



**UNIVERSIDAD DE EXTREMADURA  
CENTRO UNIVERSITARIO DE MÉRIDA**

**GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA  
INFORMACIÓN**

**TRABAJO FIN DE GRADO**

**DESARROLLO DE APLICATIVO PARA  
AUTOMATIZAR LA EMISIÓN DE  
CERTIFICADOS EN EL ESTANDAR EDCI  
EUROPASS**

**AUTOR/A: RICHARD CORCHERO VÉLEZ**

**Mérida, julio 2022**





**UNIVERSIDAD DE EXTREMADURA  
CENTRO UNIVERSITARIO DE MÉRIDA**

**GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA  
INFORMACIÓN**

**TRABAJO FIN DE GRADO**

**DESARROLLO DE APLICATIVO PARA  
AUTOMATIZAR LA EMISIÓN DE  
CERTIFICADOS EN EL ESTÁNDAR EDCI  
EUROPASS**

Autor/a: Richard Corchero Vélez  
Fdo.:

Tutor o tutores: Héctor Sánchez Santamaría  
Fdo.:



## Resumen

Teniendo en cuenta la importancia de la globalización y la digitalización del mundo en el que vivimos y la dificultad para verificar que información es cierta y cual no, este Trabajo Fin de Grado (TFG) tiene como objetivo diseñar y desarrollar un aplicativo que permita automatizar la emisión de certificados tras la realización exitosa de una formación no reglada siguiendo el estándar EDCI (Europass Digital Credentials Infrastructure) de la plataforma Europass.

El estándar EDCI es una propuesta de la Comisión Europea que nace en el año 2019 y sigue en constante evolución. El objeto de este es la creación de una infraestructura técnica que las entidades u organizaciones puedan utilizar para emitir credenciales digitales en toda la Unión Europea (UE). Este estándar proporciona un sistema seguro, fiable y resistente al fraude que garantiza la privacidad y la protección de los datos. Además, garantiza que todos los certificados emitidos por un Estado miembro puedan entenderse y verificarse en cualquier otro de manera rápida y segura.

En el apartado tecnológico, el aplicativo va a ser desarrollado como una API REST, de modo que podamos agilizar el proceso de comunicación entre la entidad que quiera emitir sus certificados en el formato EDCI, bajo el lenguaje de programación Java.

El informe final de este TFG incluirá principalmente:

- Un estudio sobre el estándar EDCI y la plataforma Europass, donde explicaremos las características del formato, requisitos necesarios para poder emitir esos certificados y su conexión con la plataforma Europass.
- Un análisis de las tecnologías escogidas y la importancia de uso que tienen para llevar a cabo el proyecto.
- La descripción de las fases de diseño e implementación del aplicativo encargado de la emisión de los certificados.

**Palabras clave:** EDCI, Automatización de certificados, Java, API REST



## **Abstract**

In view of the importance of globalisation and digitalisation of the world in which we live and the difficulty of verifying what information is true and what is not, the aim of this Final Degree Project (FDP) is to design and develop an application to automate the issuing of certificates after the successful completion of non-regulated training, following the EDCI (Europass Digital Credentials Infrastructure) standard of the Europass platform.

The EDCI standard is a proposal from the European Commission that was born in 2019 and is constantly evolving. It aims to create a technical infrastructure that entities or organisations can use to issue digital credentials throughout the European Union (EU). This standard provides a secure, reliable, and fraud-resistant system that guarantees privacy and data protection. It also ensures that all certificates issued by one Member State can be understood and verified in any other Member State quickly and securely.

In the technological section, the application will be developed as a REST API, so that we can speed up the communication process between the entity that wants to issue its certificates in EDCI format, under the Java programming language.

The final report of this FDP will mainly include:

- A study on the EDCI standard and the Europass platform, where we will explain the characteristics of the format, the requirements necessary to be able to issue these certificates and their connection with the Europass platform.
- An analysis of the chosen technologies and the importance of their use for carrying out the project.
- The description of the design and implementation phases of the application responsible for issuing the certificates.

**Keywords:** EDCI, Certificate automation, Java, API REST





## **Agradecimientos**

En primer lugar, quiero dar las gracias a Héctor Sánchez Santamaría, director de este Trabajo Fin de Grado, quien me propuso la realización de este proyecto y me metió de lleno en un tema desconocido para mí, que ha resultado muy interesante. También agradecer toda la ayuda, atención e indicaciones que me ha ofrecido durante la realización de este.

Me gustaría dedicar este Trabajo Fin de Grado a mi familia y amigos por todo el esfuerzo que han realizado estos años, que ha permitido el estudio de la titulación que estoy a punto de finalizar, y por todo el apoyo y ánimo que me han dado cada año.

También dedicárselo a mi pareja, por ser mi sostén día tras día durante estos cuatro años. Gracias por confiar en mí.

Me gustaría agradecer también a todos y todas mis compañeros y compañeras con los que he compartido estos años, en especial a Álvaro y Mario, con los cuales, no solo he compartido el significado de la palabra compañerismo, también he establecido una sólida amistad.

Por última, me gustaría agradecer al Centro Universitario de Mérida, tanto al equipo docente como al personal de administración y servicios, toda la formación que me han impartido y toda la ayuda que me han brindado durante estos años.

Muchas gracias a todos.



