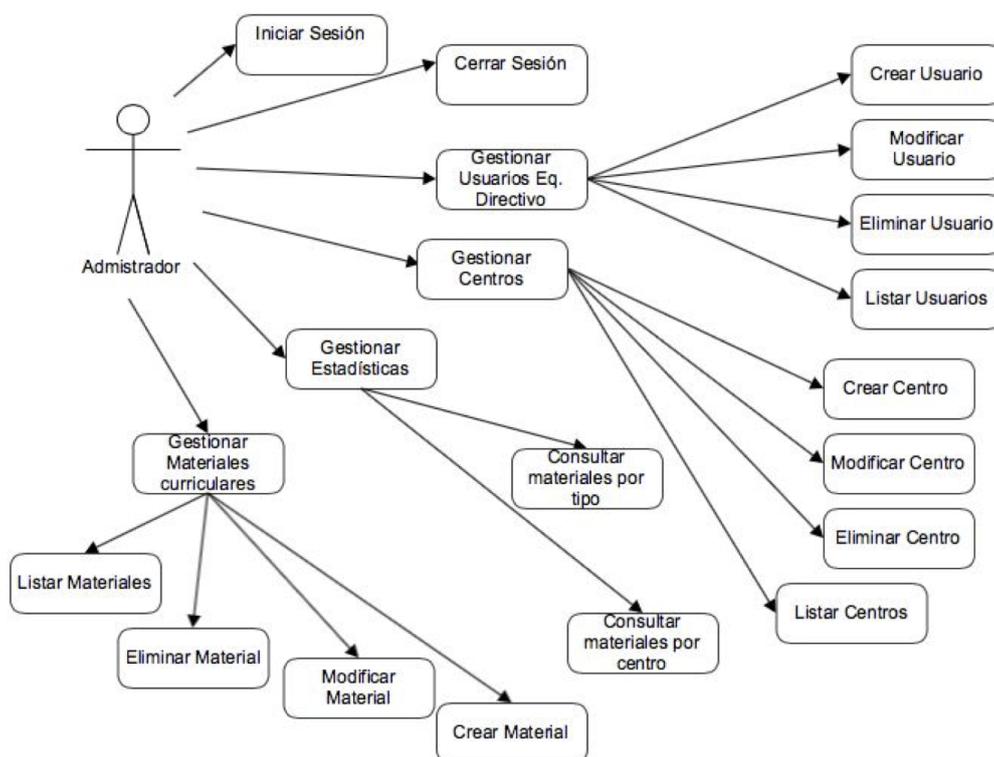


Figura 4.1: Caso de uso del actor administrador

En la figura 4.2 se exponen las acciones que podrá realizar un usuario del equipo directivo. Concretamente las operaciones son:

- Operaciones de añadir, eliminar, modificar o listar cursos en el centro educativo.
- Operaciones de añadir, eliminar, modificar o listar profesores del centro.
- Operaciones de añadir, eliminar, modificar o listar asignaturas.
- Asignar profesores a un determinado curso o asignatura.
- Listar materiales didácticos por curso, asignatura o profesor.
- Asignar materiales didácticos a un determinado curso, asignatura o profesor.
- Listar materiales didácticos por curso, asignatura o profesor.

- Inicio y fin de sesión.

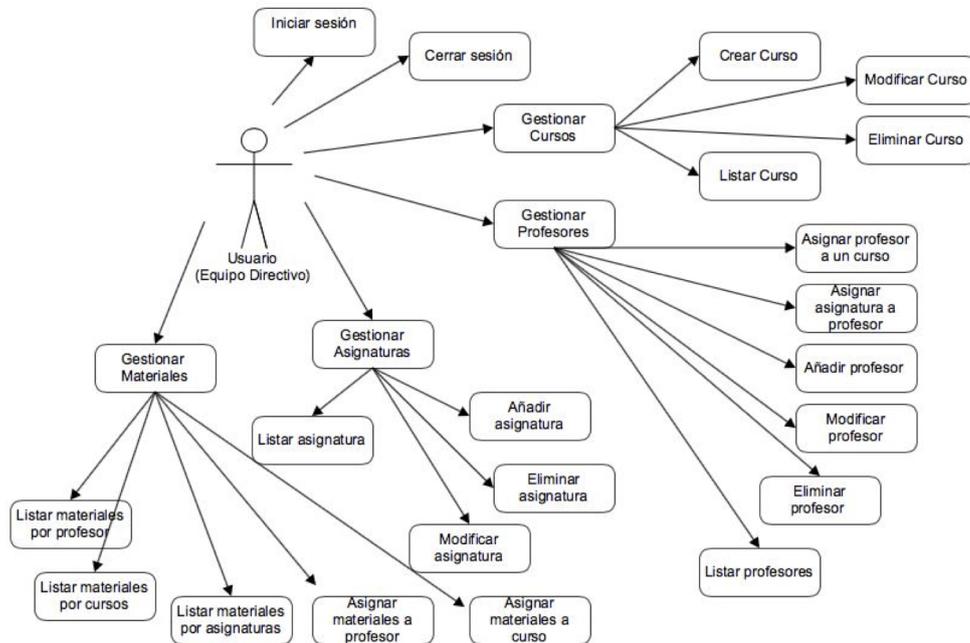


Figura 4.2: Caso de uso del actor equipo directivo

En la figura 4.3 se presentan las acciones que podrá realizar un usuario de tipo profesor. Concretamente las operaciones son:

- Operaciones de gestión de datos personales.
- Listar información de los cursos o asignaturas que imparte cada profesor.
- Asignar materiales por asignaturas o cursos.
- Listar materiales por asignaturas o cursos.
- Gestionar recomendaciones de materiales por parte de los alumnos.
- Inicio y fin de sesión.

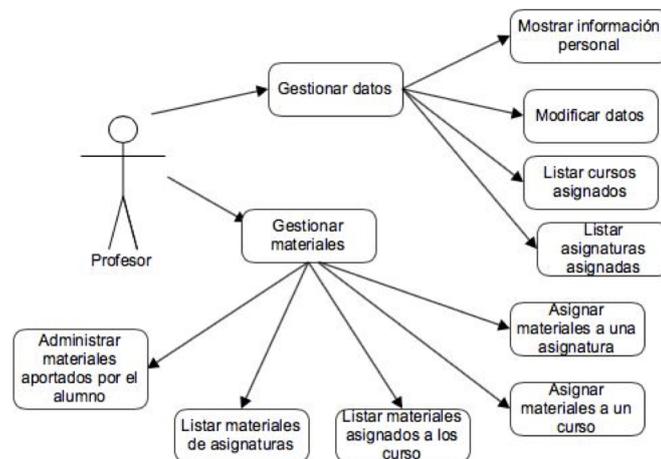


Figura 4.3: Caso de uso del actor profesor

En la figura 4.4 se muestran las acciones que podrá realizar un usuario de tipo alumno. Concretamente las operaciones son:

- Listar materiales didácticos por curso, asignatura o profesor.
- Consultas o modificación de datos personales.
- Recomendar material a un profesor.
- Inicio y fin de sesión.

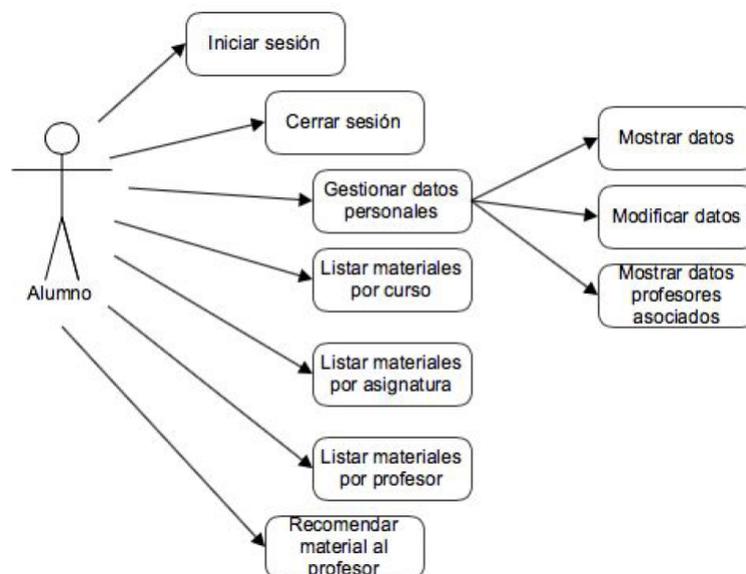


Figura 4.4: Caso de uso del actor alumno

En la figura 4.5 se exponen las acciones que podrá realizar un usuario de tipo padre/madre. Concretamente las operaciones son:

- Consultas diversas sobre datos personales.
- Listar materiales didácticos por centro, curso, profesor o asignatura.
- Listar información sobre sus descendientes.
- Inicio y fin de sesión.

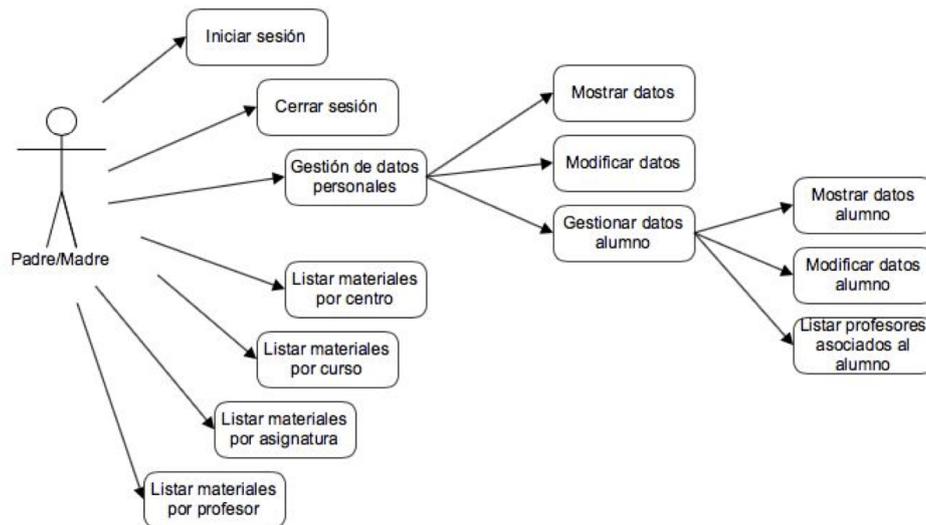


Figura 4.5: Caso de uso del actor padre/madre

En la figura 4.6 se muestran las acciones que podrá realizar un usuario de tipo móvil. Concretamente las operaciones son:

- Consultas diversas sobre materiales didácticos por cursos, profesores o asignaturas.
- Login en el sistema para autenticarse.
- Recomendar materiales curriculares a un/unos profesor/es.

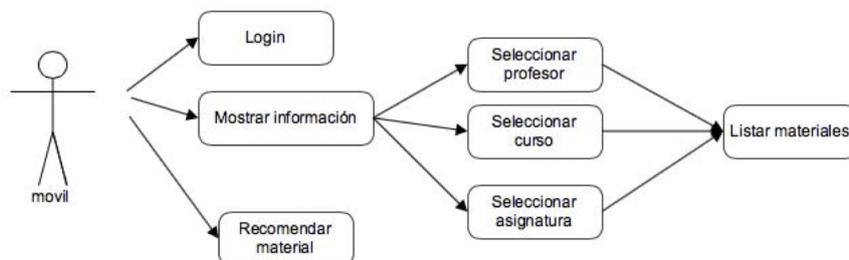


Figura 4.6: Caso de uso del actor móvil

4.2.2. Listado de casos de usos

En el cuadro 4.2 podemos ver una tabla con todos los casos de uso, actores y el código que los identificará en los casos de uso detallados.

Actores	Caso de uso	Código
Administrador, Equipo directivo, Profesor, Alumno, Padre/Madres	Iniciar sesión	A1
Administrador, Equipo directivo, Profesor, Alumno, Padre/Madres	Cerrar sesión	A2
Administrador, Equipo directivo	Crear usuarios	A3
Administrador, Equipo directivo	Modificar usuarios	A4
Administrador, Equipo directivo	Listar usuarios	A5
Administrador, Equipo directivo	Eliminar usuarios	A6
Administrador	Crear centro	A7
Administrador	Modificar centro	A8
Administrador	Listar centros	A9
Administrador	Eliminar centro	A10
Administrador	Crear material	A11
Administrador	Modificar material	A12
Administrador	Eliminar material	A13
Administrador	Listar materiales	A14
Administrador	Consultar materiales por tipo	A15
Administrador	Consultar materiales por centro	A16
Equipo directivo	Crear curso	ED1
Equipo directivo	Modificar curso	ED2
Equipo directivo	Eliminar curso	ED3
Equipo directivo	Listar cursos	ED4
Equipo directivo	Crear asignatura	ED5
Equipo directivo	Modificar asignatura	ED6
Equipo directivo	Eliminar asignatura	ED7
Equipo directivo	Listar asignaturas	ED8
Equipo directivo	Asignar un curso a profesor	ED9
Equipo directivo	Asignar una asignatura a profesor	ED10
Equipo directivo	Asignar un material a profesor	ED11
Equipo directivo, profesor	Asignar un material a un curso	ED12
Equipo directivo, padre/madre, alumno, movil	Listar materiales por profesor	ED13
Equipo directivo, profesor, padre/madre, alumno, movil	Listar materiales por asignatura	ED14
Equipo directivo, profesor, padre/madre, alumno, movil	Listar materiales por curso	ED15
Profesor	Listar cursos asignados	PR1
Profesor	Listar asignaturas asignadas	PR2
Profesor	Asignar material a una asignatura	PR3
Profesor	Gestionar recomendación	PR4

Actores	Caso de uso	Código
Profesor, Alumno, Padre/Madre	Mostrar datos personales	PR5
Profesor, Alumno, Padre/Madre	Modificar datos personales	PR6
Alumno, Padre/Madre	Listar profesores asociados	AL1
Alumno, Móvil	Recomendar material	AL2
Padre/Madre	Listar materiales por centro	PM1
Padre/Madre	Mostrar datos de alumno	PM2
Padre/Madre	Modificar datos del alumno	PM3
Movil	Login sistema	M1

4.3. Diagramas de casos de uso detallados

Una vez definido el diagrama de casos de uso del sistema de los diferentes actores, en este apartado analizaremos un análisis de los requisitos de cada uno de ellos.

4.3.1. Casos de uso detallados del actor Administrador

En este primer apartado se muestran los casos de uso detallados para el actor Administrador.

Código del caso de uso	A1
Descripción	Comprueba que el usuario existe en el sistema y su contraseña
Actores	Administrador, Equipo directivo, Profesor, Alumno, Padre/Madre
Precondiciones	Se introduce la información sobre el usuario y su contraseña
Flujo normal	1. Introducción de los datos 2. Acceso a la aplicación
Flujo alternativo	1. Introducción de los datos erróneos 2. Mensaje de error 3. Se retorna a la interfaz de introducir los datos

Tabla 4.3: Iniciar sesión

Código del caso de uso	A2
Descripción	Permite finalizar la sesión del usuario
Actores	Administrador, Equipo directivo, Profesor, Alumno, Padre/Madre
Precondiciones	Debe existir una sesión iniciada
Flujo normal	1. Cerrar sesión 2. Deja de haber una sesión iniciada
Flujo alternativo	
Postcondiciones	Se vuelve a la interfaz para introducir los datos de iniciar sesión

Tabla 4.4: Cerrar sesión

Código del caso de uso	A3
Descripción	Permite crear un nuevo usuario de tipo equipo directivo para poder iniciar sesión en la aplicación
Actores	Administrador
Precondiciones	Se debe haber iniciado sesión como administrador y rellenar el formulario con los datos del nuevo usuario
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad de nuevo usuario 2. Rellenar el conjunto de datos pertenecientes al usuario 3. Introducir todos los datos correctamente, lo que creará el nuevo usuario del sistema
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar un nuevo usuario 2. Rellenar el conjunto de datos pertenecientes al usuario de forma errónea 3. Al introducir todos los datos incorrectamente se muestra un mensaje de error y se retorna para rellenar nuevamente los campos erróneos
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.5: Crear usuario de Equipo directivo

Código del caso de uso	A4
Descripción	Consiste en modificar un usuario ya existente
Actores	Administrador, Equipo directivo
Precondiciones	Haber iniciado sesión como administrador o usuario de tipo equipo directivo y seleccionar el usuario que se desea modificar
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "modificar usuarios" 2. Seleccionar el usuario que se desea modificar 3. Se realizan los cambios pertinentes, que serán guardados
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú operabilidad "modificar usuarios" 2. Seleccionar el usuario que se desea modificar 3. Se introducen incorrectamente los nuevos datos 4. El sistema muestra un error y permite su rectificación
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.6: Modificar usuario

Código del caso de uso	A5
Descripción	Consiste en enumerar los usuarios registrados
Actores	Administrador, Equipo directivo
Precondiciones	Haber iniciado sesión como administrador o usuario de tipo equipo directivo
Flujo normal	Seleccionar del menú la operabilidad "Listar Usuarios"
Flujo alternativo	
Postcondiciones	

Tabla 4.7: Listar usuarios

Código del caso de uso	A6
Descripción	Permite eliminar un usuario ya existente en la aplicación
Actores	Administrador, Equipo directivo
Precondiciones	Haber iniciado sesión como administrador o usuario de tipo equipo directivo
Flujo normal	1. Seleccionar del menú la operabilidad "eliminar" 2. Se selecciona el usuario y se confirma su borrado del sistema
Flujo alternativo	
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.8: Eliminar usuario

Código del caso de uso	A7
Descripción	Permite crear un nuevo centro
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y no debe existir un identificador de centro utilizado (en nuestro caso el código del centro)
Flujo normal	1. Seleccionar del menú la operabilidad "crear centro" 2. Introducir los datos solicitados 3. Seleccionar la opción de guardar
Flujo alternativo	1. Seleccionar del menú la operabilidad "crear centro" 2. Introducir los datos solicitados, al introducir los datos comentemos algún error 3. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.9: Crear centro

Código del caso de uso	A8
Descripción	Consiste en modificar los datos de un centro existente
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y seleccionar el centro que se desea modificar
Flujo normal	1. Seleccionar del menú la operabilidad "modificar centro" 2. Seleccionar el centro que se desea modificar 3. Se realizan los cambios pertinentes, que serán guardados
Flujo alternativo	1. Seleccionar del menú la operabilidad "modificar centro" 2. Seleccionar el centro que se desea modificar 3. Se introducen incorrectamente los nuevos datos 4. El sistema muestra un error y permite su modificación
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.10: Modificar centro

Código del caso de uso	A9
Descripción	Consiste en enumerar los centros registrados
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador
Flujo normal	Seleccionar del menú la operabilidad "Listar centros"
Flujo alternativo	
Postcondiciones	

Tabla 4.11: Listar centros

Código del caso de uso	A10
Descripción	Permite eliminar un centro ya existente en la aplicación
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y seleccionar el centro que se desea eliminar
Flujo normal	1. Seleccionar del menú la operabilidad "Eliminar centro" 2. Se selecciona el centro y se suprime del sistema
Flujo alternativo	
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.12: Eliminar centro

Código del caso de uso	A11
Descripción	Permite crear un tipo de material didáctico (impresos, gráficos, mixtos, auditivo, tecnológico, ...)
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y no deberá existir el identificador de material (en nuestro caso las siglas del material)
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Crear material" 2. Seleccionar crear material de primer nivel o material de segundo nivel 3. Introducir los datos solicitados 4. Seleccionar la opción guardar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Crear material" 2. Seleccionar crear material de primer nivel o material de segundo nivel 3. Introducir los datos solicitados, al introducir los datos cometemos algún error 4. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.13: Crear material didáctico

Código del caso de uso	A12
Descripción	Consiste en modificar un material ya existente
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y seleccionar un material que se desea modificar
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Modificar material" 2. Seleccionar el material que se desea modificar 3. Se realizan los cambios pertinentes, que serán guardados
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Modificar material" 2. Seleccionar el material que se desea modificar 3. Se introducen incorrectamente los nuevos datos 4. El sistema muestra un error y permite su rectificación
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.14: Modificar material

Código del caso de uso	A13
Descripción	Permite eliminar un material ya existente en la aplicación
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y seleccionar el material que se desea eliminar
Flujo normal	1. Seleccionar del menú la operabilidad "Eliminar material" 2. Se selecciona el material y se suprime del sistema
Flujo alternativo	
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.15: Eliminar material

Código del caso de uso	A14
Descripción	Consiste en enumerar los materiales y submateriales registrados
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador
Flujo normal	Seleccionar del menú la operabilidad "Listar materiales"
Flujo alternativo	
Postcondiciones	

Tabla 4.16: Listar materiales

Código del caso de uso	A15
Descripción	Permite consultar los materiales por su tipo
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y el tipo de material a consultar debe estar dado de alta
Flujo normal	1. Seleccionar del menú la operabilidad "Consultar materiales por tipo" 2. Seleccionar un material específico 3. Pulsamos mostrar
Flujo alternativo	
Postcondiciones	Se mostrará por pantalla los centros asignados a ese tipo de material

Tabla 4.17: Consultar materiales por tipo

Código del caso de uso	A16
Descripción	Permite consultar los materiales asignados a un centro determinado
Actores	Administrador
Precondiciones	Haber iniciado sesión como administrador y el centro a consultar debe estar dado de alta
Flujo normal	1. Seleccionar del menú la operabilidad "Consultar materiales por centro" 2. Seleccionar un centro específico 3. Pulsamos mostrar
Flujo alternativo	
Postcondiciones	Se mostrará por pantalla los materiales didácticos utilizados por el centro seleccionado

Tabla 4.18: Consultar materiales por centro

4.3.2. Casos de uso detallados del actor Equipo directivo

En este apartado se muestran los casos de uso detallados para el actor equipo directivo, los cuales principalmente versarán con las operaciones de administración de profesores, de cursos y de asignaturas, así como la posibilidad de agregar materiales didácticos por diversos criterios.