# Índice general

<b>Introducción.....</b>	<b>1</b>
1.1. Introducción .....	1
1.2. Antecedentes .....	3
1.3. Objetivos .....	4
1.4. Estructura del documento.....	4
<b>Análisis previo .....</b>	<b>6</b>
2.1. Europass Digital Credentials Infraestructure (EDCI).....	7
<b>2.1.1. eIDAS.....</b>	<b>8</b>
<b>2.1.2. Estándares.....</b>	<b>10</b>
<b>2.1.3. Servicios.....</b>	<b>12</b>
<b>2.1.4. Software.....</b>	<b>13</b>
2.2. Emisión de credenciales para entidades emisoras.....	13
2.3. Solicitud del e-Seal para la emisión de las credenciales EDCI.....	14
<b>2.3.1 Diferencias entre QES (Qualified Electronic Seal) y AES (Advanced Electronic Seal)..</b>	<b>15</b>
<b>2.3.2 Diferencias entre e-Signature (Electronic Signature) y e-Seal (Electronic Seal).....</b>	<b>16</b>
2.4. Especificación del formato de los datos para EDCI.....	17
<b>2.4.1. Hoja 1: Credentials .....</b>	<b>17</b>
<b>2.4.2. Hoja 2: Persons.....</b>	<b>20</b>
<b>2.4.3. Hoja 3: Organisations .....</b>	<b>22</b>
<b>2.4.4. Hoja 4: Achievements .....</b>	<b>24</b>
<b>2.4.5. Hoja 5: Learning Outcomes .....</b>	<b>25</b>
<b>2.4.6. Hoja 6: Activities .....</b>	<b>26</b>
<b>2.4.7. Hoja 7: Assessments .....</b>	<b>29</b>
<b>2.4.8. Hoja 8: Entitlements .....</b>	<b>30</b>
2.5. Proceso de subida, validación, sellado, emisión y envío de las credenciales EDCI.....	31
<b>2.5.1. Testeo del Sello electrónico.....</b>	<b>32</b>
<b>2.5.2. Subida y validación del lote de credenciales.....</b>	<b>34</b>
<b>2.5.3. Previsualización de las credenciales digitales.....</b>	<b>35</b>
<b>2.5.4. Sello de las credenciales digitales mediante e-Seal.....</b>	<b>37</b>
<b>2.5.5. Emisión y envío de las credenciales por correo electrónico y al wallet del alumno.....</b>	<b>38</b>
<b>Metodología.....</b>	<b>41</b>
3.1. Metodología Iterativa Incremental .....	41
<b>3.1.1 Introducción.....</b>	<b>41</b>
<b>3.1.2 Etapas .....</b>	<b>42</b>

<b>3.1.3 Beneficios e Inconvenientes .....</b>	<b>44</b>
3.2. Metodología Orientada a Objetos.....	44
<b>Análisis del sistema.....</b>	<b>47</b>
4.1. Análisis de viabilidad .....	47
<b>4.1.1 Viabilidad técnica .....</b>	<b>48</b>
<b>4.1.2 Viabilidad económica .....</b>	<b>48</b>
<b>4.1.3 Viabilidad operacional.....</b>	<b>48</b>
4.2. Planificación estimada.....	49
<b>4.2.1 Estructura de desglose de trabajo (EDT).....</b>	<b>49</b>
<b>4.2.2 Diccionario de la EDT .....</b>	<b>51</b>
<b>4.2.3 Cronograma.....</b>	<b>55</b>
<b>Diseño .....</b>	<b>58</b>
5.1. Plantilla Excel EDC.....	59
<b>5.1.1 JExcel.....</b>	<b>59</b>
<b>5.1.2 Apache POI.....</b>	<b>60</b>
<b>5.1.3 Microsoft Visual Basic Script.....</b>	<b>61</b>
<b>5.1.4 Conclusiones .....</b>	<b>63</b>
5.2. API REST .....	63
<b>5.2.1 JAX-RS.....</b>	<b>65</b>
<b>5.2.2 Jersey .....</b>	<b>65</b>
<b>5.2.3 Conclusiones .....</b>	<b>65</b>
5.3. HTML EDC Wildcard.....	66
5.4. Formato de envío de datos.....	67
<b>5.4.1 XML.....</b>	<b>68</b>
<b>5.4.2 JSON.....</b>	<b>69</b>
<b>5.4.3 Conclusiones .....</b>	<b>70</b>
5.5. Ecosistema del proyecto.....	70
<b>Implementación .....</b>	<b>73</b>
6.1. Configuración de la plantilla Excel EDC .....	74
<b>6.1.1 Macros en Excel.....</b>	<b>74</b>
<b>6.1.2 Adaptación de la Macro AddCredentials.....</b>	<b>75</b>
<b>6.1.3 Adaptación de la Macro AddPersons .....</b>	<b>76</b>
6.2. Automatización del cumplimentado de la plantilla Excel EDC .....	77
6.3. Creación de la HTML EDC Wildcard para el SOFD.....	79
6.4. Creación del servicio API REST mediante Jersey y JAX-RS.....	80
<b>6.4.1. Maven y Eclipse EE.....</b>	<b>80</b>
<b>6.4.2. Incluir dependencia Jersey en nuestro proyecto. ....</b>	<b>80</b>
<b>6.4.3. Desarrollo del servicio mediante las anotaciones JAX-RS. ....</b>	<b>81</b>
6.5. Procesamiento del JSON y llamada a VBScript.....	85
6.6. Despliegue, Pruebas y validación del servicio REST .....	87

<b>Conclusiones .....</b>	<b>92</b>
7.1. Conclusiones.....	92
7.2. Trabajos futuros.....	94
<b>Bibliografía.....</b>	<b>96</b>
<b>Anexo I. Manual del programador .....</b>	<b>100</b>
8.1. Tecnologías y Herramientas empleadas .....	101
8.2. Componentes y Estructura.....	101
<b>8.2.1. VBScript.....</b>	<b>102</b>
<b>8.2.2. Servicio REST.....</b>	<b>102</b>
8.3. Variables.....	104
<b>8.3.1. VBScript.....</b>	<b>104</b>
<b>8.3.2. EDCIConverter .....</b>	<b>105</b>
8.4. Funciones y Métodos.....	107
<b>8.4.1. VBScript Funciones.....</b>	<b>107</b>
<b>8.4.2. VBScript Algoritmos.....</b>	<b>108</b>
<b>8.4.3. EDCIConverter .....</b>	<b>109</b>



# Índice de figuras

1.1 Grafica de evolución de las infracciones por falsificación de documentos desde 2013 hasta 2019 ...	2
1.2 Grafica de evolución del número de perfiles Europass durante el año 2022.....	3
2.1 Ecosistema de EDCI.....	8
2.2 Principales servicios del eIDAS para EDCI.....	9
2.3 Ecosistema de eIDAS.....	10
2.4 Roles y flujos que constituyen la base de la especificación de W3C.....	12
2.5. NextU en ejecución.....	32
2.6. Apartado para probar nuestro sello electrónico.....	33
2.7. Probando nuestro e-Seal mediante EDCI Issuer.....	33
2.8. Apartado para subir nuestro lote de credenciales.....	34
2.9. Validación del lote de credenciales mediante EDCI Issuer.....	34
2.10. Previsualizando la credencial digital I.....	35
2.11. Previsualizando la credencial digital II.....	36
2.12. Selección y sellado de las credenciales.....	37
2.13. Resultado final del sellado de las credenciales.....	37
2.14. Envío de las credenciales digitales selladas en Europass I.....	38
2.15. Envío de las credenciales digitales selladas en Europass II.....	38
2.16. Credencial Digital sellada depositada en la cartera o wallet de un perfil Europass.....	39
3.1. Ejemplificación de la metodología iterativa incremental.....	42
3.2. Etapas de la metodología iterativa incremental.....	43
4.1. Estructura de desglose de trabajo.....	50
5.1. Uso de la librería JExcel en Java.....	60
5.2. Ayuda de WScript.....	62
5.3. Ejemplificación de una API REST.....	64
5.4. Estructura HTML por defecto de EDC.....	67
5.5. Estructura XML de una credencial emitida en la plataforma Europass.....	68
5.6. Estructura de un archivo en formato JSON.....	69
5.7. Estructura del sistema a desarrollar.....	70
6.1. Macro contenida en la plantilla Excel EDC que desprotege todas las hojas del archivo.....	74
6.2. Ejecución de la macro AddCredentials por defecto.....	75
6.3. Estructura HTML de las credenciales del SOFD.....	79
6.4. Dependencias incluidas en nuestro fichero POM.xml.....	81
6.5. Implementación del servicio de generación de credenciales EDC mediante JAX-RS.....	82
6.6. Formato de envío JSON de los datos referentes al lote de credenciales a generar.....	83
6.7. Diagrama de clase, del conjunto de componentes, necesarios para recoger el fichero JSON de la solicitud del cliente.....	84
6.8. Configuración del archivo web.xml con Jersey.....	85
6.9. Cliente web implementado para envío de los datos de las credenciales en formato JSON.....	87
6.10. Estructura HTML y código JavaScript de nuestro cliente web.....	88
6.11. Respuesta con éxito de la solicitud a nuestro servicio Rest.....	89
6.12. Credencial validada, sellada y emitida en la plataforma Europass del SOFD.....	90
8.1. Estructura de los archivos que componen la API REST.....	103
8.2. Declaración de constantes necesarias en el módulo EDCIConverter.....	106





# Índice de tablas

2.1. Principales características del sello electrónico avanzado y calificado.....	16
2.2. Principales características de la firma y sello electrónico. ....	16
2.3. Campos de la Hoja Credentials de la plantilla Excel EDC I. ....	18
2.4. Campos de la Hoja Credentials de la plantilla Excel EDC II. ....	19
2.5. Campos de la Hoja Persons de la plantilla Excel EDC. ....	21
2.6. Campos de la Hoja Organisations de la plantilla Excel EDC.....	23
2.7. Campos de la Hoja Achievements de la plantilla Excel EDC.....	25
2.8. Campos de la Hoja Learning Outcomes de la plantilla Excel EDC. ....	26
2.9. Campos de la Hoja Activities de la plantilla Excel EDC. ....	29
2.10. Campos de la Hoja Assessments de la plantilla Excel EDC. ....	30
2.11. Campos de la Hoja Entitlements de la plantilla Excel EDC.....	31
4.1. Diccionario EDT componentes. ....	53
4.2. Diccionario EDT tareas. ....	55
4.3. Cronograma. ....	56
5.1. Componentes de Apache POI.....	61



# Algoritmos

6.1. Código adaptado para nuestro servicio contenido en la macro AddCredentials. ....	76
6.2. Código adaptado para nuestro servicio contenido en la macro AddPersons. ....	76
6.3. Código de VBScript que genera el lote de credenciales EDC que será enviado de vuelta al cliente. .....	78
6.4. Implementación de la función que genera el proceso de construcción del lote de credenciales EDC. .....	86
8.1. Proceso que nos permite rellenar la plantilla Excel con la información de las credenciales. ....	109
8.2. Proceso que nos permite rellenar la plantilla Excel con el formato HTML e imagen de fondo de cada una de las credenciales. ....	109
8.3. Método que procesa el archivo JSON y devuelve la información en un formato entendible para el VBScript. ....	110
8.4. Función para procesar el nombre completo recibido en el fichero JSON como información de dos celdas diferentes. ....	111
8.5. Función que genera en formato String la plantilla personalizada HTML para las credenciales. ...	111
8.6. Función que transforma los títulos de las credenciales, superiores de 50 caracteres, para poder visualizarse correctamente en la credencial. ....	112