Código del caso de uso	ED1
Descripción	Permite crear un curso dentro del centro donde se encuentra asignado el usuario del equipo directivo
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo y no debe existir el identificador del curso utilizado (en nuestro caso las siglas del curso)
Flujo normal	1. Seleccionar del menú la operabilidad "Crear curso" 2. Introducir los datos solicitados 3. Seleccionar la opción de guardar
Flujo alternativo	1. Seleccionar del menú la operabilidad "Crear curso" 2. Introducir los datos solicitados, al introducir los datos cometemos algún error 3. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.19: Crear curso

Código del caso de uso	ED2
Descripción	Consiste en modificar un curso ya existente
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo y seleccionar el curso que se desea modificar
Flujo normal	1. Seleccionar del menú la operabilidad "Modificar curso" 2. Seleccionar el curso que se desea modificar 3. Se realizan los cambios pertinentes, que serán guardados
Flujo alternativo	1. Seleccionar del menú la operabilidad "Modificar curso" 2. Seleccionar el curso que se desea modificar 3. Se introducen incorrectamente los nuevos datos 4. El sistema muestra un error y permite su rectificación
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.20: Modificar curso

Código del caso de uso	ED3
Descripción	Permite eliminar un curso ya existente en el centro
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo y seleccionar el curso que se desea eliminar
Flujo normal	1. Seleccionar del menú la operabilidad "Eliminar curso" 2. Se selecciona el curso y se confirma y se suprime del sistema
Flujo alternativo	
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.21: Eliminar curso

Código del caso de uso	ED4
Descripción	Consiste en enumerar los cursos registrados
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo
Flujo normal	Seleccionar del menú la operabilidad "Listar cursos"
Flujo alternativo	
Postcondiciones	Se mostrará por pantalla los cursos existente en el centro donde está el equipo directivo

Tabla 4.22: Listar cursos

Código del caso de uso	ED5
Descripción	Permite crear una nueva asignatura en el centro
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo y no debe existir el identificador de asignatura utilizado (en nuestro caso el código de la asignatura)
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Nueva asignatura" 2. Introducir los datos solicitados 3. Seleccionar la opción de guardar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Nueva asignatura" 2. Introducir los datos solicitados, al introducir los datos cometemos algún error 3. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.23: Crear asignatura

Código del caso de uso	ED6
Descripción	Consiste en modificar una asignatura ya existente
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo y seleccionar la asignatura que se desea modificar
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Modificar asignatura" 2. Seleccionar la asignatura que se desea modificar 3. Se realizan los cambios pertinentes, que serán guardados
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Modificar asignatura" 2. Seleccionar la asignatura que se desea modificar 3. Se introducen incorrectamente los nuevos datos 4. El sistema muestra un error y permite su rectificación
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.24: Modificar asignatura

Código del caso de uso	ED7
Descripción	Permite eliminar una asignatura ya existente en el centro
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo y seleccionar la asignatura que se desea eliminar
Flujo normal	1. Seleccionar del menú la operabilidad "Eliminar asignatura" 2. Se selecciona la asignatura y se confirma y se suprime del sistema
Flujo alternativo	
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.25: Eliminar asignatura

Código del caso de uso	ED8
Descripción	Consiste en enumerar las asignaturas registradas
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo
Flujo normal	Seleccionar del menú la operabilidad "Listar asignaturas"
Flujo alternativo	
Postcondiciones	

Tabla 4.26: Listar asignaturas

Código del caso de uso	ED9
Descripción	Permite añadir un curso ya existente a un profesor dado de alta en el sistema
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo. Además debe existir el curso y el profesor
Flujo normal	1. Seleccionar del menú operabilidad "Añadir curso a profesor" 2. Seleccionar profesor y curso a interrelacionar 3. Pulsar guardar
Flujo alternativo	1. Seleccionar del menú operabilidad "Añadir curso a profesor" 2. Seleccionar profesor y curso a interrelacionar 3. Seleccionar un profesor y ningún curso o viceversa 4. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.27: Asignar un curso a un profesor

Código del caso de uso	ED10
Descripción	Permite añadir una asignatura ya existente a un profesor dado de alta en el sistema
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo. Además debe existir la asignatura y el profesor
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Añadir asignatura a profesor" 2. Seleccionar profesor y asignatura a interrelacionar 3. Pulsar guardar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú operabilidad "Añadir asignatura a profesor" 2. Seleccionar profesor y curso a interrelacionar 3. Seleccionar profesor y ninguna asignatura o viceversa 4. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.28: Asignar una asignatura a profesor

Código del caso de uso	ED11
Descripción	Permite añadir un material ya existente a un profesor dado de alta en el sistema
Actores	Equipo directivo
Precondiciones	Haber iniciado sesión como equipo directivo. Además debe existir el material y el profesor
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Añadir material a profesor" 2. Introducir los datos del material asignar 3. Seleccionar el profesor que se desea añadir 4. Pulsar guardar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Añadir material a profesor" 2. Introducir los datos del material asignar erróneamente 3. No seleccionar ningún profesor a añadir 4. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.29: Asignar un material a profesor

Código del caso de uso	ED12
Descripción	Permite añadir un material ya existente a un curso dado de alta en el sistema
Actores	Equipo directivo, Profesor
Precondiciones	Haber iniciado sesión como equipo directivo. Además debe existir el material y el curso
Flujo normal	1. Seleccionar del menú la operabilidad "Añadir material a un curso" 2. Introducir los datos del material a asignar 3. Seleccionar el curso que se desea establecer 4. Pulsar guardar
Flujo alternativo	1. Seleccionar del menú la operabilidad "Añadir material a un curso" 2. Introducir los datos del material asignar erróneamente 3. No seleccionar el curso que se desea establecer 4. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.30: Asignar un material a un curso

Código del caso de uso	ED13
Descripción	Permite consultar los materiales asignados a un determinado profesor
Actores	Equipo directivo, padre/madre, alumno, móvil
Precondiciones	Debe existir el profesor en el sistema
Flujo normal	1. Seleccionar del menú la operabilidad "Consultar materiales por profesor" 2. Seleccionar el profesor 3. Pulsar buscar
Flujo alternativo	1. Seleccionar del menú la operabilidad "Consultar materiales por profesor" 2. No seleccionar un profesor del sistema 3. Se muestra un mensaje de error
Postcondiciones	Se mostrará en pantalla los materiales asignados al profesor determinado

Tabla 4.31: Listar materiales por profesor

Código del caso de uso	ED14
Descripción	Permite consultar los materiales asignados a una determinada asignatura
Actores	Equipo directivo, profesor, padre/madre, alumno, móvil
Precondiciones	Debe existir la asignatura en el sistema
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Consultar materiales por asignatura" 2. Seleccionar la asignatura 3. Pulsar buscar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Consultar materiales por asignatura" 2. No seleccionar la asignatura 3. Se muestra un mensaje de error
Postcondiciones	Se mostrará en pantalla los materiales asignados a la asignatura seleccionada

Tabla 4.32: Listar materiales por asignatura

Código del caso de uso	ED15
Descripción	Permite consultar los materiales asignados a un determinado curso
Actores	Equipo directivo, profesor, padre/madre, alumno, móvil
Precondiciones	Debe existir el curso en el sistema
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Consultar materiales por curso" 2. Seleccionar el curso 3. Pulsar buscar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Consultar materiales por curso" 2. No seleccionar un curso 3. Se muestra un mensaje de error
Postcondiciones	Se mostrará en pantalla los materiales asignados a un curso en concreto

Tabla 4.33: Listar materiales por curso

4.3.3. Casos de uso detallados del actor Profesor

En este apartado se muestran los casos de uso detallados para el actor profesor.

Código del caso de uso	PR1
Descripción	Permite consultar los cursos asignados que tiene un determinado profesor en el curso actual
Actores	Profesor
Precondiciones	Debe existir el profesor en el sistema
Flujo normal	Seleccionar del menú la operabilidad "Consultar cursos"
Flujo alternativo	
Postcondiciones	Se mostrará en pantalla la información de los cursos asignados

Tabla 4.34: Listar cursos asignados

Código del caso de uso	PR2
Descripción	Permite consultar las asignaturas establecidas a un determinado profesor en el curso actual
Actores	Profesor
Precondiciones	Debe existir el profesor en el sistema
Flujo normal	Seleccionar del menú la operabilidad "Consultar asignaturas"
Flujo alternativo	
Postcondiciones	Se mostrará en pantalla la información de las asignaturas seleccionada

Tabla 4.35: Listar asignaturas asignadas

Código del caso de uso	PR3
Descripción	Permite añadir un material a una de las asignaturas que tiene asignada el profesor logueado
Actores	Profesor
Precondiciones	Debe iniciar sesión como profesor y debe existir la asignatura
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Asignar material a una asignatura" 2. Introducir los datos del material 3. Seleccionar la asignatura 4. Pulsar guardar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Asignar material a una asignatura" 2. Introducir los datos del material erróneamente 3. No seleccionar ninguna asignatura 4. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.36: Asignar material a una asignatura

Código del caso de uso	PR4
Descripción	Permite justificar la adicción de un material aportado por el alumno a un curso o asignatura
Actores	Profesor
Precondiciones	El alumno debe pertenecer a un curso o asignatura donde el profesor imparte sus enseñanzas
Flujo normal	1. Seleccionar del menú la operabilidad "Gestiona recomendación" 2. Leer la información de un material aportado por el alumno 3. Pulsar agregar
Flujo alternativo	
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.37: Gestionar recomendación

Código del caso de uso	PR5
Descripción	Permite consultar los datos que se encuentran almacenados en el sistema
Actores	Profesor, Padre/Madre, Alumno
Precondiciones	Debe existir el usuario
Flujo normal	Seleccionar del menú la operabilidad "Datos personales"
Flujo alternativo	
Postcondiciones	Se mostrará por pantalla la información

Tabla 4.38: Mostrar datos personales

Código del caso de uso	PR6
Descripción	Consiste en modificar los datos un usuario ya existente
Actores	Profesor, Padre/Madre, Alumno
Precondiciones	Debe existir el usuario
Flujo normal	1. Seleccionar del menú la operabilidad "Modificar datos personales" 2. Se realizan los cambios pertinentes, que serán guardados
Flujo alternativo	1. Seleccionar del menú la operabilidad "Modificar datos personales" 2. Se introducen incorrectamente los nuevos datos 3. El sistema muestra un error y permite su rectificación
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.39: Modificar datos personales

4.3.4. Casos de uso detallados del actor Alumno

En este apartado se muestran los casos de uso detallados para el actor alumno, los cuales principalmente versarán sobre las operaciones de consulta de materiales didácticos y recomendaciones de materiales.

Código del caso de uso	AL1
Descripción	Permite consultar la información de los profesores que tiene asociado un alumno
Actores	Alumno, Padre/Madre
Precondiciones	Debe existir el profesor
Flujo normal	Seleccionar del menú la operabilidad "Consultar profesores"
Flujo alternativo	
Postcondiciones	Se mostrará por pantalla los profesores asociados

Tabla 4.40: Listar profesores asociados

Código del caso de uso	AL2
Descripción	Consiste en recomendar un material al profesor asociado a una asignatura o curso
Actores	Alumno, Móvil
Precondiciones	Debe existir el profesor y el curso o la asignatura
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Recomendar material" 2. Introducir los datos solicitados del material 3. Seleccionar el profesor asociado a dicho material 4. Pulsar recomendar
Flujo alternativo	<ol style="list-style-type: none"> 1. Seleccionar del menú la operabilidad "Recomendar material" 2. Introducir los datos solicitados del material erróneamente 3. No seleccionar ningún profesor 4. Se muestra un mensaje de error
Postcondiciones	Se mostrará un mensaje de confirmación

Tabla 4.41: Recomendar material

4.3.5. Casos de uso detallados del actor Padre/Madre

En este apartado se muestran los casos de uso detallados para el actor padre/madre, los cuales principalmente versarán sobre las operaciones de consulta de materiales didácticos así como consultar información de sus descendientes matriculados en un centro.

Código del caso de uso	PM1
Descripción	Permite consultar los materiales didácticos asociados al centro donde se encuentran los hijos (alumnos) matriculados
Actores	Padre/Madre
Precondiciones	Deben existir el centro y el/los alumno/s asociados a dicho centro
Flujo normal	1. Seleccionar del menú la operabilidad "Listar materiales por centro" 2. Seleccionar el centro 3. Pulsar sobre mostrar
Flujo alternativo	1. Seleccionar del menú la operabilidad "Listar materiales por centro" 2. No seleccionar ningún centro 3. Se muestra un mensaje de error
Postcondiciones	Se mostrará por pantalla la información solicitada

Tabla 4.42: Listar materiales por centro

Código del caso de uso	PM2
Descripción	Permite mostrar la información de los alumnos que tienen a cargo un padre/madre
Actores	Padre/Madre
Precondiciones	Debe existir el alumno
Flujo normal	1. Seleccionar del menú la operabilidad "Mostrar datos alumnos" 2. Seleccionar un alumno asociado 3. Pulsar sobre mostrar
Flujo alternativo	1. Seleccionar del menú la operabilidad "Mostrar datos alumnos" 2. Al no seleccionar ningún alumno se mostrará un mensaje de error
Postcondiciones	Se mostrará por pantalla la información solicitada

Tabla 4.43: Mostrar datos de alumno

Código del caso de uso	PM3
Descripción	Consiste en modificar los datos de un usuario ya existente
Actores	Padre/Madre
Precondiciones	Debe existir el alumno
Flujo normal	1. Seleccionar del menú la operabilidad "Modificar datos alumnos" 2. Seleccionar un alumno 3. Introducir los datos solicitados 4. Pulsar sobre guardar
Flujo alternativo	1. Seleccionar del menú la operabilidad "Modificar datos alumnos" 2. No seleccionar ningún alumno 3. Introducir los datos solicitados erróneamente 4. Se mostrará un mensaje de error
Postcondiciones	Se muestra un mensaje de confirmación

Tabla 4.44: Modificar datos del alumno

4.3.6. Casos de uso detallados del actor Móvil

En este apartado se muestran las operaciones que se pueden realizar con el dispositivo móvil: seleccionar materiales por profesor, curso o asignatura y recomendar materiales. En la realización de este caso de uso tendremos una única restricción: el usuario tendrá que estar registrado en el sistema para poder acceder a la aplicación.

Código del caso de uso	M1
Descripción	El usuario deberá loguearse para acceder a la aplicación móvil
Actores	Móvil
Precondiciones	Debe existir el usuario en el sistema
Flujo normal	Introducir username y password del usuario que desea acceder a la aplicación
Flujo alternativo	Si no existe el usuario se muestra un mensaje de error
Postcondiciones	Se muestra la siguiente pantalla de la aplicación

Tabla 4.45: Login sistema

4.4. Diagramas de secuencia

Un diagrama de secuencia permite mostrar la forma en que los objetos se comunican entre sí al transcurrir el tiempo. En este apartado, y de forma generalizada, se muestra el diagrama de secuencia de la aplicación web así como de la aplicación móvil. En el capítulo de implementación se mostrará con mayor detalle dichas operaciones.

4.4.1. Diagrama de secuencia de la aplicación web

En el punto anterior, hemos visto diagramas estáticos que nos permitirán identificar a los principales elementos de la aplicación, pero ahora podemos centrarnos mejor en como interactúan entre ellos.

En la figura 6.4 podemos ver el diagrama.

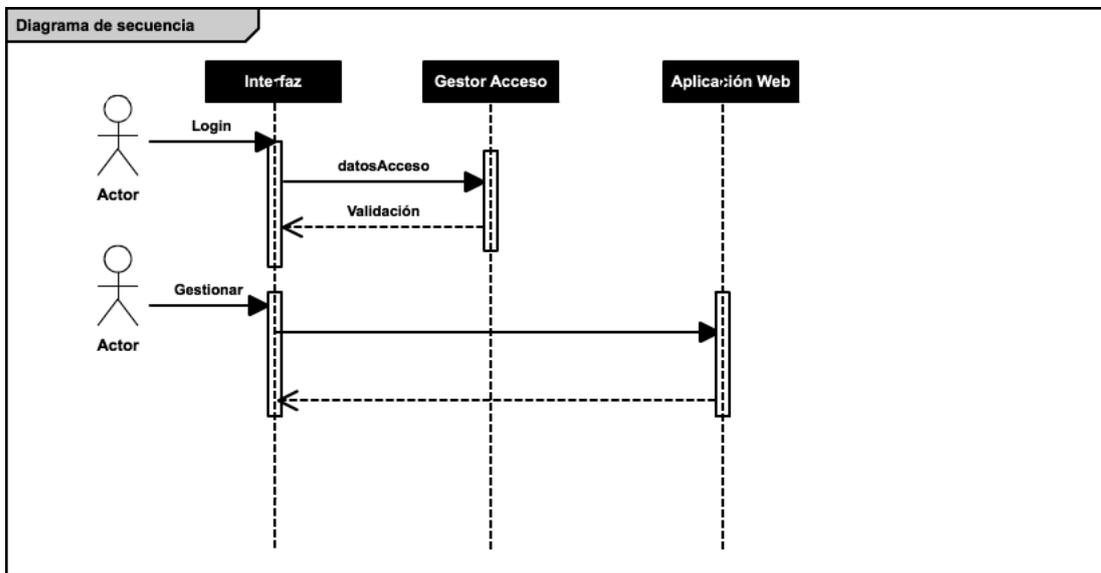


Figura 4.7: Diagrama de secuencia de la aplicación web

4.4.2. Diagrama de secuencia de la aplicación móvil

En esta sección podemos consultar el diagrama de secuencia de la aplicación Android. En la figura 4.8 podemos ver este diagrama.

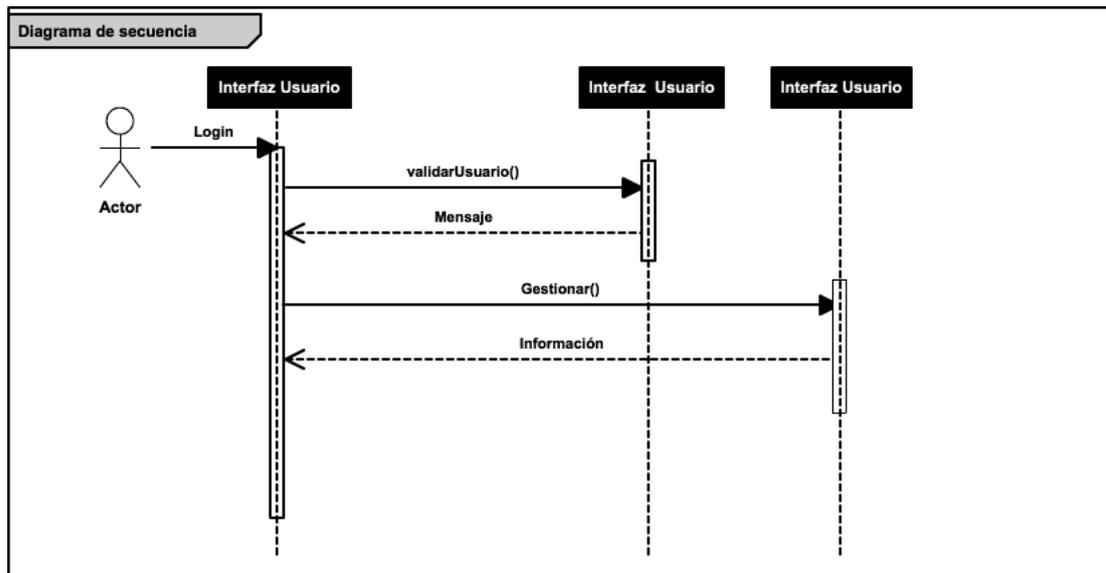


Figura 4.8: Diagrama de secuencia Android

Capítulo 5

Diseño del Sistema

Contenidos

5.1. Arquitectura del sistema	57
5.1.1. Capa de Presentación	58
5.1.2. Capa Lógica	59
5.1.3. Capa de datos	59
5.2. Diseño del sistema	59
5.2.1. Diseño de la interfaz	60
5.2.2. Acceso Web	60
5.2.3. Acceso Móvil	69
5.3. Base de datos	72
5.3.1. Servidor de base de datos MySQL	72
5.4. Diseño de la seguridad	76
5.4.1. Seguridad en la base de datos	76
5.4.2. Seguridad en las comunicaciones	76

Este capítulo describiré la arquitectura del sistema que he utilizado en las aplicaciones a desarrollar.

Finalmente, se analizará uno de los aspectos más importantes del desarrollo realizado, la seguridad de cada uno de sus componentes.

5.1. Arquitectura del sistema

A partir del análisis del capítulo anterior se observa que se deben desarrollar los siguientes sistemas:

- Sistemas de administración/control por parte del administrador en la aplicación web.
- Sistemas de administración/control por parte del perfil de usuario del equipo directivo en la aplicación web.
- Sistemas de administración/control por parte del perfil de usuario del profesor en la aplicación web.

- Sistema de gestión por parte del perfil de usuario del alumno en la aplicación web.
- Sistema de gestión por parte del perfil del usuario del padre/madre en la aplicación web.
- Sistema de gestión instalado en un cliente Android.

Para realizar el sistema web se utiliza una arquitectura basada en el modelo vista controlador (MVC). Este tipo de arquitectura software es un patrón que se ocupa de separar los datos y la lógica de negocio de una aplicación, de la interfaz de usuario y el módulo que se encarga de gestionar los eventos y las comunicaciones. Para conseguir este modelo MVC, se propone la creación de tres capas diferentes, estas son el modelo, la vista y el controlador.

Este patrón se basa en la idea de reutilización de código y la separación de conceptos, propiedades que buscan facilitar el desarrollo de las aplicaciones y sus posterior mantenimiento.

En la figura 5.1 podemos visualizar la arquitectura utilizada.