# Capítulo 1

## Introducción

### Contenidos

---

<b>1.1. Introducción.....</b>	<b>1</b>
<b>1.2. Antecedentes .....</b>	<b>3</b>
<b>1.3. Objetivos .....</b>	<b>4</b>
<b>1.4. Estructura del documento .....</b>	<b>4</b>

---

Este primer capítulo servirá para introducir este Trabajo Fin de Grado (TFG). Para ello, abordaremos los siguientes aspectos:

En la sección 1.1 se hablará de la dificultad actual para verificar que información es cierta y una pequeña introducción al estándar EDCI y la plataforma Europass.

En la sección 1.2 se describe los inicios de la plataforma Europass.

En la sección 1.3 se detallan los objetivos de este Trabajo Fin de Grado.

Finalmente, en la sección 1.4, se explicará brevemente el contenido del resto de capítulos de este TFG.

### 1.1. Introducción

Actualmente, vivimos en un mundo tecnológico y con una globalización masiva. Estos fenómenos traen consigo numerosos beneficios para la sociedad, como, por ejemplo, el aumento de las oportunidades comerciales y laborales, pero no todo son ventajas, uno de los problemas más comunes, y que además va en aumento, es la falsificación de credenciales<sup>1</sup>. La falsificación de credenciales o titulaciones se ha convertido en un problema global, llegando a representar un riesgo reputacional entre las instituciones académicas<sup>2</sup> y el mundo laboral.

---

<sup>1</sup> “Aumenta la venta de títulos universitarios falsos a través de Internet”, noticia de cadena SER, accesible en: [https://cadenaser.com/ser/2013/11/28/sociedad/1385608411\\_850215.html](https://cadenaser.com/ser/2013/11/28/sociedad/1385608411_850215.html)

<sup>2</sup> “Más de 20 universidades expiden títulos fácilmente falsificables y al margen de la ley”, noticia de El Confidencial, accesible en: [https://www.elconfidencial.com/espana/2018-10-07/universidades-espana-expiden-titulos-papel-no-legales\\_1621191/](https://www.elconfidencial.com/espana/2018-10-07/universidades-espana-expiden-titulos-papel-no-legales_1621191/)

Es por ello necesario infraestructuras que nos permitan digitalizar, validar y verificar las credenciales generadas por las diversas instituciones del mundo. En la figura 1.1<sup>3</sup>, podemos observar la evolución del número de infracciones relacionadas con la falsificación de documentos desde el año 2013 hasta el año 2019 en España, podemos observar un crecimiento notable de los casos según avanzamos.



Figura 1.1 Grafica de evolución de las infracciones por falsificación de documentos desde 2013 hasta 2019

Por otro lado, hoy en día, tenemos la facilidad para poder trabajar en cualquier parte del mundo. Sin embargo, cuando queremos desempeñar nuestra profesión en el extranjero, necesitamos pasar por un proceso de homologación para que nuestros estudios sean válidos. Este proceso requiere de entregar cierta información, pudiendo llegar a ser bastante tedioso y costoso en muchos casos.

Debido a las cuestiones expuestas, surge **EDCI (Europass Digital Credentials Infrastructure)**. **EDCI** es una propuesta de la Comisión Europea, nacida en el 2019, cuyo objetivo es la creación de una infraestructura técnica que las organizaciones puedan utilizar para emitir credenciales digitales en toda la UE (Unión Europea), proporcionando un sistema seguro, fiable y resistente al fraude que garantiza la privacidad y la protección de los datos. Además, nos garantiza que los certificados de un Estado miembros puedan entenderse y verificarse en cualquier otro fácilmente.

El desarrollo de un servicio que permita a las organizaciones generar credenciales en el formato **EDCI** y subirlas a la plataforma de Europass, son los principales marcos de desarrollo que se enmarcan en este TFG

<sup>3</sup> Datos y gráfica obtenida del Instituto Nacional de Estadística (INE), accesible en: <https://www.ine.es/jaxiT3/Tabla.htm?t=25997>

## 1.2. Antecedentes

En 1998 nace Europass, partiendo de la necesidad de constituir un Foro Europeo sobre la transparencia de las cualificaciones profesionales para reunir, de este modo, a interlocutores sociales y autoridades nacionales, con el fin de afrontar conjuntamente el tema, una cuestión vital en la movilidad de ciudadanos y trabajadores de la Unión Europea por todo su territorio.

En 2002, el Foro Europeo fue sustituido con el fin de incrementar la transparencia de la Formación Profesional por aplicación y racionalización de herramientas y redes, integrando herramientas, ya existentes (como CV europeo) en el Foro Europeo, en un único marco (la iniciativa Europass).

Tras dos años de desarrollo, en 2004, se desarrolló y aprobó un modelo de marco único y un prototipo de espacio web que llegó a los ciudadanos un año más tarde. Esta fue la primera versión de la plataforma Europass.

Durante los siguientes años, la plataforma ha seguido en constante evolución. En 2020, se llevó a cabo el lanzamiento del nuevo portal Europass, siendo esta solamente la primera fase de este lanzamiento, de las cuatro que la Comisión Europea tiene pensado cubrir.

En estos momentos, la plataforma cuenta con casi 4 millones de usuarios. En la figura 1.2<sup>4</sup>, se puede observar el número de usuarios registrados que han creado un perfil Europass a lo largo del año 2022, pudiéndose ver un constante crecimiento de perfiles en la plataforma.

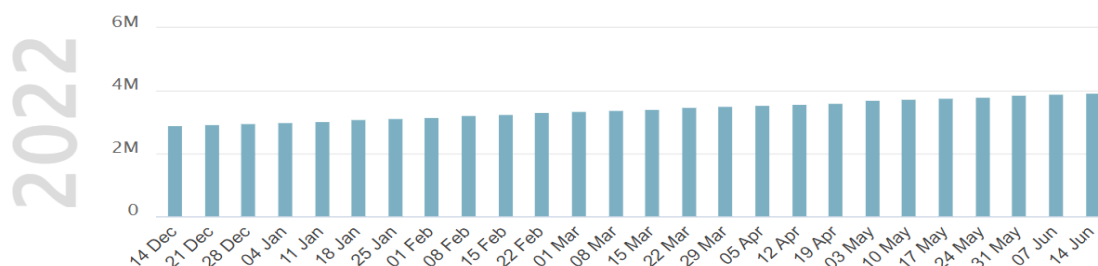


Figura 1.2 Grafica de evolución del número de perfiles Europass durante el año 2022

<sup>4</sup> Fuente: [1]

La Comisión Europea con el desarrollo de **EDCI** tiene como objetivo que todos los países miembros de la Unión Europea lleven a cabo la expedición de credenciales digitales en esta nueva infraestructura, durante los próximos años.

### 1.3. Objetivos

El objetivo de este TFG es diseñar y desarrollar un sistema informático que permita automatizar la generación de certificados digitales en el formato EDCI. Posteriormente, los certificados generados por la plataforma Europass deben ser firmados con un sello electrónico cualificado para garantizar la autenticidad del emisor y asegurar la validez del certificado. Para lograr este objetivo, se identifican los siguientes subobjetivos:

- Estudio del estándar EDCI y de la plataforma Europass.
- Estudio de las tecnologías para llevar a cabo su desarrollo
- Diseño y desarrollo del algoritmo de emisión de certificados.
- Diseño y desarrollo del servicio mediante API REST.
- Definición de la plantilla HTML según la apariencia del certificado emitido por el SOFD.
- Publicación de los certificados en Europass.

### 1.4. Estructura del documento

Este TFG está formado por 6 capítulos más:

- En el **segundo capítulo**, se expone todo el estudio previo realizado sobre la plataforma Europass y EDCI.
- En el **tercer capítulo**, se recogen las metodologías empleadas para la consecución de este TFG.
- En el **cuarto capítulo**, se detallará el análisis del servicio que se va a desarrollar junto con sus funcionalidades.
- En el **quinto capítulo**, se expondrá las características del diseño de nuestro aplicativo, así como las diferentes alternativas que se han tenido en cuenta en el mismo.
- En el **sexto capítulo**, se hablará sobre los detalles de la implementación del servicio.
- Por último, el **séptimo capítulo**, se establecen las conclusiones y trabajos futuros a partir de este TFG.





# Capítulo 2

## Análisis previo

### Contenidos

---

<b>2.1. Europass Digital Credentials Infraestructure (EDCI).....</b>	<b>7</b>
2.1.1. eIDAS.....	8
2.1.2. Estándares.....	10
2.1.3. Servicios.....	12
2.1.4. Software.....	13
<b>2.2. Emisión de credenciales para entidades emisoras.....</b>	<b>13</b>
<b>2.3. Solicitud del e-Seal para la emisión de las credenciales EDCI.....</b>	<b>14</b>
2.3.1 Diferencias entre QES (Qualified Electronic Seal) y AES (Advanced Electronic Seal)..	15
2.3.2 Diferencias entre e-Signature (Electronic Signature) y e-Seal (Electronic Seal).....	16
<b>2.4. Especificación del formato de los datos para EDCI.....</b>	<b>17</b>
2.4.1. Hoja 1: Credentials .....	17
2.4.2. Hoja 2: Persons .....	20
2.4.3. Hoja 3: Organisations .....	22
2.4.4. Hoja 4: Achievements .....	24
2.4.5. Hoja 5: Learning Outcomes .....	25
2.4.6. Hoja 6: Activities .....	26
2.4.7. Hoja 7: Assessments .....	29
2.4.8. Hoja 8: Entitlements .....	30
<b>2.5. Proceso de subida, validación, sellado, emisión y envío de las credenciales EDCI.....</b>	<b>31</b>
2.5.1. Testeo del Sello electrónico.....	32
2.5.2. Subida y validación del lote de credenciales.....	34
2.5.3. Previsualización de las credenciales digitales.....	35
2.5.4. Sello de las credenciales digitales mediante e-Seal.....	37
2.5.5. Emisión y envío de las credenciales por correo electrónico y al wallet del alumno.....	38

---

En este capítulo se expondrán detalladamente aquellos conceptos previos necesarios para el desarrollo de este Trabajo Fin de Grado.

En la sección 2.1 se explicará el concepto de EDCI junto con las características que ofrece.

En la sección 2.2 se detallará las fases necesarias para poder emitir credenciales en EDCI.

En la sección 2.3 se estudiará el concepto de e-Seal y su importancia dentro del proceso de emisión de las credenciales en la plataforma Europass.

En la sección 2.4 se explicará el formato que tienen que adquirir los datos de las organizaciones para poder llevar a cabo la validación de la credencial.

Finalmente, en la sección 2.5 se detallará el proceso de subida de los datos de credenciales preparados y los pasos de revisión, sellado digital, emisión y envío de la credencial.