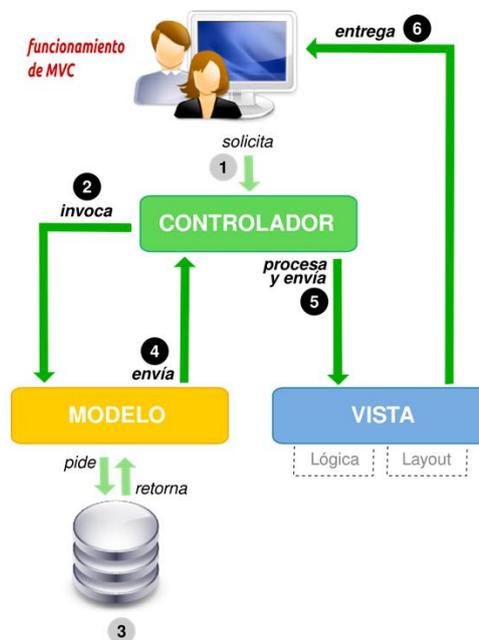


Figura 5.1: Arquitectura de la aplicación web

5.1.1. Capa de Presentación

En esta capa se muestra la interfaz de usuario basada en web, que se encarga de mostrar el sistema al usuario de una forma intuitiva y amigable, con las funcionalidades asociadas de enviar y recibir información del usuario en un proceso mínimo cuyo fin será validar las informaciones generadas por los usuarios. Estará formada por las tecnologías ejecutadas por el propio cliente (HTML5, CSS3, Bootstrap, JQuery y Javascript) como por tecnologías de servidor orientadas a gestionar la capa de presentación (JSP).

En determinados requerimientos funcionales se ha conectado esta capa con la capa lógica/negocio utilizando la tecnología AJAX, una serie de técnicas de desarrollo web que permiten

enviar y recibir datos entre sistema cliente (por ejemplo, un navegador web) y un servidor de modo asíncrono.

En nuestro caso, disponemos de dos capas de presentación, una para la aplicación web y otra para la aplicación Android.

5.1.2. Capa Lógica

Es la capa principal donde se realiza el procesamiento y transformación de los datos recibidos de la capa de presentación, estableciendo las reglas que deben cumplirse. Esta capa se comunica con la capa de datos para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Esta capa se compone de un patrón DAO (Data Access Object) y un servicio REST.

5.1.3. Capa de datos

Esta capa es la que permite dar persistencia a los datos recibidos. Estará formada por un gestor de base de datos que se encarga de realizar todo el almacenamiento de datos, recibir peticiones para almacenar datos o recuperar información desde la capa de lógica.

El sistema gestor de base de datos utilizado es MySQL. Este sistema contiene la base de datos de la aplicación, además de un conjunto de tablas adaptadas a nuestra aplicación.

5.2. Diseño del sistema

El propósito del diseño es la de crear una arquitectura para la naciente implementación, [...] el diseño arquitectural sólo puede comenzar una vez que el equipo tenga un entendimiento razonable de los requerimientos del sistema. [...] Tanto diseño, como el análisis, nunca terminan realmente hasta que el sistema final es entregado. Durante esta fase, se alcanza un cierto grado de culminación al poner en su lugar la mayoría de las decisiones estratégicas de diseño y al establecer políticas para diversos problemas tácticos. [...] El diseño se enfoca en la estructura, estática y dinámica. Su propósito principal es el de crear el 'esqueleto' concreto del sistema sobre el cual todo el resto de la implementación se basa. [25]

Estas palabras definen claramente qué es el diseño. La creación de la estructura básica del sistema es la tarea clave, aunque también se buscan otras cosas, en particular patrones que simplifiquen el diseño.

La parte de diseño la he dividido en dos bloques generalmente, por un lado está el diseño del servicio web y por otro el diseño del servicio móvil. Además se define el diseño de datos, que representa el modelo entidad/relación de la aplicación su paso a tablas para la base de datos.

El desarrollo del servicio web ha sido realizado utilizando como base el framework Bootstrap, que ha facilitado en gran medida la implementación del sistema, y utilizando como servidor de aplicaciones Apache Tomcat 8 para ejecutar el código realizado en Java.

Así mismo, sobre la misma aplicación se ha realizado la implementación del sitio web utilizando para ello los lenguaje de programación HTML5, CSS3, JQuery, Javascript y, de modo testimonial JSP:

- A través de HTML5, el contenido es gestionado de un modo más dinámico que con versiones anteriores de este lenguaje, permitiendo añadir elementos como vídeos o interaccionar con los elementos de los formularios.
- Por medio de CSS3, se ha proporcionado estilo al sitio para dar un entorno de navegación atractivo al usuario final. Se ha prestado especial atención a este punto, pues he querido diseñar una interfaz sencilla, amigable e intuitiva para la interacción de los diferentes perfiles de usuario con la aplicación web.
- Se ha utilizado JQuery y Javascript para gestionar la lógica interna del sitio, así como las comunicaciones con los servicios REST a través de invocaciones asíncronas (AJAX). Esto ha sido realizado de este modo debido al dinamismo que otorga al sistema.
- JSP como tecnología de integración entre la parte de cliente y la parte de servidor, siendo utilizado para gestionar los contenidos generados por la aplicación.

El desarrollo del servicio móvil se ha realizado para Android, que consistirá en una pasarela de datos entre el servidor de datos MySQL y el servicio web.

5.2.1. Diseño de la interfaz

La interfaz supone la frontera entre el sistema informático y el exterior del mismo. Define como interactúan usuario y ordenador y tiene un papel importante en la aceptación del sistema.

Una interfaz debe ser sencilla, consistente, atractiva y debe permitir al usuario interactuar con el sistema de forma cómoda.

Se ha establecido una estructura de diseño diferenciada en tres partes: la cabecera, el cuerpo y el pie.

- Cabecera: contiene la información relativa al centro educativo así como al perfil del usuario.
- Cuerpo: se divide en dos partes, por un lado está la zona principal, donde se muestra la mayoría de los datos del sistema y por otro lado está el menú lateral, donde se encuentran las diferentes opciones con las que el usuario puede interactuar con el sistema.
- Pie: no se le ha incorporado ningún tipo de información.

En la figura 5.2se muestra el esqueleto de diseño del sistema eXtreBooks.

5.2.2. Acceso Web

El sistema eXtreBooks dispone de una página web con toda la información necesaria para poder utilizar el sistema, tanto la gestión de centros como el listado de alumnos matriculados, así como los materiales curriculares gestionados por los profesores.

5.2.2.1. Requisitos mínimos

Los requisitos necesarios para poder interactuar en el sistema eXtreBooks pasan por disponer de una conexión a Internet, y un navegador web que admita las siguientes características:

- HTML 5

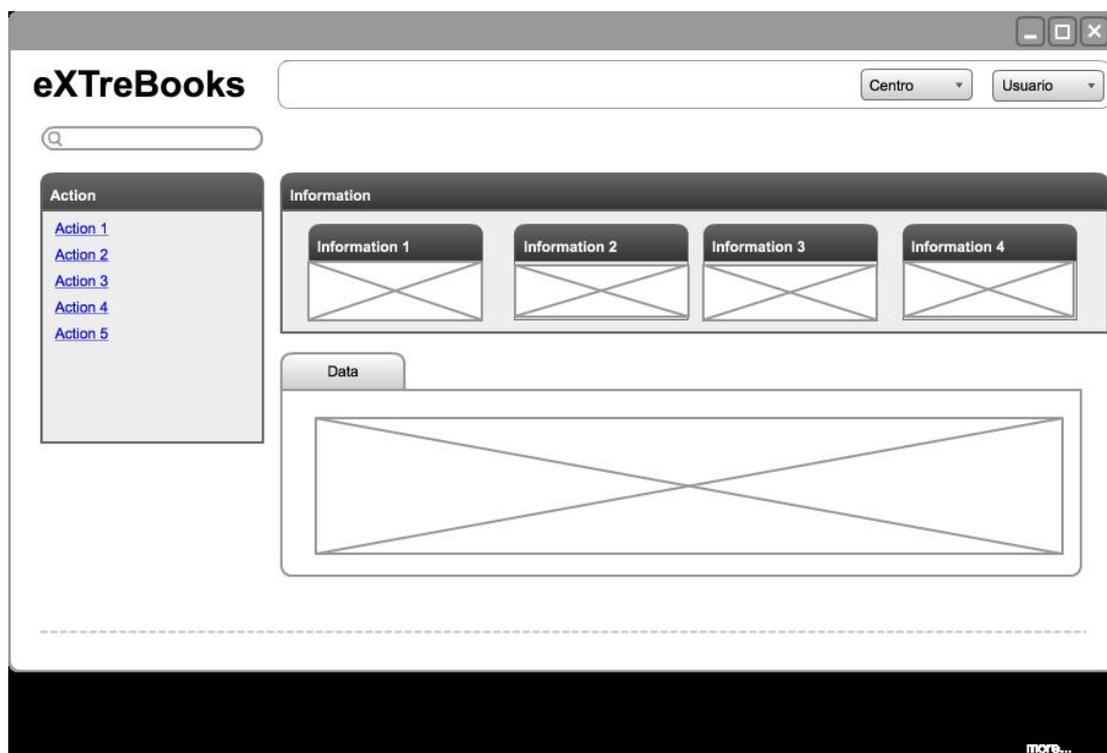


Figura 5.2: Esquema web del sistema

- CSS 3
- Javascript

5.2.2.2. Navegación por el sitio

El sitio web se encuentra formado por distintas secciones en las que el usuario podrá encontrar la información que precise acerca del sistema eXTreBooks:

- Inicio de sesión: esta pantalla es la puesta en marcha del sistema. En ella se observará el logotipo de la aplicación y un mensaje de bienvenida.
- Pantalla principal: esta pantalla tiene una estructura en común para todos los usuarios, y tiene como misión simplificar la generación de informaciones.
- Pantalla de información: esta pantalla se encuentra situada en la parte superior de la interfaz principal, y tiene como misión dar una información rápida sobre los datos del centro educativo y los datos del perfil del usuario.

Pantalla de inicio de sesión

En la figura 5.3 podemos ver el diseño de interfaz de la pantalla de inicio de sesión.

Esta pantalla será común para los distintos perfiles de usuarios. Dispone de una caja de inicio de sesión con los campos necesarios para introducir el email del usuario y contraseña.

Se dispondrá además, de un mecanismo de recuperación de contraseña a través del correo electrónico del usuario.



Figura 5.3: Diseño de la interfaz de inicio de sesión

Pantalla principal

La figura 5.4 muestra la pantalla que nos aparece una vez logueado en el sistema. Esta pantalla se divide en cuatro partes:

- Menú de opciones del usuario: este menú se encuentra situado a la izquierda de la interfaz principal, y dependiendo del perfil del usuario permitirán unas acciones u otras.
- Menú indicativo: este menú se encuentra situado en el centro de la interfaz principal, y tiene como misión mostrar información sobre los datos del sistema.
- Menú información: este menú se encuentra situado en el centro de la interfaz y debajo del menú indicativo. Cada perfil de usuario será asesorado sobre las distintas funciones que puede realizar.

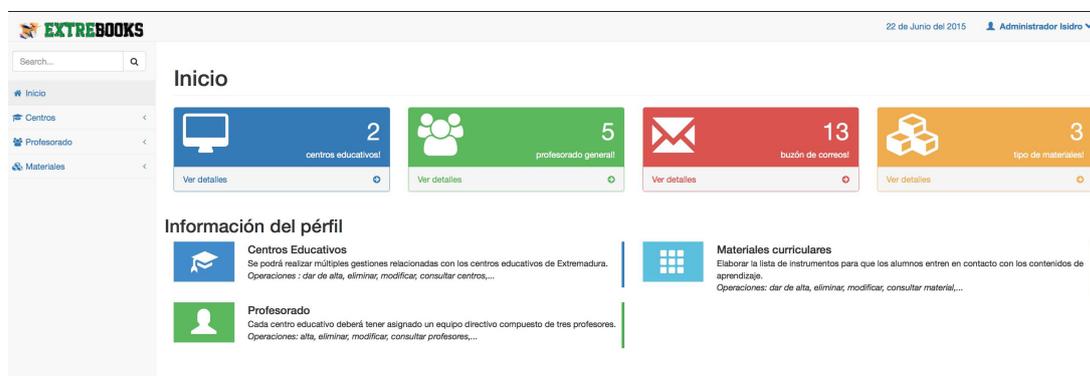


Figura 5.4: Diseño de la interfaz de la pantalla principal

Pantalla de Información

La figura 5.5 muestra la pantalla de información, diferenciando dos opciones desplegadas para la muestra de información.



Figura 5.5: Diseño de la interfaz de información

5.2.2.3. Interfaz sistema web del administrador

La interfaz utilizada por el administrador del sistema para realizar sus funcionalidades ha sido implementada de forma amigable e intuitiva donde las acciones disponibles están en el menú de opciones. Pudiendo diferenciar las siguientes funcionalidades:

Pantalla de gestión de centros

La figura 5.6 muestra una de las funcionalidades del centro: alta centro, listar centro y modificar centro.

Figura 5.6: Diseño de la interfaz de alta del centro

Pantalla de gestión de profesorado

La figura 5.7 muestra una de las funcionales del profesorado general: alta profesor, listar profesor y asignar dirección equipo directivo del centro.

Pantalla de gestión de materiales

La figura 5.8 muestra una de las funcionalidades de la gestión de materiales: nueva categoría y listar categoría.

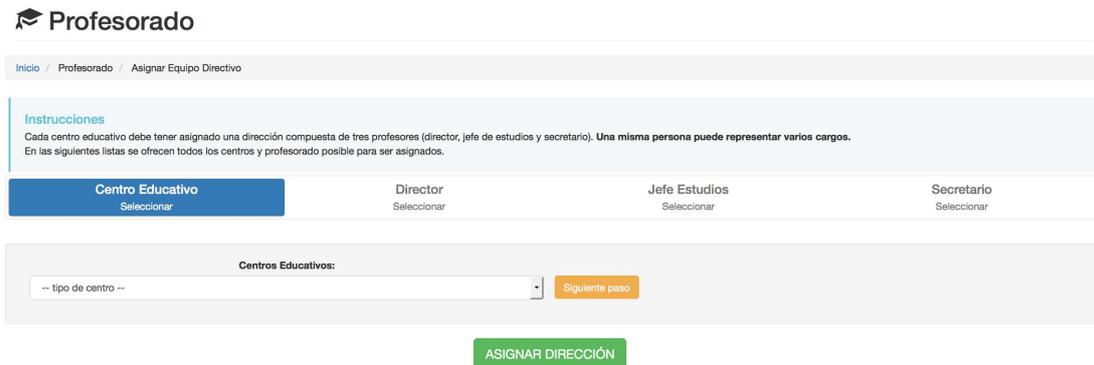


Figura 5.7: Diseño de la interfaz de asignar equipo directivo



Figura 5.8: Diseño de la interfaz de listar tipo de materiales

5.2.2.4. Interfaz sistema web del equipo directivo

La interfaz utilizada por el equipo directivo de un centro, al igual que las interfaces del resto de usuario se ha diseñado de forma muy intuitiva para facilitar su utilización y comprensión. Desatacamos las siguientes funcionalidades:

Pantalla de gestión de cursos académicos

La figura 5.9 muestra una de las funcionalidades de cursos académicos: nuevo curso y actualizar cursos.

En el centro de la pantalla se muestra un formulario que debemos rellenar para poder crear un nuevo curso dentro del centro. Una vez relleno pulsamos el botón registrar y se nos mostrará un mensaje de confirmación de la creación de usuario.

Pantalla de gestión de profesorado del centro educativo

La figura 5.10 muestra una de las funcionalidades del profesorado: nuevo profesor, listar profesor y actualizar profesor.

En esta pantalla se muestran los profesores presentes en el centro mostrando toda su información a través de una lista desplegable.

Pantalla de gestión de asignaturas

La figura 5.11 muestra una de las funcionalidades de las asignaturas: nueva asignatura, actualizar asignatura y listar materiales.

En esta pantalla se muestra un listado de todas las asignaturas de un centro, con las opciones de editar o eliminar cualquier asignatura.

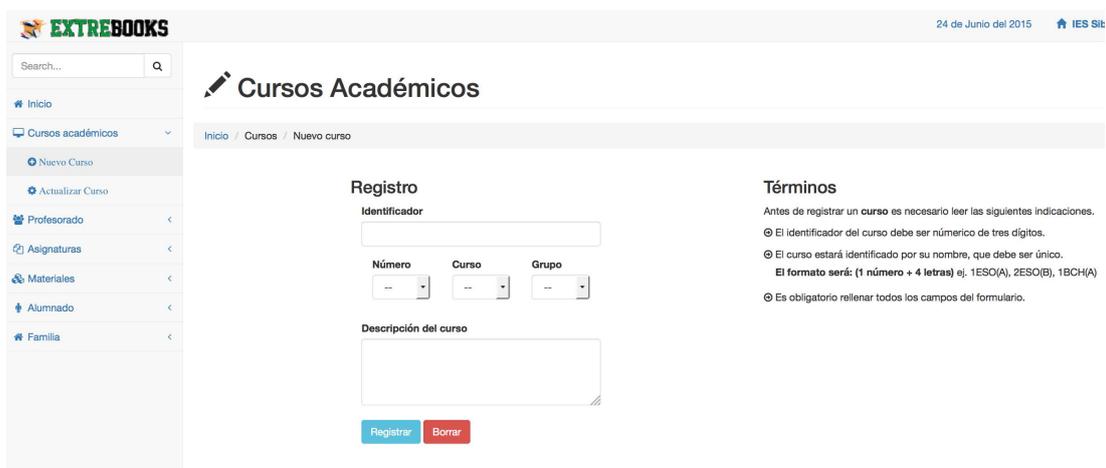


Figura 5.9: Diseño de la interfaz de nuevo curso



Figura 5.10: Diseño de la interfaz de listar profesores

Pantalla de gestión de materiales

La figura 5.12 muestra una de las funcionalidades de los materiales: nueva categoría de material, nuevo material y actualizar materiales.

En esta pantalla se muestra un listado de todos los materiales asignados al centro, especificando profesor y asignatura donde se utiliza ese material. Al hacer clic sobre el material se nos mostrará más información sobre sus características.

Pantalla de gestión de alumnos/as

La figura 5.13 muestra una de las funcionalidades del alumnado: nuevo alumno y listar alumnado.

En el centro de la pantalla se muestra un formulario para la creación de un nuevo alumno. Al pulsar sobre el botón registrar se muestra un mensaje de confirmación.

Pantalla de gestión de familias

La figura muestra una de las funcionalidades de las familias: nueva familia y listar familias.

En el centro de la pantalla se muestra un formulario para dar de alta al tutor/a de un alumno del centro. Al pulsar sobre el botón se muestra un mensaje de confirmación.

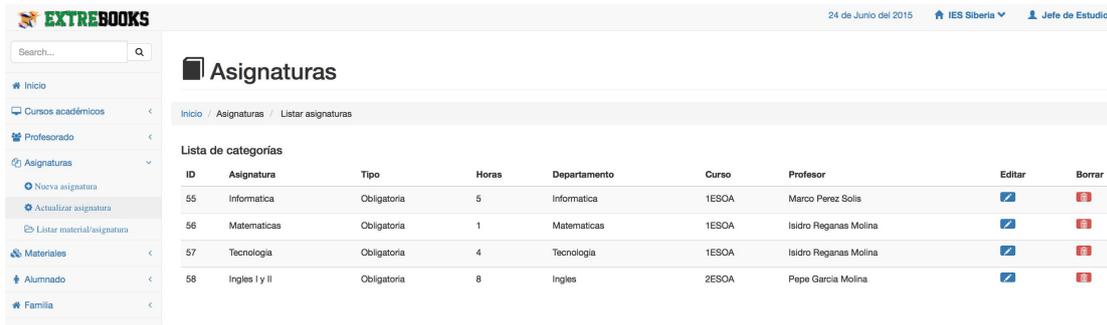


Figura 5.11: Diseño de la interfaz de editar asignaturas

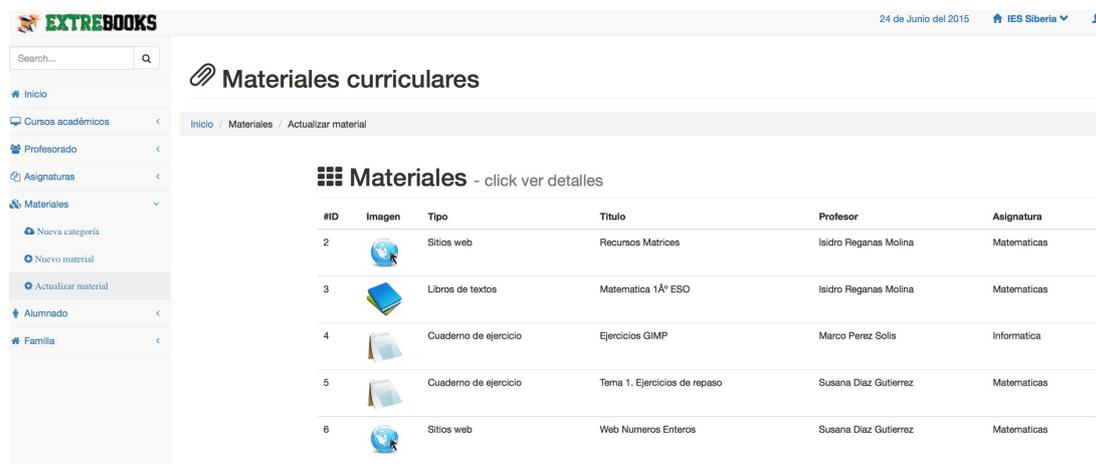


Figura 5.12: Diseño de la interfaz de listar materiales

5.2.2.5. Interfaz sistema web del profesor

La interfaz utilizada por los profesores, al igual que el resto de interfaces se caracteriza por un diseño fácil e intuitivo de manejar.

Destacamos las siguientes funcionalidades:

Pantalla de gestión de cursos académicos

La figura 5.15 muestra la funcionalidad de listar los cursos a los cuales están asignados un determinado profesor.

Esta pantalla muestra información por medio de un panel informativo sobre los cursos asignados a un profesor, así como otros profesores que imparten en ese mismo curso y los alumnos matriculados.

Pantalla de gestión de asignaturas

La figura 5.16 muestra la funcionalidad de listar las asignaturas impartidas por un determinado profesor.

Esta pantalla muestra información por medio de un panel informativo sobre las asignaturas que imparte un profesor y los alumnos matriculados.