## **2.1. Europass Digital Credentials Infrastructure (EDCI).**

EDCI es una iniciativa de la Comisión Europea puesta en marcha en el año 2019. Su principal misión es establecer una infraestructura estándar que pueda ser utilizada por todas las organizaciones y centros de la Unión Europea para la emisión de credenciales digitales, de manera, segura y fiable.

La seguridad y protección que ofrece EDCI a las organizaciones no es su única baza, sino que, además, permite que cualquier credencial digital emitida en esta infraestructura pueda ser fácilmente validada y verificada en cualquier país miembro de la Unión Europea, evitando así los procesos de homologación entre Estados miembros, reduciendo tanto el gasto temporal como los costes en personal.

Podemos desglosar EDCI [2] en cuatro partes: **eIDAS, Estándares, Servicios, Software.**

En la figura 2.1<sup>5</sup>, podemos observar este desglose de los componentes de EDCI, los cuales, trabajan de manera conjunta para crear credenciales digitales válidas legalmente, seguras y fácilmente verificables.

---

<sup>5</sup> Fuente: [2]

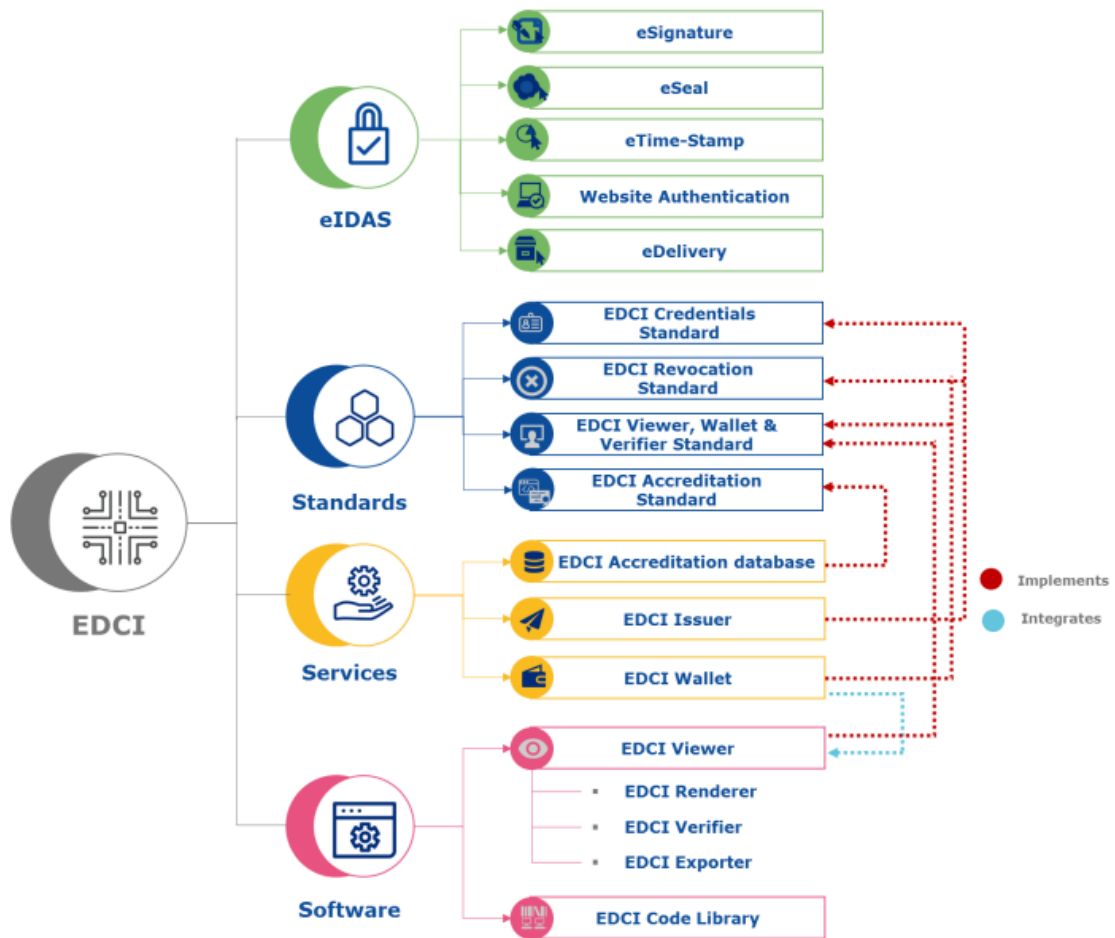


Figura 2.1 Ecosistema de EDCI.

### 2.1.1. eIDAS.

**eIDAS (electronic IDentification, Authentication and trust Services)** es un reglamento de la UE [3], aplicado desde 2016, sobre identificación electrónica y servicios de confianza referente a las transacciones electrónicas en el Mercado Único Europeo [4].

eIDAS ofrece a EDCI, un marco regulador sólido que protege y permite llevar a cabo las operaciones de autenticación, validación y emisión de credenciales digitales en toda la Unión Europea. En la figura 2.2<sup>6</sup>, podemos observar los principales componentes de eIDAS relevantes para EDCI.

<sup>6</sup> Fuente: [2]



Figura 2.2 Principales servicios del eIDAS para EDCI.