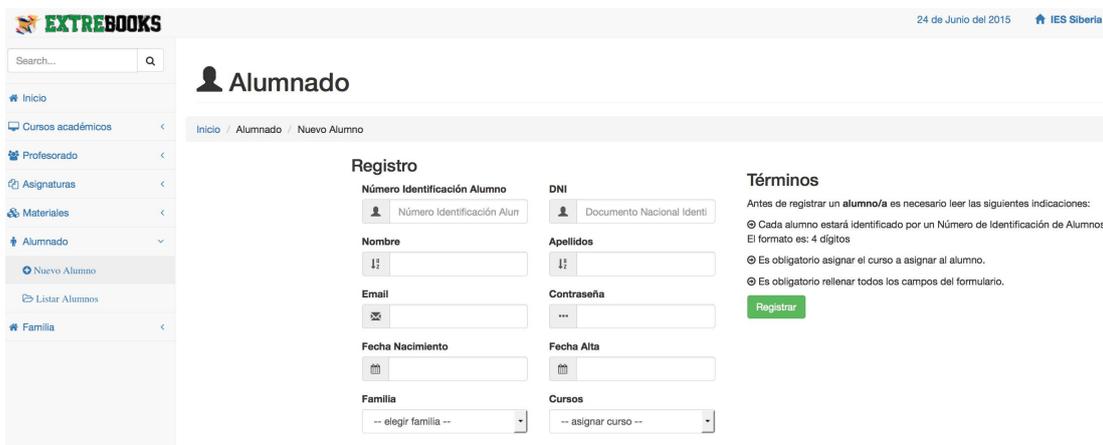


Figura 5.13: Diseño de la interfaz de alta alumno

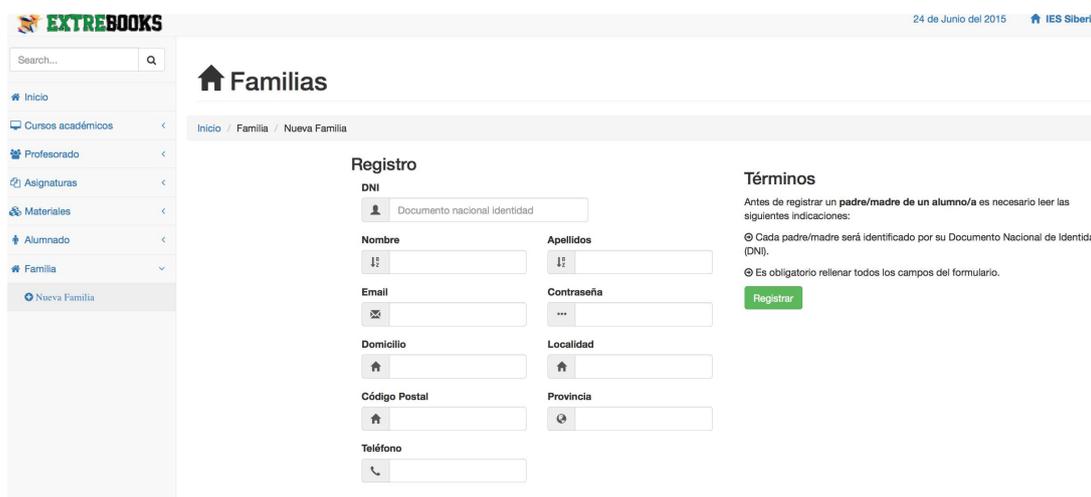


Figura 5.14: Diseño de la interfaz de alta de familia

Pantalla de gestión de materiales

La figura 5.17 muestra una de las funcionalidades de los materiales: asignar material, listar materiales y validar material.

Esta pantalla muestra información sobre los materiales que un profesor determina en sus asignaturas y cursos.

5.2.2.6. Interfaz sistema web del alumno

La interfaz utilizada por los alumnos, al igual que el resto de interfaces es caracterizada por un diseño fácil e intuitivo de manejar.

Destacamos las siguientes funcionalidades:

Pantalla de gestión de asignaturas

La figura 5.18 muestra la funcionalidad de listar las asignaturas cursadas por un determinado alumno.

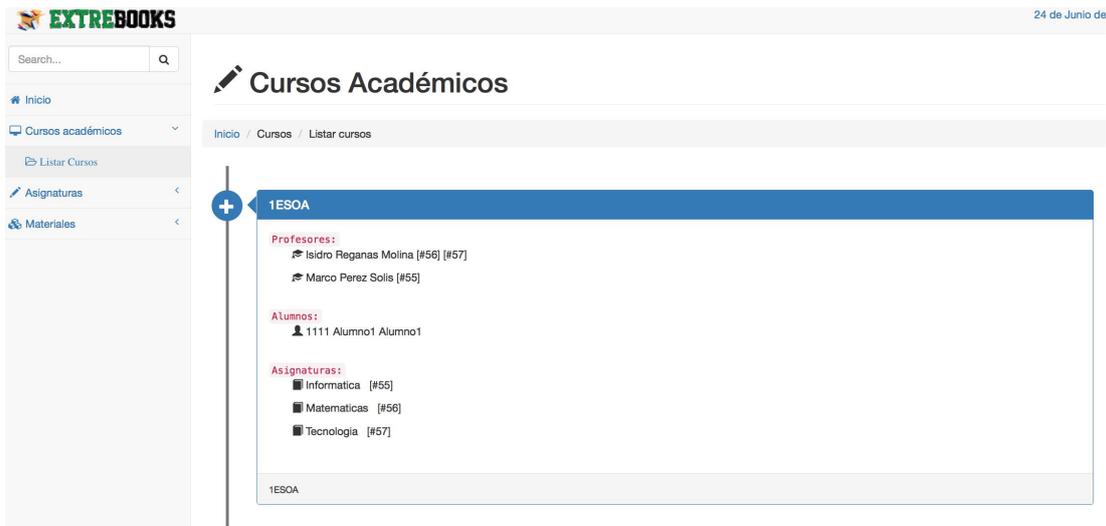


Figura 5.15: Diseño de interfaz de listar cursos académicos

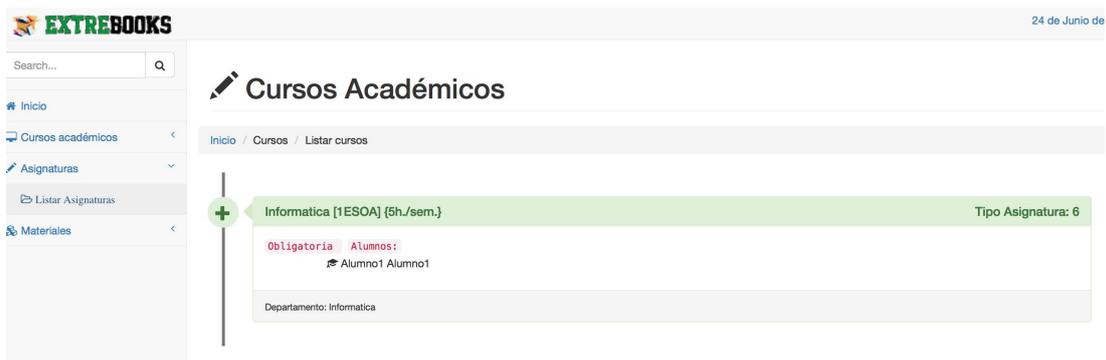


Figura 5.16: Diseño de interfaz de listar asignaturas

En esta pantalla se muestra información por medio de listas desplegables de las asignaturas en las que está matriculado un alumno.

Pantalla de gestión de materiales

La figura muestra una de las funcionalidades de los materiales: listar materiales y recomendar material.

En esta pantalla se muestra información por medio de listas desplegable de los materiales asignados tanto al curso como a las asignaturas matriculadas por el alumno.

5.2.2.7. Interfaz sistema web de la familia

La interfaz utilizada por la familia, al igual que el resto de interfaces es caracterizada por un diseño fácil e intuitivo de manejar.

Destacamos las siguientes funcionalidades:

Pantalla de gestión de asignaturas

La figura 5.20 muestra la funcionalidad de listar la información de las asignaturas y mate-

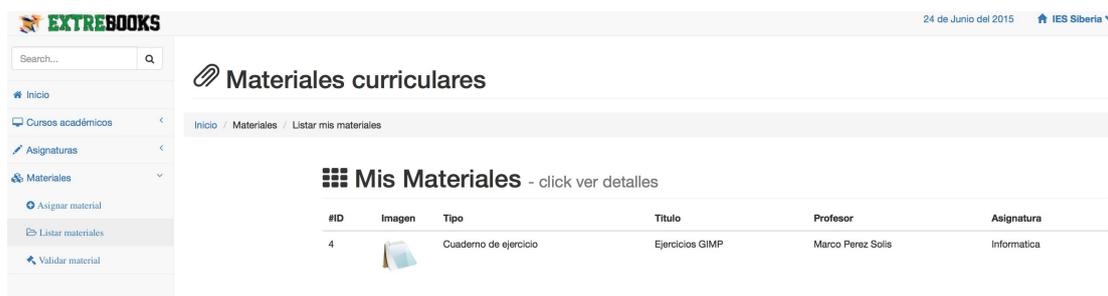


Figura 5.17: Diseño de la interfaz de listar materiales



Figura 5.18: Diseño de la interfaz de listar asignaturas

riales asignados a las clases de los hijos asignados.

Esta pantalla muestra información de los materiales asignados a las asignaturas o curso donde sus hijos se encuentran matriculados.

5.2.3. Acceso Móvil

La interfaz del sistema móvil es simple e intuitiva, siendo la misma para todos los usuarios. Seguidamente explico las diferentes pantallas en la que se compone el diseño móvil.

Pantalla principal

El diseño de la pantalla principal es simple, está compuesta de un mensaje de información como podemos ver en la figura 5.21.

Transcurrido unos segundos, de forma automática se redireccionará a la pantalla de inicio de sesión.

Pantalla de inicio de sesión

Esta pantalla tiene como finalidad implementar seguridad sobre los datos del usuario móvil. La implementación se realizará a través de un simple formulario donde se especificará un email y una password para poder acceder al sistema.



Figura 5.19: Diseño de la interfaz de listar materiales

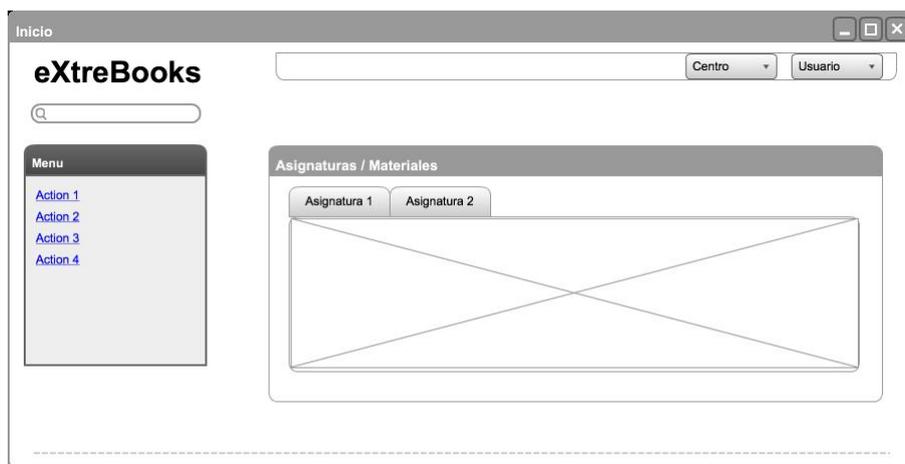


Figura 5.20: Diseño interfaz de listar materiales

La figura 5.21 muestra un boceto del diseño de la interfaz de inicio de sesión.

Pantalla de gestión de asignaturas

En la figura 5.22 se muestra la pantalla donde se podrá consultar el listado de asignaturas que cada alumno tiene asignado. Al pulsar en cada asignatura se mostrarán los detalles de la misma y los materiales asignados por el profesor.

Pantalla de gestión de profesores

En la figura 5.22 se muestra la pantalla donde se podrá consultar el listado de profesores que cada alumno tiene asignado. Al pulsar en cada profesor se mostrarán los detalles del mismo así como la opción de recomendar cualquier tipo de material.

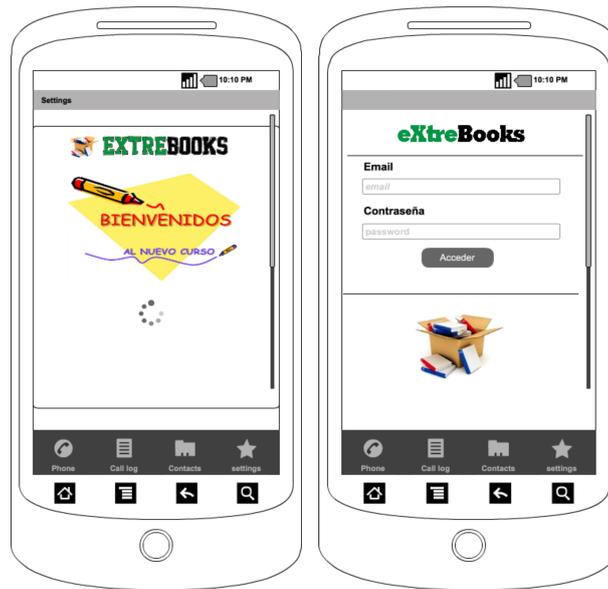


Figura 5.21: Diseño de la interfaz inicio de sesión



Figura 5.22: Diseño de la interfaz de profesores y asignaturas

Pantalla de información del centro

En la figura se muestra la pantalla donde se podrá consultar la información asociada al centro, donde se encuentra el alumno matriculado, dando la opción de poder enviar un email, conocer la ruta de llegada,...



Figura 5.23: Diseño de la interfaz del centro

5.3. Base de datos

Una base de datos es un conjunto de datos, que relacionados entre sí de un modo coherente, tiene una finalidad concreta.

Directamente relacionado con ellas, un Sistema Gestor de Base de datos (DBMS) es un conjunto de programas y herramientas de propósito general, que permitirán a los interesados describir los tipos que formarán la base de datos y sus relaciones, proceder con el almacenamiento de dichos datos, consultar los datos existentes y compartir la base de datos para facilitar su acceso concurrente. [22]

5.3.1. Servidor de base de datos MySQL

El servidor de base de datos MySQL es una de las soluciones más utilizada a nivel global, ofreciendo un servidor rápido, multiusuario, multi-hilo y robustez.

Entre otras, los servidores de bases de datos MySQL cuentan con las siguientes características: [12]

- Gran compatibilidad ya que MySQL se encuentra disponible en múltiples plataformas.
- Los servidores MySQL han sido diseñados para funcionar en varias capas y utilizando módulos independientes.
- Ejecución multi-hilo a través de los hilos del núcleo del sistema operativo.
- Utilización de motores de almacenamiento tanto transaccionales como no transaccionales.

En el caso de eXtreBooks, se ha optado por utilizar la versión libre del servidor de base de datos con el fin de gestionar la persistencia en el servicio web.

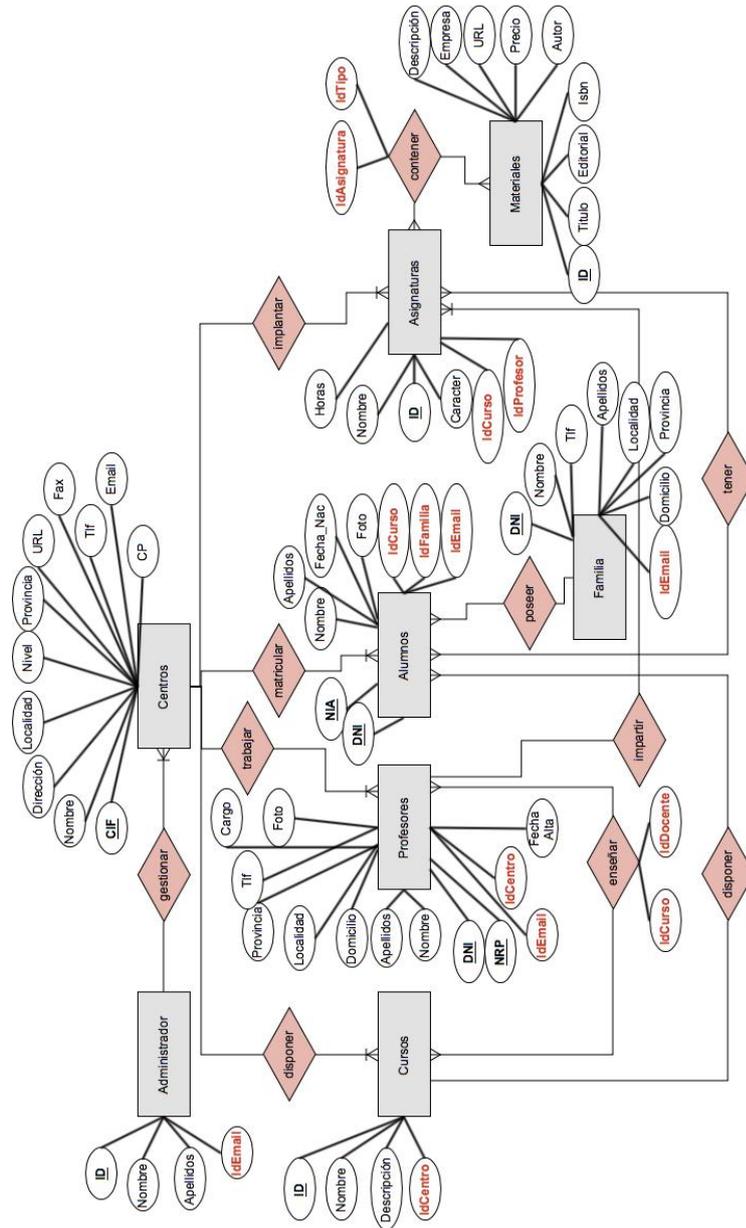


Figura 5.24: Diagrama entidad-relación de la base de datos

El diseño de la base de datos, a pesar de tener una funcionalidad sencilla, ha sido realizado pensando en la escalabilidad en un futuro próximo. Por ello, se ha decidido crear las siguientes tablas que permitirán gestionar el sistema en función de los perfiles de los usuarios, tal y como puede verse en la figura 5.24:

- **Registro:** almacena información relativa a los usuarios autorizados a utilizar el sistema.
Está compuesta por el email del usuario que será único en la aplicación y será el identificador de usuario de la aplicación, el campo contraseña y el campo tipo que determinará el tipo de usuario que accede al sistema.
- **Centros:** almacena información relativa a los centros donde se implantará el sistema.
Entre la información que se almacena destacamos: el cif del centro que será el identificador, nombre del centro, domicilio, email, fax,... y nivel educativo del centro.
- **Cursos:** almacena información de las aulas que serán gestionadas en cada centro.
Entre la información que almacena destacamos: el identificador del aula, nombre del aula, descripción e identificador del centro asociado al aula.
- **Asignaturas:** almacena información relativa a las asignaturas implantadas en los centros.
Entre la información que almacena destacamos: el identificador de la asignatura, nombre de la asignatura, departamento al que pertenece,...así como el identificador del curso donde se imparte esa asignatura y el profesor tutor de la asignatura.
- **Materiales:** almacena información sobre los materiales curriculares a incorporar en las asignaturas.
Esta tabla es una de la más importante, pues en ella se almacena los datos de los materiales didácticos que utiliza un profesor o se asignan a una asignatura. Entre la información que se almacena destacamos: el tipo del material, el identificador del material, el título del material, el autor, ...así como el identificador de la asignatura y el identificador del profesor donde se utiliza el material.
- **Recomendaciones:** almacena información sobre las aportaciones de los alumnos a las asignaturas cursadas.
Esta tabla contiene la información de los materiales que el usuario puede recomendar a un profesor para una determinada asignatura. Entre la información que se almacena destacamos: el identificador que será único, el identificador del alumno que hace la recomendación, la descripción así como el identificador de la asignatura y el identificador del profesor, al cual se le va a recomendar dicho material.
- **Dirección:** almacena información relativa a los usuarios que son equipo directivo de un centro.
Está compuesta por el identificador del centro al cual se le va a asignar el equipo directivo y los identificadores de los profesores a los cuales se le van asignar como equipo directivo del centro.
- **Docentes:** almacena información sobre los profesores de un centro.

Entre la información que almacena destacamos: el número de registro personal y documento nacional de identidad, que será el identificador único de cada profesor, nombre y apellidos del profesor, dirección,... así como el identificador del centro al que pertenece.

- **Alumnos:** almacena información relativa a los alumnos matriculados en un centro.

Entre la información que almacena destacamos: el número de identificación del alumno, que será único en la aplicación, nombre y apellidos del alumno, fecha de nacimiento, ... así como el identificador del curso al que pertenece y el identificador de la familia.

- **Administrador:** almacena información sobre el administrador general del sistema.

Esta tabla contiene información sobre el identificador del administrador que será único, nombre y apellidos del administrador y el email para su acceso al sistema.

- **Familias:** almacena información sobre los padres/madres de los alumnos matriculados en un centro.

Se utiliza para relacionar al alumno con la familia que lo tutoriza. Entre la información que almacena destacamos: el identificador del tutor/a legal, el nombre y apellidos, domicilio familiar,...

En la figura 5.25 se muestra un ejemplo del anterior modelo, en relacional con datos reales para de esta forma nos ayude a comprender mejor el sistema. Como puede observarse:

- La *tabla Administrador* almacena los datos personales del administrador único del sistema (tipo = 1), cuyo acceso está controlado por la tabla Registro donde el campo password está cifrado por el algoritmo AES.
- La *tabla Registro* almacena los usuarios que se han definido en el sistema. Como puede observarse, cada usuario tiene un tipo que determinará las funcionalidades a realizar en el sistema.

En el ejemplo, el usuario Luis tiene el tipo = 2, que significa que pertenecerá al equipo directivo del centro, mientras que el usuario Carmen tiene el tipo = 3 haciendo referencia a una profesora del centro.

- La *tabla Centros* almacena las informaciones de los centros educativos que va generando el administrador del sistema, en nuestro caso, dos centros del nivel de IES.
- La *tabla Profesores* almacena dos profesores identificados por su número de registro personal (*nrp*) del centro educativo con cif igual a 'X1234567X'.
- La *tabla Cursos* almacena dos cursos representados por su id, que están implantados en el centro con identificador cif igual a 'X1234567X'.
- La *tabla Asignaturas* nos permite relacionar las asignaturas que tienen asociadas cada profesor, curso y centro. Concretamente, la asignatura de Matemáticas está asociada al profesor con identificador '12345100' que corresponde al nombre de 'Luis García', que se imparte en el curso '001' correspondiente a 'IesoA' dentro del centro 'X1234567X' perteneciente al IES Siberia Extremeña.