A continuación, se hace una breve descripción de la funcionalidad de cada componente:

- **e-Signature (Electronic Signature) o Firma electrónica**, sirve para identificar a los dueños de la credencial.
- **e-Seal (Electronic Seal) o Sello electrónico**, sirve para identificar a las organizaciones en el sello digital de las credenciales.
- **eTime-Stamp (Electronic Time Stamp) o Sello Temporal**, sirve para marcar temporal (establecer fecha y hora) de todas las transacciones que forman parte del sistema.
- **Website Authentication o Autenticación de sitio web**, sirve asegurar y autenticar todos los datos procedentes de una web.
- **eDelivery (Electronic Delivery) o Entrega Electrónica**, sirve para transferir datos digitales de manera segura. En el caso de EDCI, nos permite transferir las credenciales a las wallets o carteras de los usuarios de la plataforma Europass.

En próximos apartados de este capítulo, se detallarán mejor los conceptos anteriores y su función dentro de la emisión de las credenciales por las instituciones. A continuación, en la figura 2.3<sup>7</sup> podemos observar una imagen del ecosistema eIDAS.

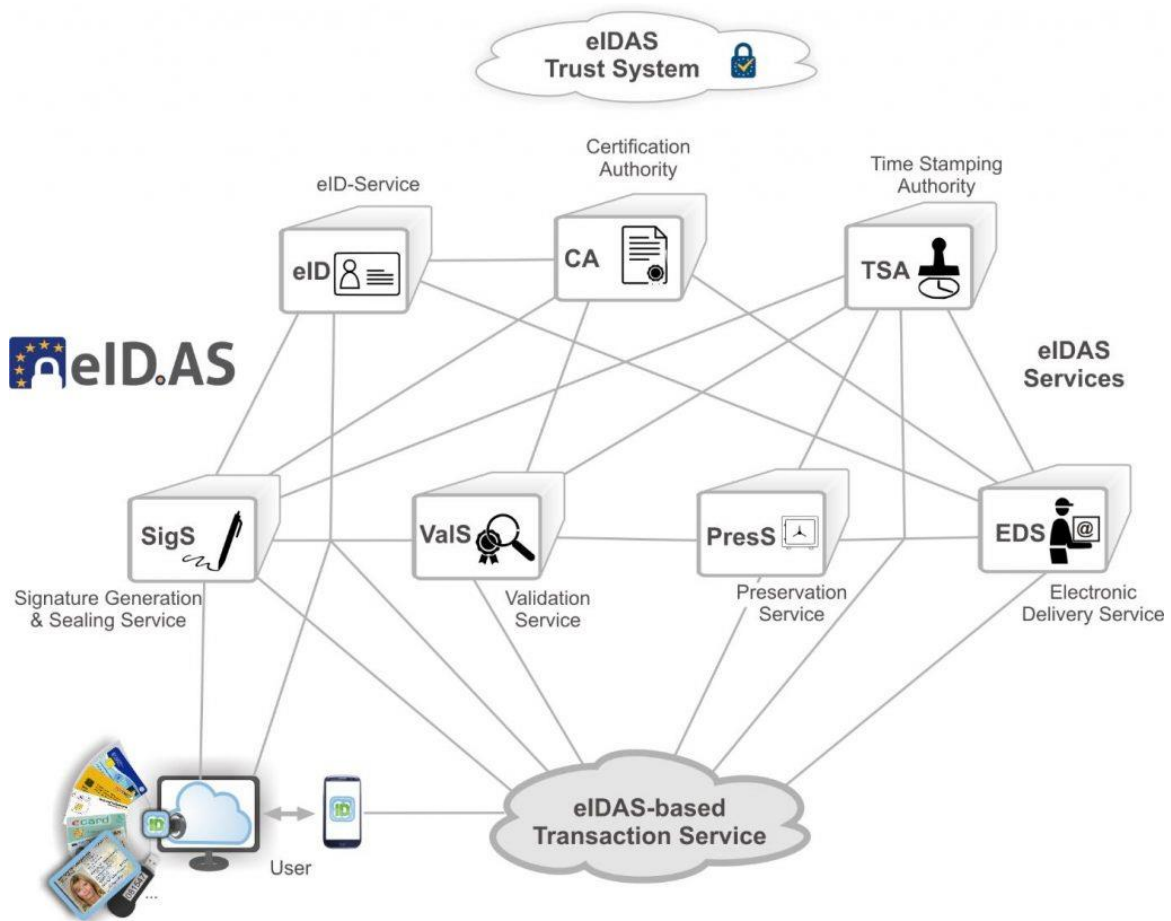


Figura 2.3 Ecosistema de eIDAS

## 2.1.2. Estándares.

Para poder llevar a cabo EDCI, la Comisión Europea desarrollo una serie de estándares sobre los que sustentarán los servicios que proporcionaría la plataforma Europass, referidos a la emisión de credenciales digitales. A continuación, se hace una pequeña descripción del alcance de cada estándar:

<sup>7</sup> Imagen e información obtenida del blog oficial de eIDAS, accesible en: <https://blog.eid.as>

- **EDCI Credentials Standard**, encargado de detallar el formato, los metadatos y las características de seguridad de una credencial firmada digitalmente conforme a Europass.
- **EDCI Credential Revocation Standard**, encargado de detallar el formato de los certificados de revocación de credenciales, los requisitos de las listas de revocación, así como las API para consultar dichas listas.
- **EDCI Viewer, Wallet & Verifier Standard**, encargado de detallar las funcionalidades básicas del software de la cartera de credenciales (Wallet) digitales de EDCI, el visor de credenciales (Viewer) y el estándar de verificación de EDCI.
- **EDCI Accreditation Standard**, encargado de detallar a los Estados miembros de la UE como deben almacenar los datos de cualificación para que sean accesible por los servicios de la UE, como pueden guardar directamente estos datos en la base de datos de acreditación de EDCI a través de la API y como el software de terceros puede consultar dicha base de datos.