- La *tabla Materiales* almacena los materiales didácticos que se van a utilizar. Está relacionada con la tabla profesores y la tabla asignaturas para concretar que materiales utilizan cada profesor y que materiales utilizan cada asignatura, clasificándose por su tipo.

5.4. Diseño de la seguridad

La información que se va a desarrollar va a manejar información sensible, como son contraseñas de acceso al sistema, números de teléfonos, códigos de identificación, ... Por este motivo, la seguridad será uno de los aspectos a tener en cuenta a la hora de realizar la aplicación.

La seguridad de la aplicación debe integrarse en cada uno de los elementos que forma el sistema, basándonos en los siguientes principios: [13]

- Privacidad. La privacidad consiste en la ocultación de la información utilizando mecanismos de cifrado para que terceras personas, aun accediendo a ella, no puedan procesarla.
- Autenticación. La autenticación debe evitar que terceras personas que no estén registradas puedan acceder al sistema.
- Autorización. La autorización debe garantizar que cada usuario sólo podrá acceder a la información que tiene permiso o derecho.
- Integridad: La integridad debe mantener la consistencia de la información de la aplicación.

Estos mecanismos serán implantados en cada uno de los elementos de nuestro sistema.

5.4.1. Seguridad en la base de datos

Los principios de seguridad citados anteriormente deberán ser implantados en nuestro sistema de base de datos de la siguiente forma:

- Cada usuario del sistema tendrá asignado un email y una contraseña para poder acceder a la base de datos (**Autenticación**).
- Cada usuario del sistema tendrá asignadas funcionalidades distintas (**Autorización**).
- La información sensible del sistema estará cifrada. Si accede a la base de datos la contraseña será visualizada pero no legible (**Privacidad**).
- La integridad de los datos generados por el sistema estará controlada por el propio SGBD (**Integridad**).

5.4.2. Seguridad en las comunicaciones

La seguridad de las comunicaciones vendrá implantada principalmente por la utilización del protocolo HTTPS (Hypertext Transfer Protocol Secure), es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, que garantiza que la información entre cliente y servidor no sea legible para terceras personas.

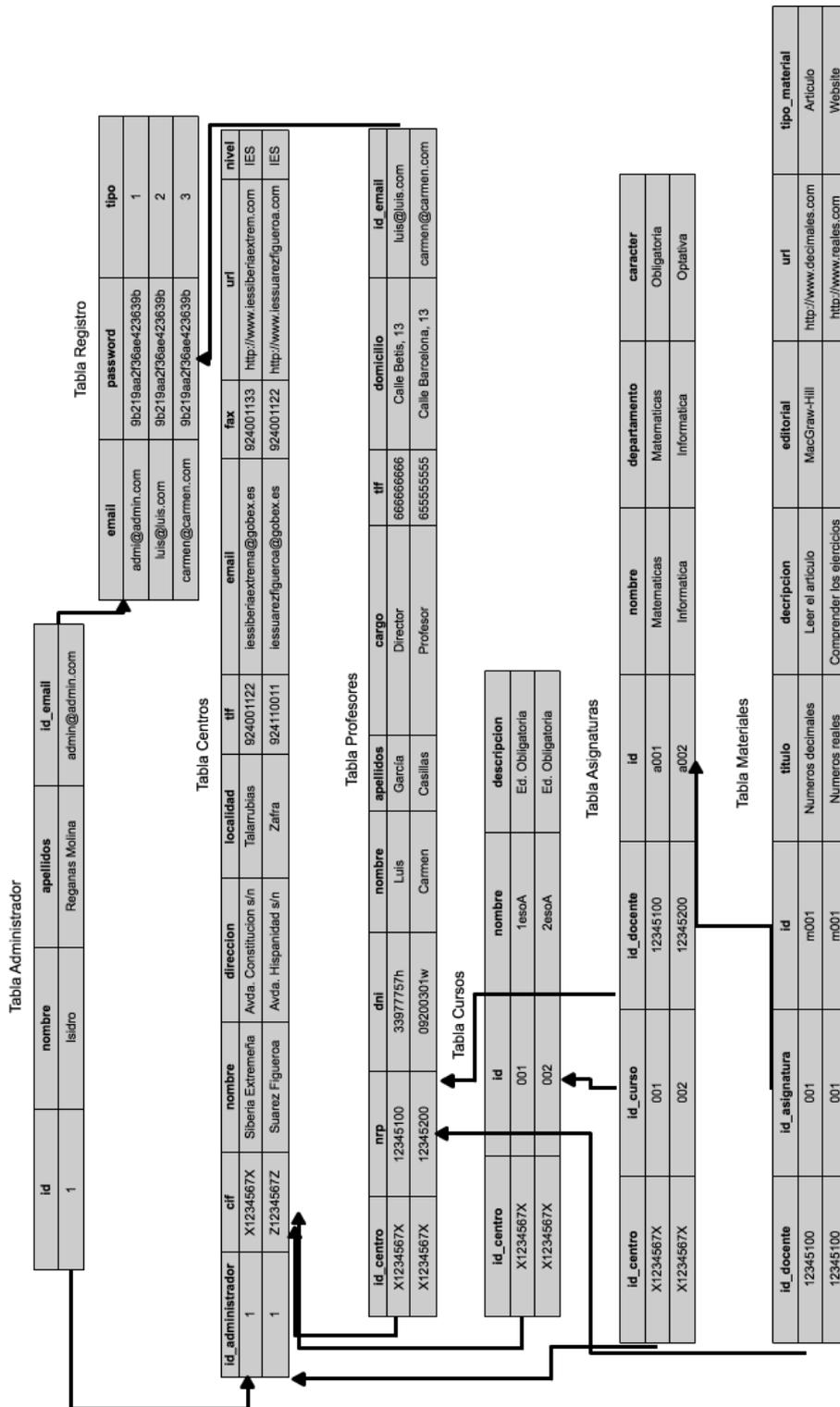


Figura 5.25: Modelo del sistema eXtreBooks

Para ello, la página web codifica la sesión con certificado digital. De este modo, garantiza que la información que envíe desde dicha página no podrá ser interceptada y utilizada por terceros. Estos certificados de seguridad son conocidos como SSL (Secure Sockets Layer), que crean un canal cifrado (cuyo nivel depende del servidor remoto y del navegador utilizado por el cliente). [26]

La seguridad ofrecida por SSL responde a dos principios:

1. El cifrado de la información. Todos los datos que circulan se convierten en ininteligibles, excepto entre el usuario que establece la conexión y el servidor en el que está alojado el sitio web.
2. La autenticación. Garantiza la identidad de sitio ante una autoridad de certificación.

El principio paso a paso: [16]

La **clave privada** se instala en el servidor. Esta clave es la que crea el sello de certificación del sitio web.

La **clave pública** es la otra parte del certificado SSL, también instalada en su sitio web. Permite que los usuarios del sitio cifren sus datos, algo muy útil si necesitan comunicar información sensible (número de tarjeta de crédito, número de cuenta...). Los datos se cifran antes de ser enviados. La clave privada es la clave espejo, la única que podrá descifrar los datos.

En la figura 5.26 se puede ver un esquema del funcionamiento:

Paso 1. Solicitud de establecimiento de una conexión segura por SSL.

Paso 2. Presentación del certificado. Comprobaciones realizadas en el certificado: validez y firma por un tercero de confianza.

Paso 3. Transmisión de una clave de encriptación única (codificada con la clave pública del servidor).

Paso 4. Desencriptado de la clave de encriptación por el servidor utilizando su clave privada (establecimiento de la conexión segura).

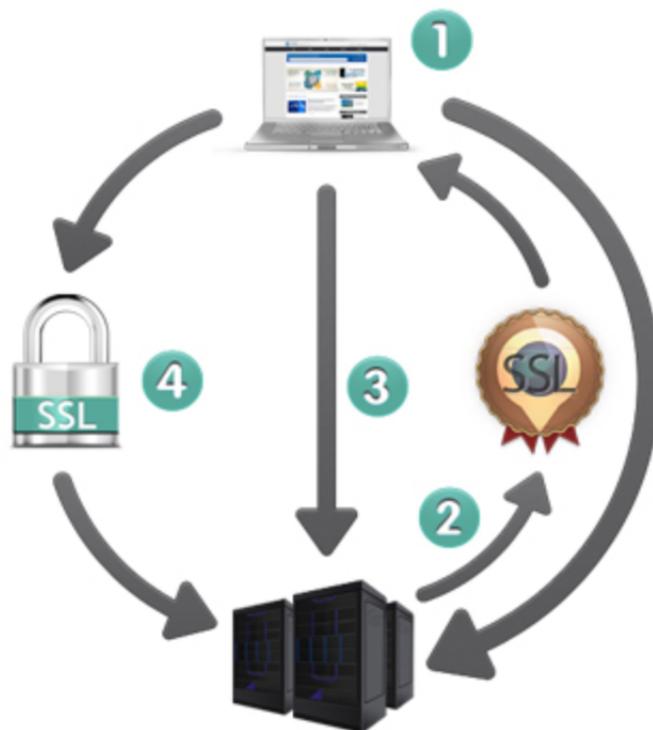


Figura 5.26: Seguridad ofrecida por SSL [16]

Capítulo 6

Implementación

Contenidos

6.1. Análisis y diseño de la Arquitectura	81
6.1.1. Capa de datos	82
6.1.2. Capa lógica	84
6.1.3. Capa de presentación	90
6.2. Implementación de la seguridad del sistema	92
6.2.1. Seguridad en la base de datos	92

Una vez realizado el análisis y diseño del sistema a desarrollar, llega el momento de realizar la especificación o pautas seguidas en su implementación. En este apartado, se describirá cómo y con qué tecnologías se ha implementado el proyecto, detallando los aspectos más importantes.

IMPORTANTE: Debido a las dimensiones del proyecto, la implementación que se ha llevado a cabo sólo hace referencia al servicio web, dejando la implementación del servicio móvil para trabajos futuros. En la fase de análisis, se decidió abordar tanto todas las funcionalidades del sistema web como todos los posibles actores que intervienen en él. Se ha realizado la implementación del administrador del portal, del actor equipo directivo, del actor profesor así como del actor alumno, dejando también para futuros trabajos la implementación del actor familia.

6.1. Análisis y diseño de la Arquitectura

Durante la fase de diseño, se esbozó la arquitectura del sistema, sin entrar en aspectos tecnológicos, como lenguajes de programación, entornos, plataformas que se iban a utilizar, . . . En esta sección se analiza, para cada unas de las capas establecidas en la fase de diseño: presentación, lógica y datos, qué tecnologías se han usado.

En la figura 6.1 podemos visualizar la arquitectura del sistema utilizada. [20]



Figura 6.1: Arquitectura del sistema

En la arquitectura se puede observar que se ha realizado la implementación de dos aplicaciones: una aplicación web accesible desde cualquier navegador y una aplicación usando un dispositivo móvil Android. Además, existe una capa intermedia, común a ambas aplicaciones, que aporta la lógica del sistema. Finalmente, se encuentra la capa de datos soportada por una base de datos MySQL.

Seguidamente, analizaré de forma detallada, los distintos componentes de cada una de las capas de la arquitectura utilizada.

6.1.1. Capa de datos

En la etapa de diseño se especificó que el sistema está soportado por un gestor de base de datos de software libre, como es MySQL [12]. Allí se plasmó el modelo entidad-relación para conocer la estructura lógica, por lo que ahora daré a conocer la estructura física de dicho modelo. En la figura 6.2 se puede observar dicho modelo.

El código SQL para la creación de tablas no presenta dificultades, tanto sólo esclarecer que la tabla *'Registro'* contiene la información confidencial de los usuarios, y por tanto, dicha información estará cifrada, concretamente el campo password. Este atributo se ha definido como "varbinary" para poder almacenar información binaria y posteriormente se realizará sobre él la operación de cifrado. En el **Anexo I** se puede ver la importación de la base de datos.

Se usa como algoritmo de cifrado AES (Advanced Encryption Standard), que utiliza dos funciones que permite encriptar (AES_ENCRYPT) y desencriptar (AES_DECRYPT) datos. El cifrado AES usa una codificación con una clave de 128 bits de longitud, considerándose como las funciones de encriptado más seguras disponibles actualmente en MySQL.

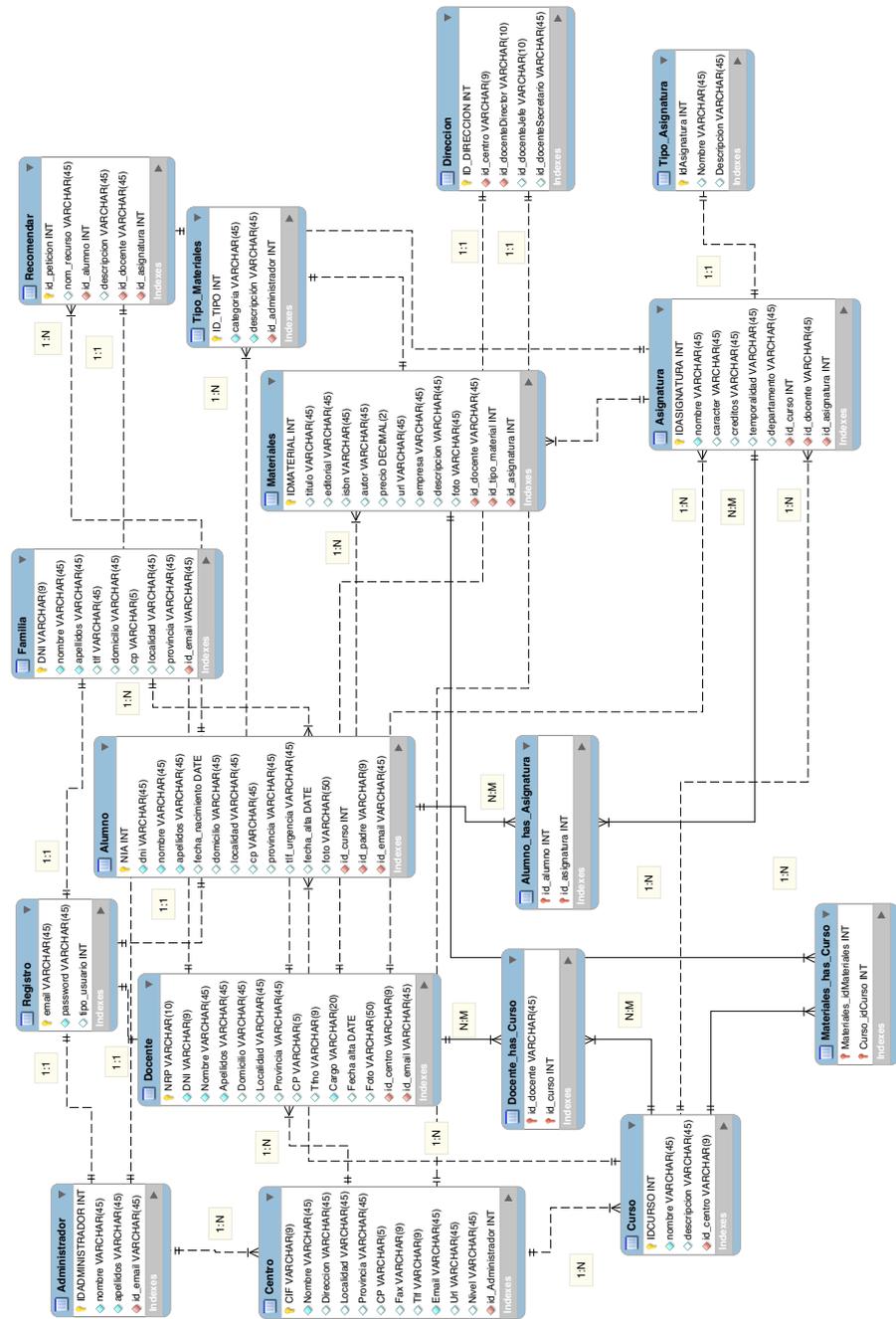


Figura 6.2: Implementación de la base de datos

6.1.2. Capa lógica

Esta capa intermedia permite realizar la comunicación entre los clientes y los datos almacenados en el sistema, adaptándola a las vistas del usuario. Dentro de esta capa lógica se pone de manifiesto los siguientes elementos.

6.1.2.1. Entorno de desarrollo

El desarrollo de la capa lógica se ha realizado bajo la plataforma de desarrollo Eclipse Juno como plataforma de integración de herramientas de desarrollo. Es open source y multi-plataforma.

Algunas de las principales características de este IDE genérico son: [4]

- Perspectivas, editores y vista: Eclipse utiliza un concepto de trabajo basado en ventanas y editores, relacionadas entre sí, y que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.
- Gestión de proyectos: Eclipse se basa en proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser código fuente, documentación, ficheros de configuración, ...
- Depurador de código: Entre las múltiples herramientas que proporciona Eclipse, está un potente depurador con su respectiva perspectiva para la depuración de código.
- Extensa colección de plugins: Al haber sido un estándar de facto durante mucho tiempo, existe una gran cantidad de plugins bien desarrollados por Eclipse o por terceros. Como mención, citaré el plugin JDT encargado del soporte del IDE al lenguaje Java.

6.1.2.2. Servidor de aplicaciones

Los servidores web son los encargados de recibir las peticiones referidas a páginas o elementos de la web, a través del protocolo http o https y de devolver el resultado de la petición, que suele ser un recurso alojado en el servidor.

Sin embargo, no tiene capacidad para descifrar el código de estos elementos. Esta tarea la dejan en manos del cliente que hizo la petición. Cuando trabajamos con elementos más complejos, entonces las páginas y otros elementos que almacena el servidor, contendrá código que debe ser procesado antes de entregarlo al cliente.

Un servidor de aplicaciones es el elemento software que proporciona servicios de aplicación a los clientes, gestionando la mayor parte de las funciones lógicas de negocio y de acceso a los datos de la aplicación.

Como consecuencia del éxito del lenguaje de programación Java, el término servidor de aplicaciones usualmente hace referencia a un servidor de aplicaciones Java EE o Java Platform Enterprise [8]. Se trata de una plataforma de programación para desarrollar y ejecutar software de aplicaciones en este lenguaje. Permite utilizar arquitectura de N capas distribuidas y se apoya ampliamente en componentes software modulares ejecutándose sobre un servidor de aplicaciones.

La especificación original J2EE fue desarrollada por la empresa Sun Microsystems [10]. La versión actual tiene el nombre de Java EE 8 y fue lanzada en 2014. Existen multitud de servidores de aplicaciones, siempre y cuando soporte la especificación establecida por JavaEE.

En base a las múltiples documentaciones analizadas en Internet sobre tipos de servidores de aplicaciones, he decidido utilizar Tomcat v8 como contenedor web basado en el lenguaje Java que actúa como motor de servlets y JSPs, convirtiéndose en la implementación por referencia.

Las principales características de la última versión estable son: [15]

- Autenticación de acceso básico.
- Negociación de credenciales.
- Soporte para el protocolo HTTPS.
- Posibilidad de alojamiento compartido.
- CGI o interfaz de entrada común.
- Servlets de Java.
- SSL (Secure Sockets Layer)

6.1.2.3. Arquitectura de la capa lógica

En la figura se muestra la arquitectura utilizada en la capa intermedia de mi aplicación, la cual se encuentra constituida por tres partes diferenciadas: controlador, dao, modelo y vista.

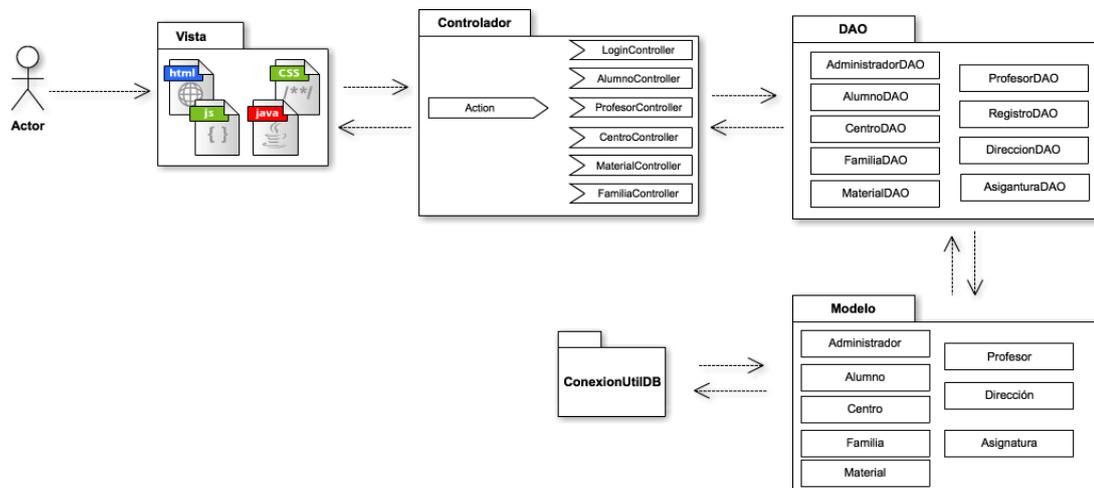


Figura 6.3: Arquitectura del sistema

Controlador

El controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Es decir, es el punto de entrada a la aplicación y quién se encarga de encauzar cualquier petición a la clase, que debe realizar la ejecución de la acción. El controlador en la web suele estar diseñado para un servlet cuyas funciones serían: [1]

1. Interceptar las peticiones HTTP del cliente.
2. Traducir cada petición en una operación específica de negocio a ser realizada.
3. Invocar o bien delegar en un manejador la operación.
4. Ayudar a seleccionar la siguiente vista a mostrar al cliente.
5. Devolver la vista al cliente.

En nuestro caso, existe un único controlador principal denominado **Action.java**, implementado en un servlet, el cual dependiendo de la operación a realizar traslada su ejecución a varios controladores secundarios: LoginController, MaterialController, CentroController, ProfesorController, FamiliaController y AlumnoController.

- Action.java es un servlet que hereda de la clase HttpServlet, que se encuentra en el paquete javax.servlet.http. Una parte importante de la declaración del servlet es la anotación @WebServlet (/Action), su función es registrar el servlet para todas las peticiones al servidor con la sintaxis http://url_servidor/eXtreBooks/Action.
- LoginController.java es un servlet que se encarga de las peticiones relacionadas con el inicio de sesión de los usuarios. Cuando a este servlet le llega una petición desde el Action, dependiendo del contenido de la petición ejecutará una acción u otra.
- MaterialController.java es el controlador de todas las operaciones relacionadas con los materiales curriculares.
- CentroController.java es el controlador de todas las operaciones relacionadas con los centros educativos.
- ProfesorController.java es el controlador de todas las operaciones relacionadas con los profesores.
- FamiliaController.java es el controlador de todas las operaciones relacionadas con los tutores de los alumnos.
- AlumnoController.java es el controlador de todas las operaciones relacionadas con los alumnos.

Las peticiones que pueden llegar de los distintos usuarios, solamente tiene un único punto de entrada, el servlet Action, de esta forma, todo el control de seguridad debe realizarse únicamente en este servlet, en lugar de realizarlos en todos.

En el algoritmo 6.1 se muestra parte del código del controlador Action.java.

Algoritmo 6.1 Código de ejemplo del controlador Action.java

```
01 @WebServlet("/action")
02 public class action extends HttpServlet {
03     String action = request.getParameter("action");
04     if (action==null || action.isEmpty() )
05     {
06         RequestDispatcher view = request.getRequestDispatcher("/back-
07         end/pages/contenedor.jsp"); view.forward(request, response); return;
08     }else if (action.contains("User") || (action.contains("Perfil"))){
09         request.getRequestDispatcher("/LoginController").forward(request, response);
10     }else if (action.contains("Material")){
11         request.getRequestDispatcher("/MaterialController").forward(request, response);
12     }else if (action.contains("Centro") || (action.contains("Curso") || (ac-
13         tion.contains("Asignatura")))){
14         request.getRequestDispatcher("/CentroController").forward(request, response);
15     }else if (action.contains("Profesor")){
16         request.getRequestDispatcher("/ProfesorController").forward(request, response);
17     }else if (action.contains("Familia")){
18         request.getRequestDispatcher("/FamiliaController").forward(request, response);
19     }else if (action.contains("Alumno")){
20         request.getRequestDispatcher("/AlumnoController").forward(request, response);
21     }
```

Se observa que cuando se recibe la petición, se comprueba el tipo de acción. Si coincide con alguna de las acciones descritas, se procede con su ejecución y en caso contrario se muestra un código de error. La amplia mayoría de veces estas acciones son conducentes a otras acciones secundarias controladas por otro controlador del sistema.

DAO o Data Access Object

El sistema eXtreBooks almacena y recoge datos de una base de datos. El objetivo fundamental a la hora de diseñar el sistema fue mantener la independencia entre el sistema y la base de datos, de cómo se accede a los datos o incluso de si hay o no base de datos detrás. La aplicación debe conseguir los datos o ser capaz de guardarlo en algún sitio, pero no tiene por qué saber de dónde los está sacando o dónde se guardan.

Para conseguir esta independencia, hay una forma bastante eficiente en el mundo JEE y de aplicaciones web, es lo que se conoce como **patrón DAO** (Data Access Object). [3]

Utilizamos el patrón DAO para:

- Abstractar y encapsular los accesos.
- Gestionar la conexiones a la fuente de datos.
- Obtener los datos almacenados.

En nuestro caso, existen quince patrones DAO: AdministradorDAO, AlumnoDAO, AsignaturaDAO, CentroDAO, ProfesorDAO, MaterialDAO, RegistroDAO, LoginDAO, FamiliaDAO, ... los cuales obtendrán los datos de la base de datos, los modelará según la capa de modelo y devolverá la información a la capa de presentación para su visualización.

La implementación de estas clases se ha realizado como cualquier otra clase Java. Cabe destacar que en el acceso a los datos se ha usado un **pool de conexiones** para aprovechar los flujos de conexión con la base de datos en lugar de usar una única conexión con JDBC (Java Database Connectivity).

En el algoritmo 6.2 puede verse parte del código de la clase MaterialDAO. En este caso se realizan operaciones con la información de los materiales didácticos.

Algoritmo 6.2 Código de ejemplo del patrón DAO

```

01 public class MaterialDAO {
02 private Connection connection;
03 public Statement st = null;
04 public ResultSet rs = null;
05 public Material_model getMaterialById(int id){
06 Material_model material=new Material_model();
07 try {
08 PreparedStatement preparedStatement = connection. prepareStatement("SELECT *
FROM Materiales WHERE IdMaterial = ?");
09 preparedStatement.setInt(1, id);
10 ResultSet rs = preparedStatement.executeQuery();
11 if (rs.next()) {
12 material.setIdMaterial(rs.getInt("IdMaterial"));
13 material.setTitulo(rs.getString("Titulo"));
14 material.setEditorial(rs.getString("Editorial"));
15 material.setIsbn(rs.getString("Isbn"));
16 material.setAutor(rs.getString("Autor"));
17 material.setPrecio(rs.getDouble("Precio"));
18 material.setUrl(rs.getString("Url"));
19 material.setEmpresa(rs.getString("Empresa"));
20 material.setDescripcion(rs.getString("Descripcion"));
21 material.setFoto(rs.getString("Foto"));
22 material.setId_docente(rs.getString("id_docente"));
23 material.setTipo_material(rs.getInt("id_tipo_material"));
24 material.setId_asignatura(rs.getInt("id_asignatura"));
25 }
26 } catch (SQLException e)
27 { e.printStackTrace(); }
28 return material;
29 }

```

Modelo

El modelo es la representación de la información, con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso. Envía a la vista aquella parte de la información que en cada momento se le solicita, para que sea mostrada.

En nuestro caso, una vez obtenido los datos de la capa de datos, se representan en una clase para su tratamiento en la capa de presentación. Concretamente, se han realizados cator-

ce modelos de datos en las siguientes clases: Profesor, Material, Alumno, Familia, Registro, Dirección, Asignatura, Centro, ...

En el algoritmo 6.3 se muestra parte del código de la clase modelo Material.

6.1.2.4. Diagrama de secuencia

Con el objetivo de facilitar al lector la comprensión de la arquitectura de la capa lógica, se muestra en la figura 6.4, el diagrama de secuencia de listar los profesores de un determinado centro, cuando inicia sesión el administrador.

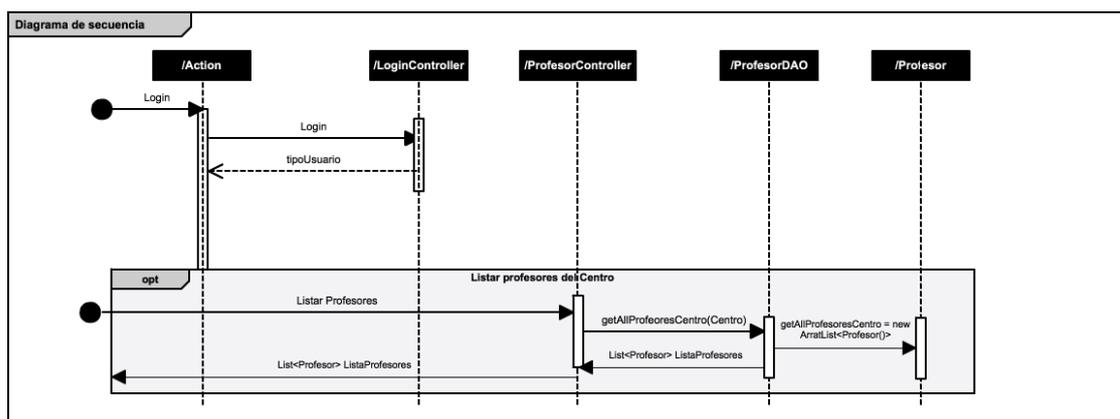


Figura 6.4: Diagrama de secuencia aplicación web

6.1.2.5. Particularidades de la implementación

En este apartado voy a resaltar algunas particularidades que he realizado en la implementación con el fin de aplicar optimizaciones al proyecto.

- Un archivo de propiedades Java, es un documento en el cual podemos almacenar propiedades de configuración de nuestra aplicación. En base a esto, se ha definido un fichero de propiedades denominado *consultas.properties*, donde se encuentran almacenadas todas las consultas que se van a utilizar en la aplicación. De este modo, si se necesita modificar el código de una consulta no es necesario compilar y empaquetar nuevamente la aplicación, sino que se recupera cada vez que se ejecuta la consulta.
- Cuando trabajamos con una base de datos, es posible que haya sentencias SQL que tengamos que ejecutar varias veces durante la sesión, aunque sea con distintos parámetros. Casi todas las bases de datos tienen previsto un mecanismo para que en estos casos, la ejecución de estas sentencias repetidas sea más rápida. Para ello, indicaremos previamente, el tipo de sentencia que vamos a usar, de forma que la base de datos la precompila y la guarda en condiciones de ser ejecutada inmediatamente, sin necesidad de analizar en cada caso. Esto es lo que se conoce como *prepareStatement*.
- Para la conexión con la base de datos se ha utilizado un pool de conexiones con el servidor Apache Tomcat desde la aplicación web. Usar un pool de conexiones es más eficiente que abrir una nueva conexión cada vez que sea necesario. Además, si fuese necesario hacer una migración de base de datos es mucho más rápido si solo tenemos que cambiar las

credenciales en el pool de conexiones, sin tener que tocar nada de código, pues toda la configuración se realiza en *context.xml*.

- En el diseño y desarrollo de la aplicación web se ha utilizado un *Responsive Web Design* o diseño web adaptable para poder adaptar la apariencia del sistema eXtreBooks al dispositivo que se esté utilizando para su visualización.

6.1.3. Capa de presentación

La capa de presentación puede suponer a priori un proceso fácil de diseñar e implementar. Sin embargo, unos de los principales quebraderos de cabeza con los que se encuentra un desarrollador a la hora de iniciar un nuevo proyecto web, es la creación de un *layout básico* sobre el que trabajar. [5]

Este layout debe tener como objetivo fundamental la adaptación de la interfaz web a los distintos navegadores, configuraciones o incluso dispositivos. Por ello, la adopción de un framework de apoyo puede ahorrarnos mucho tiempo y esfuerzo.

Un framework, es un marco de trabajo, un conjunto de bibliotecas orientadas a la reutilización de componentes de software para el desarrollo rápido de aplicaciones [6]. Con el fin de cubrir estas necesidades surgen múltiples frameworks de diseño de sitios y aplicaciones web, entre los que puedo destacar: 960gs, Bootstrap, Foundation v5.5, Skeleton v2.0.1, Pure, UI Kit, Gumby, ...Sin embargo, como toda tecnología presentan sus pros y contras en el uso de este tipo de herramientas.

Entre las principales ventajas: [6]

- Rapidez. El usar un framework te permitirá ir mucho más rápido en la creación de sitios web una vez sepas cómo funciona el framework.
- Comodidad. Al haber sido ya desarrolladas muchas de las funcionalidades que utilizan los sitios web, se reduce el tiempo de pensar cómo hacer las cosas.
- Componentes. Los frameworks robustos traen muchos componentes adicionales en forma de plugins de JavaScript, lo que te permite implementar soluciones un poco más avanzadas sin necesidad de desarrollarlas tú mismo.
- Soporte. Los frameworks conocidos tienen una amplia comunidad online detrás, por lo que será fácil encontrar respuesta a las dudas que se vayan planteando.

Entre las principales desventajas:

- Aprendizaje. Cada framework funciona de un modo distinto, con lo que hay que aprender a utilizarlo.
- Exceso de código. Un buen framework viene pensado para cubrir las necesidades de la mayoría de desarrolladores, pero esto no significa que en cada proyecto se vayan a desarrollar todas ellas.
- Semántica. Los frameworks utilizan un sistema de grid con una serie de clases extra que permiten identificar el número de columnas a ocupar. Los nombres de estas clases no son semánticos y, por tanto, el código del framework está más pensado para funcionalidad y rapidez que para semántica.

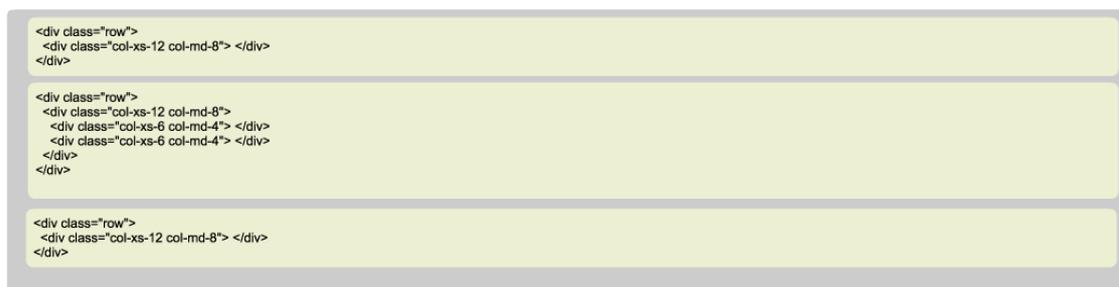


Figura 6.5: Esquema de rejillas del servicio web

- Aspecto final. Al traer muchas cosas ya desarrolladas “de fábrica”, el aspecto puede ser bastante similar de un sitio a otro, si no se personaliza el CSS del framework.

En base a esta información, procedí a la creación del servicio web utilizando como framework de diseño **Bootstrap v3.0**, tanto para el desarrollo del front-end como del back-end del sistema eXtreBooks. Bootstrap es de código abierto y disponible en GitHub, soportado en la mayoría de navegadores modernos, tanto de escritorio como dispositivos móviles.

Para usar Bootstrap en nuestro proyecto, lo primero que tuve que hacer es descargar los ficheros compilados de css, javascript e imágenes e incluirlos en el proyecto. En el algoritmo 6.5 se muestra el esqueleto para su incorporación en el proyecto.

Es importante saber que se necesita el framework JQuery para que los plugins javascript funciones. Estos componentes añaden o extienden funcionalidades a la interfaz de usuario como tooltips, cuadro de diálogos, botones, alertas, etc...

Sistema de cuadrículas

Bootstrap incluye un sistema de cuadrículas o rejillas (grid) responsive (diseño web adaptado a todas las pantalla de dispositivos). En concreto, dispone de 12 columnas con un ancho estándar de 940 pixeles [2]. Para tratar estas cuadrículas, existen clases predefinidas, para una fácil disposición de los contenidos en la página web.

En el proyecto eXtreBooks, la organización de estas cuadrículas o rejillas se muestra en la figura 6.5. Este diseño en rejilla se realiza mediante filas y columnas, donde se colocan los contenidos y siempre utilizando la clase CSS *.container-fluid* para que los elementos tengan una anchura variable.

Para la implementación del sitio web se ha utilizado principalmente HTML5, CSS3, JSP, Javascript, JQuery y Ajax, tecnologías que definí en el capítulo 2 dentro del punto 2.3.1 de este proyecto. El resto de componentes utilizados en el diseño del sistema, como botones, formularios, tablas, ventanas modales, ... son los predefinidos por Bootstrap, listos para utilizarse y que cubren las necesidades más habituales de los diseños actuales para la web.

Para poder acceder al servicio web, es necesario crear un manualmente en la base de datos un usuario administrador. Con el usuario administrador ya creado podremos proceder a la creación de usuarios, centros educativos,... así como resto de elementos de información del sistema desde la aplicación web.

6.2. Implementación de la seguridad del sistema

En el capítulo de diseño especificué que la seguridad es una parte fundamental de mi aplicación ya que el sistema utiliza datos sensibles que deben ser protegidos. Se habló fundamentalmente de implementar métodos de seguridad en la base de datos.

6.2.1. Seguridad en la base de datos

La información confidencial, como son la contraseña del usuario, número de identificación personal, número registro personal,... se almacenarán en la base de datos cifrados con AES. El uso de la encriptación AES es sencillo y se realiza con dos funciones AES_ENCRYPT y AES_DECRYPT.

Para realizar operaciones con los datos cifrados, el uso de la función AES_ENCRYPT es más que suficiente, ya que recibe dos parámetros: el dato que se quiere guardar (String) y la clave (String) que se utilizará para cifrar. Para poder guardar estas cadenas cifradas, es necesario que el tipo de datos en la base de datos de la columna sea varbinary.

Para poder descifrar un valor se usa la función AES_DECRYPT, recibe como parámetros la cadena cifrada y la clave que se utilizó para cifrar.

Para insertar una clave cifrada utilizaremos la siguiente sentencia 6.6

Algoritmo 6.6 Sentencia SQL para insertar una clave cifrada

```
INSERT INTO 'Registro' ('email','password',tipo) VALUES (email_usuario,
AES_ENCRYPT(password_cifrada, 'clave')), tipo_usuario)
```

Para descifrar una clave de la base de datos usaremos la sentencia 6.7

Algoritmo 6.7 Sentencia SQL para descifrar una clave cifrada

```
SELECT AES_DECRYPT(password, 'clave') as pass FROM 'Registro' WHERE email LIKE
'email_usuario'
```

La siguiente medida de seguridad implantada en la base de datos, ha sido la autenticación en el acceso al sistema de base de datos, por lo tanto, no podrá acceder a la base una autoridad que no disponga de contraseña y acceso a MySQL.

Algoritmo 6.3 Código ejemplo del modelo Material

```
01 public class Material_model {
02 private int IdMaterial;
03 private String Titulo;
04 private String Editorial;
05 private String Isbn;
06 private String Autor;
07 private Double Precio;
08 private String Url;
09 private String Empresa;
10 private String Descripcion;
11 private String Foto;
12 private String id_docente;
13 private int tipo_material;
14 private int id_asignatura;
15 public Material_model(){}
16 public Material_model( parametrizados){}
17 public int getIdMaterial() { return IdMaterial; }
18 public void setIdMaterial(int idMaterial) { IdMaterial = idMaterial; }
19 public String getTitulo() { return Titulo; }
20 public void setTitulo(String titulo) { Titulo = titulo; }
21 public String getEditorial() { return Editorial; }
22 public void setEditorial(String editorial) { Editorial = editorial; }
23 public String getIsbn() { return Isbn; }
24 public void setIsbn(String isbn) { Isbn = isbn; }
25 public String getAutor() { return Autor; }
26 public void setAutor(String autor) { Autor = autor; }
27 public Double getPrecio() { return Precio; }
28 public void setPrecio(Double precio) { Precio = precio; }
29 public String getUrl() { return Url; }
30 public void setUrl(String url) { Url = url; }
31 public String getEmpresa() { return Empresa; }
32 public void setEmpresa(String empresa) { Empresa = empresa; }
33 public String getDescripcion() { return Descripcion; }
34 public void setDescripcion(String descripcion) { Descripcion = descripcion; }
35 public String getFoto() { return Foto; }
36 public void setFoto(String foto) { Foto = foto; }
37 public String getId_docente() { return id_docente; }
38 public void setId_docente(String id_docente) { this.id_docente = id_docente; }
39 public int getTipo_material() { return tipo_material; }
40 public void setTipo_material(int tipo_material) { this.tipo_material = tipo_material; }
41 public int getId_asignatura() { return id_asignatura; }
42 public void setId_asignatura(int id_asignatura) { this.id_asignatura = id_asignatura; }
```

Algoritmo 6.4 Fichero context.xml

```

01 <Context>
02     <Resource      name="jdbc/ConexionMySQL"      auth="Container"      ty-
pe="javax.sql.DataSource"
03 maxActive="20" maxIdle="10" maxWait="5000"
04 username="usuarioBD" password="passwordBD"
05 driverClassName="com.mysql.jdbc.Driver"
06 url="jdbc:mysql://localhost:3306/BaseDeDatos"/>
07 </Context>

```

Los parámetros de configuración son los siguientes:

- maxActive: Indica el número máximo de conexiones que pueden estar abiertas al mismo tiempo.
 - maxIdle: El número máximo de conexiones inactivas que permanecerán abiertas, si el número de conexiones inactivas es muy bajo puede darse el caso de que las conexiones se cierren, porque se llega al máximo de conexiones inactivas y se vuelvan a abrir inmediatamente, reduciendo la eficiencia, ya que se perdería la ventaja del uso de pool de conexiones en cuanto a que no hay que abrir una conexión cada vez que es necesario.
 - maxWait: Es el tiempo máximo (en ms) que se esperará a que haya una conexión disponible (inactiva), si se supera este tiempo se lanza una excepción.
 - username: Usuario de la base de datos.
 - password: Password para el usuario introducido en username.
 - driverClassName: Nombre del driver JDBC para conectar con la base de datos.
 - url: URL de la base de datos a la que nos queremos conectar.
-

Algoritmo 6.5 Esquema eXtreBooks template

```

01 <!DOCTYPE html>
02 <html>
03 <head>
04 <title>eXtreBooks Template</title>
05 <meta name="viewport" content="width=device-width, initial-scale=1.0">
06 <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
07 <link href="css/bootstrap-responsive.css" rel="stylesheet">
08 </head>
09 <body>
10 <h1>Bienvenidos a eXtreBooks!</h1>
11 <script src="http://code.jquery.com/jquery.js"></script>
12 <script src="js/bootstrap.min.js"></script>
13 </body>
14 </html>

```

Capítulo 7

Manual de Usuario

Contenidos

7.1. Aplicación Web	95
7.1.1. Usuario Administrador	95
7.1.2. Usuario Equipo directivo	96
7.1.3. Usuario Profesor	98
7.1.4. Usuario Alumno	99

En este capítulo se explica cómo se utiliza el sistema eXtreBooks para la gestión de materiales curriculares en los centros educativos.

7.1. Aplicación Web

Libros, revistas, material fungible escolar, ...la mayor parte de los profesores y familias apreciamos esa colaboración necesaria que debería existir en la gestión de materiales didácticos utilizados diariamente por los alumnos. Sin embargo, no siempre es posible garantizar que dicha respuesta se dará en el tiempo y forma deseable: esa es la semilla de la que parte eXtreBooks.

Desde eXtreBooks se quiere dar una respuesta inmediata y adecuada ante cualquier tipo de eventualidad posible, ya sea en la gestión de tareas, recopilación de material visual, ... ofreciendo una variada gama de soluciones para tratar de maximizar esa relación de tres vértices, entre centros educativos, familias y profesorado.

El sistema eXtreBooks dispone de una interfaz web que gestiona toda la información relacionada con los profesores, aulas, asignaturas, materiales, .. Además, permite controlar toda esta información dependiendo del perfil del usuario que se haya logueado. El sistema reconoce cinco tipos de usuarios, el usuario administrador, el usuario equipo directivo, el usuario profesor, el usuario alumno y el usuario familia.

7.1.1. Usuario Administrador

El usuario administrador será el encargado de todas las funciones de gestión y administración del sistema eXtreBooks. En concreto, las acciones disponibles son las siguientes, tal y como se muestra en la figura 7.1.

- **Centros.** Esta opción permite llevar el control de los centros educativos donde se implanta el sistema. Al seleccionar esta acción, se despliega un submenú con tres opciones: nuevo centro, listar centros y modificar centros.

La operación de *nuevo centro*, permite a través de un simple formulario, dar de alta un centro educativo independientemente del nivel que tenga. La operación de *listar centro*, permite mostrar tanto la información propia del centro, como la información académica del mismo. La operación de *modificación/eliminación de centro*, permite modificar los datos de un centro así como su eliminación del sistema.

- **Profesorado.** Esta opción permite llevar el control de los profesores registrados en el sistema. Al seleccionar esta acción, se despliega un submenú con tres opciones: alta profesor, listar profesores y asignar dirección.

La operación de *alta de profesor*, permite con un simple formulario, dar de alta a un profesor en el sistema y asignarle un determinado centro. La operación *listar profesores* mostrará por pantalla toda la información de los profesores registrados en el sistema. La operación *asignar dirección*, establece el equipo directivo de un determinado centro, que serán los encargados de gestionar asignaturas, alumnos, profesorado, materiales del centro asignado.

- **Materiales.** Esta opción permite llevar el control de las distintas categorías de materiales a incorporar a los centros educativos. Al seleccionar esta acción, se despliega un submenú con dos opciones: nueva categoría y listar categoría.

La operación de *nueva categoría*, permite dar de alta un nuevo tipo de material a ser utilizado por el centro. La operación *listar categoría*, muestra por pantalla los tipos de materiales utilizados por los centros registrados en el sistema.



Figura 7.1: Página principal del usuario administrador

7.1.2. Usuario Equipo directivo

El usuario equipo directivo será el encargado de todas las funciones de gestión y administración del centro educativo al que fue asignado por parte del administrador del sistema. En concreto, las acciones disponibles son las siguientes, tal y como se muestran en las figuras 7.2 y 7.3.

- Cursos académicos. Esta opción lleva el control de los cursos implantados en el centro. Al seleccionar esta acción, se despliega un submenú con dos opciones: nuevo curso y actualizar curso.

La operación *nuevo curso*, permite con un simple formulario dar de alta un nuevo curso dentro del centro, especificando el número y el grupo. La operación *actualizar curso*, permite modificar/eliminar la información de un curso implantado en el centro, a través de una ventana emergente.

- Profesorado. Esta opción lleva el control de los profesores que dan clases en el centro educativo. Al seleccionar esta acción, se despliega un submenú con tres opciones: nuevo profesor, listar profesores y actualizar profesor.

La opción *nuevo profesor*, dará de alta un profesor en el centro a través de un simple formulario. La operación *listar profesor*, mostrará tanto los datos del empleado como los datos educativos de dicho profesor. La operación *actualizar profesor*, permitirá modificar los datos del profesor así como su eliminación del centro educativo.

- Asignaturas. Esta opción lleva el control de las asignaturas que se imparten en el centro educativo. Al seleccionar esta acción, se despliega un submenú con tres opciones: nueva asignatura, actualizar asignatura o listar asignaturas.

La opción *nueva asignatura*, dará de alta una nueva asignatura donde se especificará el departamento al que pertenece, el curso y profesor encargado de dicha asignatura. La operación *actualizar asignatura*, va a permitir modificar los datos de una determinada asignatura así como modificar el profesor o curso asignado. La operación *listar asignatura*, no solo mostrará su información sino también los materiales curriculares asignados a esa asignatura por parte de los profesores encargados.

- Materiales. Esta opción lleva el control de los materiales asignados a una determinada asignatura. Al seleccionar esta acción, se despliega un submenú con tres opciones: nueva categoría, nuevo material y actualizar material.

La operación *nueva categoría*, permite establecer un nuevo tipo material para ser utilizado en el centro. La operación *nuevo material*, establece un nuevo material curricular asignado a una determinada asignatura y gestionado por un determinado profesor. La operación *actualizar material*, muestra un listado de todos los materiales utilizados por los profesores en las asignaturas que dan clases.

- Alumnado. Esta opción lleva el control de los alumnos matriculados en el centro educativo y asignado a un curso. Al seleccionar esta acción, se despliega un submenú con dos opciones: nuevo alumno y listar alumnos.

La operación *nuevo alumno*, permite dar de alta un alumno con un simple formulario donde especificaremos el curso asignado y la familia responsable de ese alumno. La operación *listar alumno*, muestra tanto la información personal de dicho alumno como su información académica.

- Familia. Esta opción lleva el control de las familias donde sus hijos se encuentran matriculados en un centro educativo. Al seleccionar esta acción, se despliega un submenú con una sola opción: nueva familia.

La operación *nueva familia*, permite dar de alta al tutor/a responsable del alumno que se encuentra matriculado en el centro.

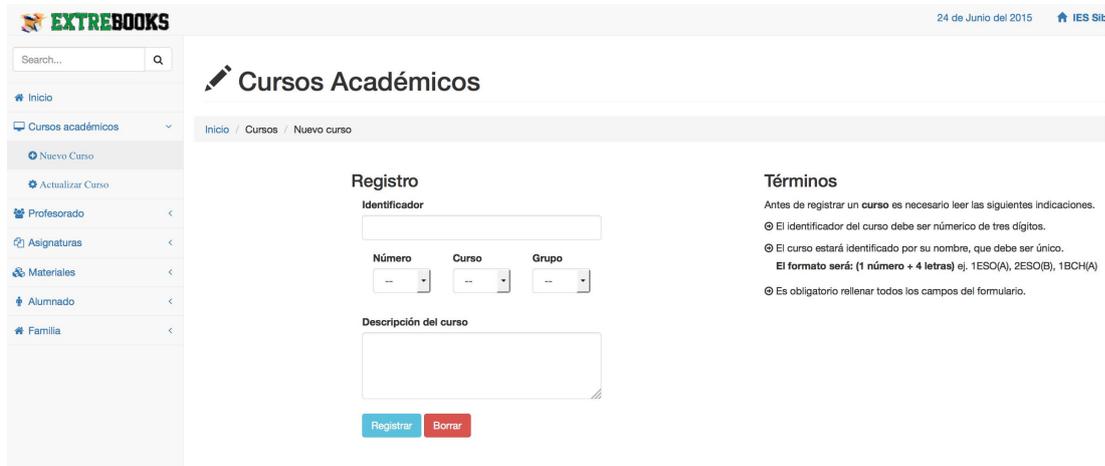


Figura 7.2: Página principal del usuario equipo directivo

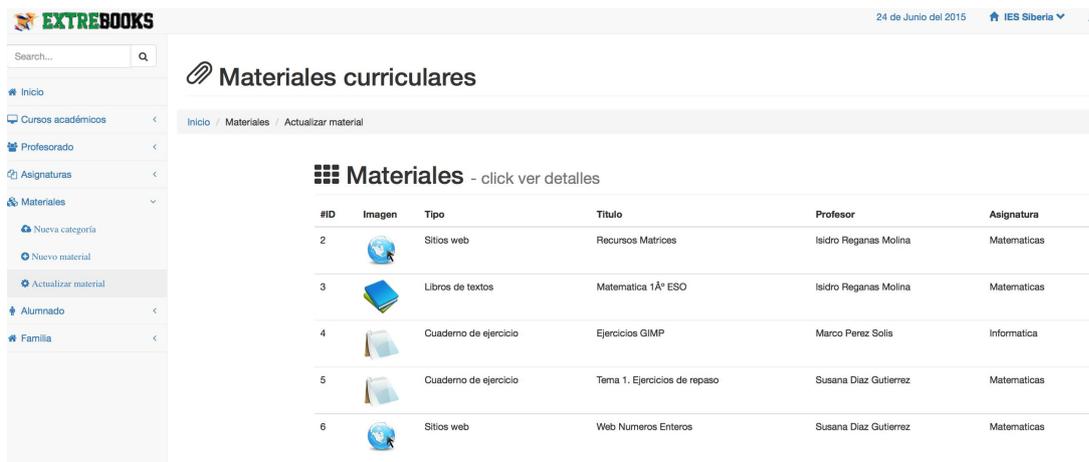


Figura 7.3: Página principal del usuario equipo directivo

7.1.3. Usuario Profesor

El usuario profesor, será el encargo de todas las funciones de gestión de los cursos y asignaturas, asignados por el equipo directivo del centro al que pertenece y de la administración de los materiales, que aportará a cada una de las asignaturas impartidas. En concreto, las acciones disponibles son las siguientes, tal y como se muestran en la figura 7.4 .

- Cursos académicos. Esta opción permite visualizar información detallada de los cursos asignado al profesor, mostrando alumnos matriculados, asignaturas impartidas así como el resto de profesores que imparten asignaturas en ese curso.
- Asignaturas. Esta opción permite visualizar información detallada de las asignaturas

que imparte el profesor, mostrando información de las características de las asignaturas y alumnos matriculados.

- **Materiales.** Esta opción lleva el control de los materiales que el profesor asigna a una asignatura. Al seleccionar esta acción, se despliega un submenú con tres opciones: asignar material, listar materiales y validar material.

La operación *asignar material*, permite determinar los materiales que una asignatura llevará incorporada. La operación *listar materiales*, muestra por pantalla todos los materiales asignados por el profesor, clasificados por cursos y asignaturas. La operación *validar material*, permite aprobar un material recomendado por un alumno en una asignatura.

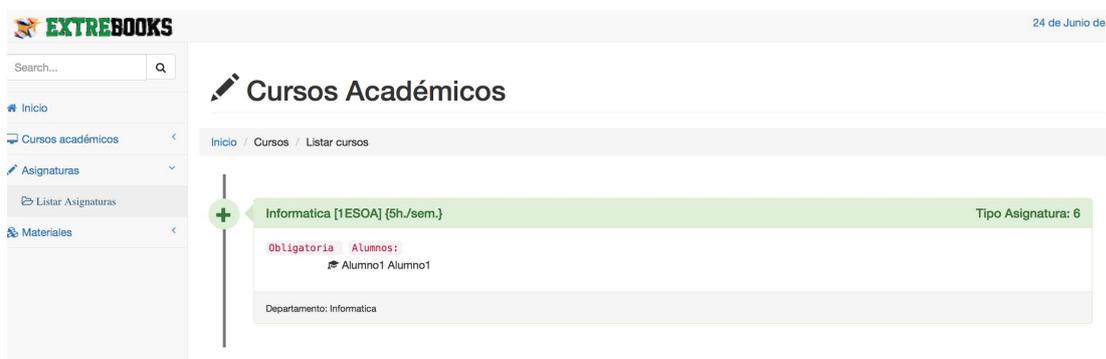


Figura 7.4: Página principal del usuario profesor

7.1.4. Usuario Alumno

El usuario alumno sólo podrá acceder a sus datos, podrá consultar las asignaturas donde se encuentra matriculado así como los materiales asignados a dichas asignaturas. En concreto, las acciones disponibles son las siguientes, tal y como se muestran en la figura 7.5.

- **Asignaturas.** Esta opción permite visualizar la información de las diversas asignaturas donde el alumno se encuentra matriculado, así como información educativa de dichas asignaturas.
- **Materiales.** Esta opción lleva el control de los materiales que los profesores asignan a las asignaturas donde el alumno está matriculado. Al seleccionar esta acción, se despliega un submenú con dos opciones: listar materiales y recomendar material.

La operación *listar materiales* muestra, por pantalla los diversos tipos de materiales que cada asignatura tiene asignada. La operación *recomendar material*, permite que un alumno pueda recomendar algún tipo de material a un profesor en una determinada asignatura.



Figura 7.5: Página principal del usuario alumno

Capítulo 8

Conclusiones y Trabajos Futuros

Contenidos

8.1. Conclusiones	101
8.2. Trabajos futuros	101
8.3. Planificación estimada vs Planificación Real	103

En este último capítulo se resumiré las principales conclusiones que a nivel global se pueden extraer de todo el trabajo realizado así como los trabajos futuros que pueden realizarse.

8.1. Conclusiones

La educación demanda, hoy más que nunca, una mejora de las relaciones entre el centro educativo, las familias y el profesorado, dando lugar a una colaboración estrecha y animosa. Pero es difícil realizar en la práctica, puesto que existen muchos factores contrapuestos que dificultan este apoyo y colaboración mutua. Cuando hablo de participación, no solo se refiere a las citaciones de las entrevistas del tutor/a, sino a las actividades y materiales didácticos que se propagan entre las diversas asignaturas, permitiendo que exista una relación de confianza y ayuda entre los padres y los profesores/as.

Esta relación se convierte hoy en día en un pilar muy fuerte para la eficacia y respuestas positivas del desarrollo educativo y personal del alumno/a. Para tratar este tema, tras finalizar este trabajo fin de grado se ha desarrollado una aplicación vía web que permite gestionar y administrar los materiales curriculares de un centro educativo.

De este modo, es posible comprobar el punto en el que está monitorizada cualquier asignatura en relación a los materiales didácticos, extrayendo como principal conclusión que, a pesar de estas brechas de desinformación, se puede dar una cooperación adecuada, dando lugar a nuevos patrones de relaciones entre familias y centros educativos.

8.2. Trabajos futuros

Desde la amplitud del proyecto, existen múltiples enfoques posibles de trabajo futuro sobre él:

Cliente móvil

En la actualidad, los teléfonos móviles se están convirtiendo en herramientas multimedia de calidad. Los alumnos deben establecer una relación directa del mundo que les rodea y la clase a través de una herramienta cercana y controlable por ellos mismo, sin ayuda directa del profesor. Por ello, se hace necesario desarrollar el actor móvil de forma que pueda gestionar la información de eXtreBooks de una manera rápida y sencilla. Esta aplicación móvil desarrollada para Android, será muy intuitiva, tal y como la hemos definido en la parte de diseño, donde se podrá controlar la información de las asignaturas, profesores y materiales asignados a cada alumno. Del mismo modo, el tutor/a (familia) del alumno podrá controlar dicha información.

Mejorar la seguridad del sistema

En el mercado, podemos encontrar múltiples herramientas que permiten añadir mayor seguridad a una base de datos como MySQL, tal es el caso de MyDiamo. Esta herramienta es una solución de cifrado que se integra a la perfección con MySQL, donde las funcionalidades de cifrado pueden establecerse a nivel de columnas o tablas completas. Además, es posible implementar el control de acceso, gestión de privilegios de cifrado/descifrado y funciones de auditoría de sus mecanismos de seguridad.

También es conveniente añadir un mayor nivel de seguridad a las comunicaciones, usando para ello el protocolo HTTPS o el uso de certificados electrónicos.

Mayor integración en la estructura educativa

El sistema debe ser implementado para que sea fácilmente integrable con otras plataformas o tecnologías web usadas en la actualidad por los centros educativos. Así que se propone una posible ampliación del proyecto en relación a la creación de un grupo de trabajo a nivel intercentro. Además, como trabajo futuro se podría implementar una mayor compatibilidad con los centros educativos de infantil y primaria, así como niveles educativos inferiores (guardería), ya que en estos centros, las asignaturas, los profesores, la carga lectiva difieren en relación a los centros de enseñanza secundaria.

También enfocar el sistema como un mecanismo de aviso, alertando mediante SMS o correos electrónicos de los futuros materiales necesarios en el aula, dando una mayor implicación a las familias.

Nuevas funcionalidades

- Sería interesante, que el sistema eXtreBooks, tanto en su servicio web como en su servicio móvil, generase mecanismos de estadísticas y gráficas sobre la utilización de los diversos tipos de herramientas por asignatura o por profesor. De esta manera, habría un seguimiento y control de aquellos materiales más demandados por el centro.
- Podría ser interesante, mejorar las relaciones intercentros en la generación de materiales. Se propone mejorar el sistema, para crear un espacio común de materiales, donde los profesores puedan intercambiar materiales didácticos y enriquecer las asignaturas con nuevas aportaciones curriculares.
- Disponer de un sistema de compra online asociadas a empresas del sector, de tal manera que el sistema facilite la adquisición de aquellos materiales que el alumno precise en las asignaturas.

diagrama**Tarea**

Nombre	Fecha de inicio	Fecha de fin
Requisitos	2/02/15	25/02/15
Toma	2/02/15	2/02/15
Casos de uso	3/02/15	16/02/15
Documentación	17/02/15	25/02/15
Análisis	26/02/15	20/03/15
Entorno Tecnológico	26/02/15	26/02/15
Escenarios	27/02/15	2/03/15
Modelo de datos	3/03/15	9/03/15
Interfaces Usuario	10/03/15	13/03/15
Documentación	16/03/15	20/03/15
Diseño	23/03/15	17/04/15
Arquitectura	23/03/15	27/03/15
Clases	30/03/15	1/04/15
Presentación	2/04/15	8/04/15
Documentación	9/04/15	17/04/15
Desarrollo	20/04/15	30/06/15
Diseño Web	20/04/15	15/05/15
Programación	18/05/15	19/06/15
Documentación	22/06/15	30/06/15
Implementación	1/07/15	3/07/15
Pruebas	1/07/15	2/07/15
Documentación	2/07/15	5/07/15

Tabla 8.1: Planificación real

8.3. Planificación estimada vs Planificación Real

En el cuadro 8.1 se puede visualizar la estimación de tiempo real que se ha necesitado para la realización del proyecto, en un diagrama de Gantt. Se puede observar, que la planificación real es ligeramente superior a la planificación estimada, debido a factores de carga de trabajo. Se había estimado, una dedicación al principio del proyecto que rondaba las 2 o 3 horas diarias dependiendo del día, siendo posteriormente las jornadas más largas llegando a 8 horas. Además, la labor de desarrollo del proyecto ha sido compaginada con mi labor profesional de trabajo.

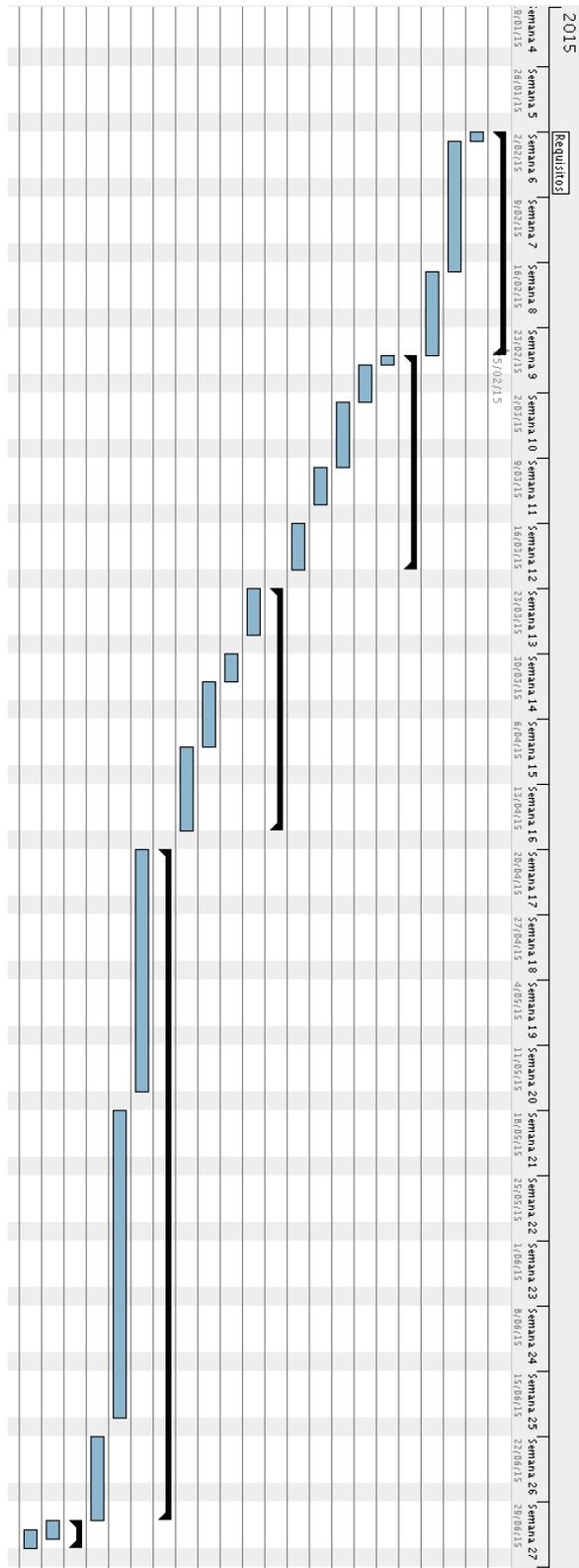


Figura 8.1: Diagrama de Gantt

Bibliografía

- [1] Aprenda servlets de java como si estuviera en primero. tecnun [online] available <http://www4.tecnun.es/asignaturas/informat1/ayudainf/aprendainf/jvaservlets/servlets.pdf>.
- [2] Bootstrap v3.0. libroweb [online] available <http://www.libroweb.es>.
- [3] Data access object - wikipedia,available https://es.wikipedia.org/wiki/data_access_object.
- [4] Eclipse ide. genbeta [online].. available <http://www.genbetadev.com/herramientas/eclipse-ide>.
- [5] Framework layout. genbeta [online] available <https://www.genbetadev.com/record/1712>.
- [6] Framework. wikipedia [online] available <https://es.wikipedia.org/wiki/framework>.
- [7] Historia css. libroweb tutorial. [online]. available <http://www.libroweb.es>.
- [8] Jajava. wikipedia. [online]. available http://es.wikipedia.org/wiki/java_ee.
- [9] Javascript. opencertiac [online]. available <http://www.opencertiac.org/javascript.html>.
- [10] Javjava. oracle [online]. available <http://docs.oracle.com/javase/7/api>.
- [11] Modelo vista controlador. tú código fuente. [online]. available <http://www.tucodigofuente.com>.
- [12] Mysql. wikipedia. [online]. available <https://es.wikipedia.org/wiki/mysql>.
- [13] Seguridad de la información. wikipedia. [online]. available https://es.wikipedia.org/wiki/seguridad_de_la_informaci3n.
- [14] Servicios web. myslide.es [online]. available http://download.myslide.es/uploads/check_up03/242015/55
- [15] Tutorial javaee-javahispao [online]. available <http://www.javahispano.org/storage/contenidos/javaee.pdf>.
- [16] ¿cómo funciona ssl? innovation is freedmon. [online]. available <https://www.ovh.es/ssl/funcionamiento-ssl.xml>.
- [17] ¿qué es wikisaber.es? wikisaber.es [online]. available <http://www.wikisaber.es>.
- [18] Cajilima Alvarado and José Ricardo. Desarrollo de una aplicación, para dispositivos móviles que permita administrar pedidos y controlar rutas de los vendedores, aplicada a la empresa: Almacenes juan eljuri cía. ltda. división perfumería. 2015.

- [19] Luis Álvarez Álvarez. Diseño y desarrollo de una aplicación web para gestión docente. 2009.
- [20] EUGENIA BAHIT. El paradigma de la programación orientada a objetos en php con el patrón arquitectónico mvc.[en línea], 2011.
- [21] O Cepeda, Daida González Salamanca, Manuel Area Moreira, et al. Un análisis de las actividades didácticas con tic en aulas de educación secundaria. 2011.
- [22] Ramez Elmasri, Shamkant B Navathe, Verónica Canivell Castillo, Beatriz Galán Espiga, and Gloria Zaballa Pérez. *Fundamentos de sistemas de bases de datos*. Addison-Wesley, 2002.
- [23] Vicenç Fernández Alarcón. Desarrollo de sistemas de información: Una metodología basada en el modelado. 2006.
- [24] Francisco García García. Contenidos educativos digitales: Construyendo la sociedad del conocimiento. *Red digital: Revista de Tecnologías de la Información y Comunicación Educativas*, (6):1, 2005.
- [25] Ivar Jacobson, Grady Booch, James Rumbaugh, James Rumbaugh, and Grady Booch. *The unified software development process*, volume 1. Addison-wesley Reading, 1999.
- [26] Jamie Jaworski and Paul J Perrone. *Seguridad en Java*. Prentice Hall, 2001.
- [27] Rafael Navarro Marset. Rest vs web services, 2006.
- [28] Juan José Miranda Miranda. *Gestión de proyectos: identificación, formulación, evaluación financiera-económica-social-ambiental*. MMEditores, 2005.
- [29] Roger S Pressman. *Ingeniería del Software: Un enfoque práctico*. Mikel Angoar, 1997.
- [30] Pamela Mollinedo Rosado. Html. *ACALAN Revista de la Universidad Autónoma del Carmen*, 1(1), 2012.
- [31] Antonio Sarasa and JM Canabal. Agrega: un proyecto de software libre de la web 2.0 de la administración pública. *Agrega Documentos*, 2012.
- [32] María Helena Trejos Arroyave and Diego Fernando Zamora Cardona. Criterios de evaluación de plataformas de desarrollo de aplicaciones empresariales para ambientes web. 2012.

Capítulo 9

Anexo I. Importación de la base de datos

A continuación se muestra el código SQL mediante el cual crear la base de datos necesaria para el correcto funcionamiento del sistema eXtreBooks

```
– Schema extrebooks – _____
CREATE SCHEMA IF NOT EXISTS ‘extrebooks‘ DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ; USE ‘extrebooks‘ ;
_____
– Table ‘extrebooks’.‘Registro‘ – _____
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Registro‘ (
‘email‘ VARCHAR(45) NOT NULL,
‘password‘ VARBINARY NOT NULL,
‘tipo_usuario‘ INT NULL,
PRIMARY KEY (‘email‘))
ENGINE = InnoDB;
_____
– Table ‘extrebooks’.‘Administrador‘ – _____
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Administrador‘ (
‘IDADMINISTRADOR‘ INT NOT NULL AUTO_INCREMENT,
‘nombre‘ VARCHAR(45) NOT NULL,
‘apellidos‘ VARCHAR(45) NOT NULL,
‘id_email‘ VARCHAR(45) NOT NULL,
PRIMARY KEY (‘IDADMINISTRADOR‘),
INDEX ‘fk_Administrador_Registro1_idx‘ (‘id_email‘ ASC),
CONSTRAINT ‘fk_Administrador_Registro1‘
FOREIGN KEY (‘id_email‘)
REFERENCES ‘extrebooks’.‘Registro‘ (‘email‘)
ON DELETE NO ACTION ON UPDATE NO ACTION)
ENGINE = InnoDB;
_____
– Table ‘extrebooks’.‘Centro‘ – _____
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Centro‘ (
```

```

‘CIF‘ VARCHAR(9) NOT NULL,
‘Nombre‘ VARCHAR(45) NOT NULL,
‘Direccion‘ VARCHAR(45) NULL,
‘Localidad‘ VARCHAR(45) NULL,
‘Provincia‘ VARCHAR(45) NULL,
‘CP‘ VARCHAR(5) NULL,
‘Fax‘ VARCHAR(9) NULL,
‘Tif‘ VARCHAR(9) NULL,
‘Email‘ VARCHAR(45) NOT NULL,
‘Url‘ VARCHAR(45) NULL,
‘Nivel‘ VARCHAR(45) NULL,
‘id_Administrador‘ INT NOT NULL,
PRIMARY KEY (‘CIF‘),
INDEX ‘fk_Centro_Administrador1_idx‘ (‘id_Administrador‘ ASC),
CONSTRAINT ‘fk_Centro_Administrador1‘
FOREIGN KEY (‘id_Administrador‘)
REFERENCES ‘extrebooks‘.‘Administrador‘ (‘IDADMINISTRADOR‘)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
COMMENT = ‘Administrador_idAdministrador’;

```

– **Table ‘extrebooks‘.‘Tipo_Materiales‘** –

```

CREATE TABLE IF NOT EXISTS ‘extrebooks‘.‘Tipo_Materiales‘ (
‘ID_TIPO‘ INT NOT NULL,
‘categoria‘ VARCHAR(45) NOT NULL,
‘descripción‘ VARCHAR(45) NOT NULL,
‘id_administrador‘ INT NOT NULL,
PRIMARY KEY (‘ID_TIPO‘),
INDEX ‘fk_Tipo_Materiales_Administrador1_idx‘ (‘id_administrador‘ ASC),
CONSTRAINT ‘fk_Tipo_Materiales_Administrador1‘
FOREIGN KEY (‘id_administrador‘)
REFERENCES ‘extrebooks‘.‘Administrador‘ (‘IDADMINISTRADOR‘)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

– **Table ‘extrebooks‘.‘Docente‘** –

```

CREATE TABLE IF NOT EXISTS ‘extrebooks‘.‘Docente‘ (
‘NRP‘ VARCHAR(10) NOT NULL,
‘DNI‘ VARCHAR(9) NOT NULL,
‘Nombre‘ VARCHAR(45) NOT NULL,
‘Apellidos‘ VARCHAR(45) NOT NULL,
‘Domicilio‘ VARCHAR(45) NULL,
‘Localidad‘ VARCHAR(45) NULL,
‘Provincia‘ VARCHAR(45) NULL,

```

```

'CP' VARCHAR(5) NULL,
'Tfno' VARCHAR(9) NULL,
'Cargo' VARCHAR(20) NOT NULL,
'Fecha alta' DATE NULL,
'Foto' VARCHAR(50) NULL,
'id_centro' VARCHAR(9) NOT NULL,
'id_email' VARCHAR(45) NOT NULL,
PRIMARY KEY ('NRP'),
INDEX 'fk_Docente_Centro1_idx' ('id_centro' ASC),
INDEX 'fk_Docente_Registro1_idx' ('id_email' ASC),
CONSTRAINT 'fk_Docente_Centro1'
FOREIGN KEY ('id_centro')
REFERENCES 'extrebooks'.'Centro' ('CIF')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Docente_Registro1'
FOREIGN KEY ('id_email')
REFERENCES 'extrebooks'.'Registro' ('email')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

– **Table 'extrebooks'.'Curso'** –

```

CREATE TABLE IF NOT EXISTS 'extrebooks'.'Curso' (
'IDCURSO' INT NOT NULL,
'nombre' VARCHAR(45) NOT NULL,
'descripcion' VARCHAR(45) NULL,
'id_centro' VARCHAR(9) NOT NULL,
PRIMARY KEY ('IDCURSO'),
INDEX 'fk_Curso_Centro1_idx' ('id_centro' ASC),
CONSTRAINT 'fk_Curso_Centro1'
FOREIGN KEY ('id_centro')
REFERENCES 'extrebooks'.'Centro' ('CIF')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

– **Table 'extrebooks'.'Asignatura'** –

```

CREATE TABLE IF NOT EXISTS 'extrebooks'.'Asignatura' (
'IDASIGNATURA' INT NOT NULL AUTO_INCREMENT,
'nombre' VARCHAR(45) NOT NULL,
'caracter' VARCHAR(45) NULL,
'creditos' VARCHAR(45) NULL,
'temporalidad' VARCHAR(45) NULL,
'departamento' VARCHAR(45) NULL,
'id_curso' INT NOT NULL,
'id_docente' VARCHAR(45) NOT NULL,

```

```

PRIMARY KEY ('IDASIGNATURA'),
INDEX 'fk_Asignatura_Curso1_idx' ('id_curso' ASC),
INDEX 'fk_Asignatura_Docente1_idx' ('id_docente' ASC),
CONSTRAINT 'fk_Asignatura_Curso1'
FOREIGN KEY ('id_curso')
REFERENCES 'extrebooks'.Curso ('IDCURSO')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Asignatura_Docente1'
FOREIGN KEY ('id_docente')
REFERENCES 'extrebooks'.Docente ('NRP')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

– **Table 'extrebooks'. 'Materiales'** –

```

CREATE TABLE IF NOT EXISTS 'extrebooks'. 'Materiales' (
'IDMATERIAL' INT NOT NULL AUTO_INCREMENT,
'titulo' VARCHAR(45) NULL, 'editorial' VARCHAR(45) NULL,
'isbn' VARCHAR(45) NULL,
'autor' VARCHAR(45) NULL,
'precio' DECIMAL(2) NULL,
'url' VARCHAR(45) NULL,
'empresa' VARCHAR(45) NULL,
'descripcion' VARCHAR(45) NULL,
'foto' VARCHAR(45) NULL,
'id_docente' VARCHAR(45) NOT NULL,
'id_tipo_material' INT NOT NULL,
'id_asignatura' INT NOT NULL,
PRIMARY KEY ('IDMATERIAL'),
INDEX 'fk_Materiales_Docente1_idx' ('id_docente' ASC),
INDEX 'fk_Materiales_Tipo_Materiales1_idx' ('id_tipo_material' ASC),
INDEX 'fk_Materiales_Asignatura1_idx' ('id_asignatura' ASC),
CONSTRAINT 'fk_Materiales_Docente1'
FOREIGN KEY ('id_docente')
REFERENCES 'extrebooks'.Docente ('NRP')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Materiales_Tipo_Materiales1'
FOREIGN KEY ('id_tipo_material')
REFERENCES 'extrebooks'.Tipo_Materiales ('ID_TIPO')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Materiales_Asignatura1'
FOREIGN KEY ('id_asignatura')
REFERENCES 'extrebooks'.Asignatura ('IDASIGNATURA')
ON DELETE NO ACTION

```

```
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
– Table ‘extrebooks’.‘Curso_has_Docente’ –
```

```
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Curso_has_Docente’ (
‘Curso_idCurso’ INT NOT NULL,
‘Docente_NRP’ VARCHAR(45) NOT NULL,
PRIMARY KEY (‘Curso_idCurso’, ‘Docente_NRP’),
INDEX ‘fk_Curso_has_Docente_Docente1_idx’ (‘Docente_NRP’ ASC),
INDEX ‘fk_Curso_has_Docente_Curso1_idx’ (‘Curso_idCurso’ ASC),
CONSTRAINT ‘fk_Curso_has_Docente_Curso1’
FOREIGN KEY (‘Curso_idCurso’)
REFERENCES ‘extrebooks’.‘Curso’ (‘IDCURSO’)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT ‘fk_Curso_has_Docente_Docente1’
FOREIGN KEY (‘Docente_NRP’)
REFERENCES ‘extrebooks’.‘Docente’ (‘NRP’)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
– Table ‘extrebooks’.‘Asignatura_has_Docente’ –
```

```
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Asignatura_has_Docente’ (
‘Asignatura_idAsignatura’ INT NOT NULL,
‘Docente_NRP’ VARCHAR(45) NOT NULL,
PRIMARY KEY (‘Asignatura_idAsignatura’, ‘Docente_NRP’),
INDEX ‘fk_Asignatura_has_Docente_Docente1_idx’ (‘Docente_NRP’ ASC),
INDEX ‘fk_Asignatura_has_Docente_Asignatura1_idx’ (‘Asignatura_idAsignatura’
ASC),
CONSTRAINT ‘fk_Asignatura_has_Docente_Asignatura1’
FOREIGN KEY (‘Asignatura_idAsignatura’)
REFERENCES ‘extrebooks’.‘Asignatura’ (‘IDASIGNATURA’)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT ‘fk_Asignatura_has_Docente_Docente1’
FOREIGN KEY (‘Docente_NRP’)
REFERENCES ‘extrebooks’.‘Docente’ (‘NRP’)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
– Table ‘extrebooks’.‘Familia’ –
```

```
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Familia’ (
‘DNI’ VARCHAR(9) NOT NULL,
```

```

'nombre' VARCHAR(45) NOT NULL,
'apellidos' VARCHAR(45) NOT NULL,
'tlf' VARCHAR(45) NULL,
'domicilio' VARCHAR(45) NULL,
'cp' VARCHAR(5) NULL,
'localidad' VARCHAR(45) NULL,
'provincia' VARCHAR(45) NULL,
'id_email' VARCHAR(45) NOT NULL, PRIMARY KEY ('DNI'),
INDEX 'fk_Familia_Registro1_idx' ('id_email' ASC),
CONSTRAINT 'fk_Familia_Registro1'
FOREIGN KEY ('id_email')
REFERENCES 'extrebooks'.'Registro' ('email')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Table 'extrebooks'.'Alumno' -----

```

CREATE TABLE IF NOT EXISTS 'extrebooks'.'Alumno' (
'NIA' INT NOT NULL,
'dni' VARCHAR(45) NOT NULL,
'nombre' VARCHAR(45) NOT NULL,
'apellidos' VARCHAR(45) NOT NULL,
'fecha_nacimiento' DATE NULL,
'domicilio' VARCHAR(45) NULL,
'localidad' VARCHAR(45) NULL,
'cp' VARCHAR(45) NULL,
'provincia' VARCHAR(45) NULL,
'tlf_urgencia' VARCHAR(45) NULL,
'fecha_alta' DATE NULL,
'foto' VARCHAR(50) NULL,
'id_curso' INT NOT NULL,
'id_padre' VARCHAR(9) NOT NULL,
'id_docente' VARCHAR(45) NOT NULL,
'id_email' VARCHAR(45) NOT NULL,
PRIMARY KEY ('NIA'),
INDEX 'fk_Alumno_Curso1_idx' ('id_curso' ASC),
INDEX 'fk_Alumno_Padres/Madres1_idx' ('id_padre' ASC),
INDEX 'fk_Alumno_Docente1_idx' ('id_docente' ASC),
INDEX 'fk_Alumno_Registro1_idx' ('id_email' ASC),
CONSTRAINT 'fk_Alumno_Curso1'
FOREIGN KEY ('id_curso')
REFERENCES 'extrebooks'.'Curso' ('IDCURSO')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Alumno_Padres/Madres1'
FOREIGN KEY ('id_padre')
REFERENCES 'extrebooks'.'Familia' ('DNI')

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Alumno_Docente1'
FOREIGN KEY ('id_docente')
REFERENCES 'extrebooks'.Docente ('NRP')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Alumno_Registro1'
FOREIGN KEY ('id_email')
REFERENCES 'extrebooks'.Registro ('email')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

– **Table 'extrebooks'.Docente_has_Curso'** –

```

CREATE TABLE IF NOT EXISTS 'extrebooks'.Docente_has_Curso' (
'id_docente' VARCHAR(45) NOT NULL,
'id_curso' INT NOT NULL,
PRIMARY KEY ('id_docente', 'id_curso'),
INDEX 'fk_Docente_has_Curso_Curso1_idx' ('id_curso' ASC),
INDEX 'fk_Docente_has_Curso_Docente1_idx' ('id_docente' ASC),
CONSTRAINT 'fk_Docente_has_Curso_Docente1'
FOREIGN KEY ('id_docente') REFERENCES 'extrebooks'.Docente ('NRP')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Docente_has_Curso_Curso1'
FOREIGN KEY ('id_curso')
REFERENCES 'extrebooks'.Curso ('IDCURSO')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

– **Table 'extrebooks'.Alumno_has_Asignatura'** –

```

CREATE TABLE IF NOT EXISTS 'extrebooks'.Alumno_has_Asignatura' (
'id_alumno' INT NOT NULL,
'id_asignatura' INT NOT NULL,
PRIMARY KEY ('id_alumno', 'id_asignatura'),
INDEX 'fk_Alumno_has_Asignatura_Asignatura1_idx' ('id_asignatura' ASC),
INDEX 'fk_Alumno_has_Asignatura_Alumno1_idx' ('id_alumno' ASC),
CONSTRAINT 'fk_Alumno_has_Asignatura_Alumno1'
FOREIGN KEY ('id_alumno')
REFERENCES 'extrebooks'.Alumno ('NIA')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Alumno_has_Asignatura_Asignatura1'

```

```
FOREIGN KEY ('id_asignatura')
REFERENCES 'extrebooks'.'Asignatura' ('IDASIGNATURA')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

– **Table 'extrebooks'.'Materiales_has_Asignatura'** –

```
CREATE TABLE IF NOT EXISTS 'extrebooks'.'Materiales_has_Asignatura' (
'Materiales_idMateriales' INT NOT NULL,
'Asignatura_idAsignatura' INT NOT NULL,
PRIMARY KEY ('Materiales_idMateriales', 'Asignatura_idAsignatura'),
INDEX 'fk_Materiales_has_Asignatura1_Asignatura1_idx' ('Asignatura_idAsignatura' ASC),
INDEX 'fk_Materiales_has_Asignatura1_Materiales1_idx' ('Materiales_idMateriales' ASC),
CONSTRAINT 'fk_Materiales_has_Asignatura1_Materiales1'
FOREIGN KEY ('Materiales_idMateriales')
REFERENCES 'extrebooks'.'Materiales' ('IDMATERIAL')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Materiales_has_Asignatura1_Asignatura1'
FOREIGN KEY ('Asignatura_idAsignatura')
REFERENCES 'extrebooks'.'Asignatura' ('IDASIGNATURA')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

– **Table 'extrebooks'.'Materiales_has_Curso'** –

```
CREATE TABLE IF NOT EXISTS 'extrebooks'.'Materiales_has_Curso' (
'Materiales_idMateriales' INT NOT NULL,
'Curso_idCurso' INT NOT NULL,
PRIMARY KEY ('Materiales_idMateriales', 'Curso_idCurso'),
INDEX 'fk_Materiales_has_Curso_Curso1_idx' ('Curso_idCurso' ASC),
INDEX 'fk_Materiales_has_Curso_Materiales1_idx' ('Materiales_idMateriales' ASC),
CONSTRAINT 'fk_Materiales_has_Curso_Materiales1'
FOREIGN KEY ('Materiales_idMateriales')
REFERENCES 'extrebooks'.'Materiales' ('IDMATERIAL')
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT 'fk_Materiales_has_Curso_Curso1'
FOREIGN KEY ('Curso_idCurso')
REFERENCES 'extrebooks'.'Curso' ('IDCURSO')
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

– **Table ‘extrebooks’.‘Recomendar’** –

```
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Recomendar’ (  
‘id_peticion’ INT NOT NULL,  
‘nom_recurso’ VARCHAR(45) NULL,  
‘id_alumno’ INT NOT NULL,  
‘descripcion’ VARCHAR(45) NULL,  
‘id_docente’ VARCHAR(45) NOT NULL,  
‘id_asignatura’ INT NOT NULL, PRIMARY KEY (‘id_peticion’),  
INDEX ‘fk_Recomendar_Alumno1_idx’ (‘id_alumno’ ASC),  
INDEX ‘fk_Recomendar_Docente1_idx’ (‘id_docente’ ASC),  
INDEX ‘fk_Recomendar_Asignatura1_idx’ (‘id_asignatura’ ASC),  
CONSTRAINT ‘fk_Recomendar_Alumno1’  
FOREIGN KEY (‘id_alumno’)  
REFERENCES ‘extrebooks’.‘Alumno’ (‘NIA’)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT ‘fk_Recomendar_Docente1’  
FOREIGN KEY (‘id_docente’)  
REFERENCES ‘extrebooks’.‘Docente’ (‘NRP’)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT ‘fk_Recomendar_Asignatura1’  
FOREIGN KEY (‘id_asignatura’)  
REFERENCES ‘extrebooks’.‘Asignatura’ (‘IDASIGNATURA’)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

– **Table ‘extrebooks’.‘Direccion’** –

```
CREATE TABLE IF NOT EXISTS ‘extrebooks’.‘Direccion’ (  
‘ID_DIRECCION’ INT NOT NULL,  
‘id_centro’ VARCHAR(9) NOT NULL,  
‘id_docenteDirector’ VARCHAR(10) NOT NULL,  
‘id_docenteJefe’ VARCHAR(10) NULL,  
‘id_docenteSecretario’ VARCHAR(45) NULL,  
PRIMARY KEY (‘ID_DIRECCION’),  
INDEX ‘fk_Direccion_Centro1_idx’ (‘id_centro’ ASC),  
INDEX ‘fk_Direccion_Docente1_idx’ (‘id_docenteDirector’ ASC),  
CONSTRAINT ‘fk_Direccion_Centro1’  
FOREIGN KEY (‘id_centro’)  
REFERENCES ‘extrebooks’.‘Centro’ (‘CIF’)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT ‘fk_Direccion_Docente1’  
FOREIGN KEY (‘id_docenteDirector’)  
REFERENCES ‘extrebooks’.‘Docente’ (‘NRP’)
```

```
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```