Para llevar a cabo la creación de los estándares anteriores, la Comisión Europea diseño también modelo de datos para EDCI.

El modelo de datos de EDCI (**EDCI Data Model**) es una extensión del modelo de datos de credenciales verificables del W3C [5] expresado como XML/XSD, con el fin de proporcionar un formato estandarizado de descripción del aprendizaje en la Unión Europea y el Espacio Económico Europeo. Además, El modelo de datos también está alineado con el estándar ELMO/EMREX basado también en el estándar XML [6]. En la figura 2.4<sup>8</sup>, podemos observar un ejemplo del ecosistema en el que se basa esta especificación.

---

<sup>8</sup> Fuente: [5]

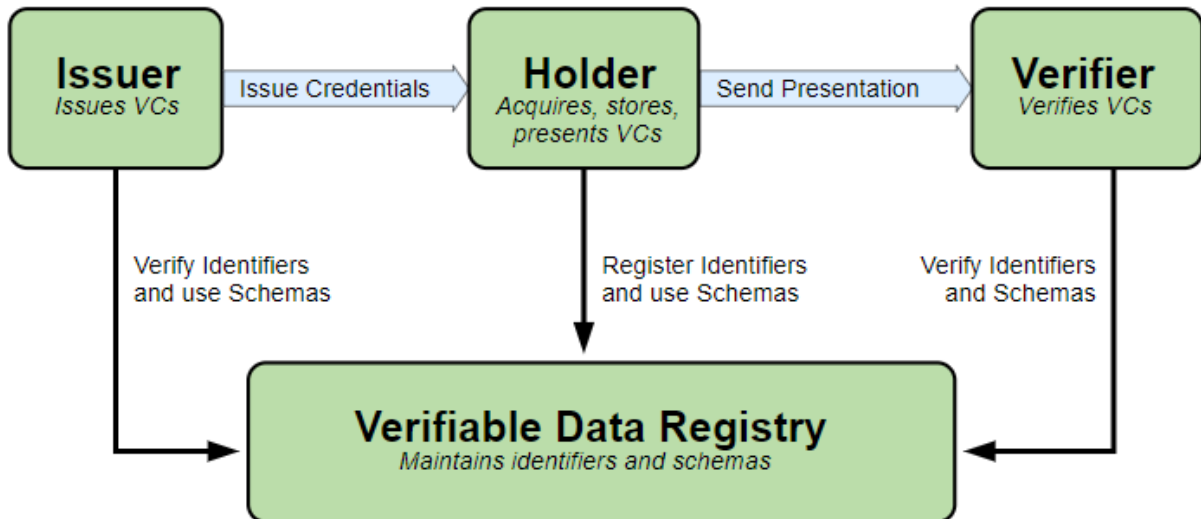


Figura 2.4 Roles y flujos que constituyen la base de la especificación de W3C.

### 2.1.3. Servicios.

EDCI ofrece una serie de servicios, los cuales, están contruidos o implementados sobre los estándares anteriores. Estos servicios se ejecutarán como servicios en la nube, pudiendo ser utilizados mediante la plataforma Europass. A continuación, realizaremos una breve descripción de los servicios de EDCI:

- **EDCI Accreditation Database**, este servicio ofrece un registro de todas las credenciales que han sido acreditadas y que se corresponden con el **EQF (The European Qualifications Framework for Lifelong Learning)** [7].
- **EDCI Issuer**, este servicio ofrece una herramienta de software basada en la web que permite a cualquier organismo adjudicador (instituciones de educación y capacitación) emitir lotes de credenciales digitales validas legalmente.
- **EDCI Wallet**, este servicio ofrece una cartera o billetera web mediante la cual cualquier propietario podrá ver, almacenar, verificar y compartir sus credenciales digitales en la nube. Este servicio es fundamental para que las instituciones puedan emitir las credenciales digitales de sus usuarios directamente a su cartera o billetera de la plataforma Europass.



---

## 2.1.4. Software.

Podemos dividir el software de EDCI en dos bloques:

- **EDCI Viewer** (también conocido como visor de credenciales), es la parte del software que permite, a los propietarios, ver y verificar las credenciales digitales.
- **EDCI Code Library**, librería formada por fragmentos extraídos de **EDCI Wallet** y de **EDCI Issuer**, que ayudan a terceros a desarrollar sus propias versiones o a integrar sistemas de terceros con los servicios de EDCI.

El **EDCI Viewer** está formado por tres módulos. A continuación, se detalla una breve descripción de cada módulo:

- **EDCI Renderer**, módulo que permite a los usuarios ver sus credenciales digitales en un formato gráfico establecido por el organismo o institución que emitió la credencial.
- **EDCI Verifier**, módulo encargado de comprobar que la credencial digital cumple con las reglas de verificación establecidas en los estándares.
- **EDCI Exporter**, módulo que permite la conversión de la credencial digital en otros formatos como PDF.

La Comisión Europea pone a disposición de terceros un repositorio en Github<sup>9</sup> con el contenido de **EDCI Code Library**, de manera que las organizaciones puedan incorporar rápidamente los componentes de EDCI en su software y llevar a cabo sus desarrollos propios.

## 2.2. Emisión de credenciales para entidades emisoras.

Para llevar a cabo la emisión de las credenciales, utilizaremos el servicio **EDCI Issuer**<sup>10</sup>. Como comentamos en el capítulo anterior, este servicio ofrece una herramienta web que permite a cualquier institución emitir lotes de credenciales digitales y enviarlas mediante correo electrónico o directamente al **EDCI Wallet**

---

<sup>9</sup> Repositorio EDCI Code Library de la Comisión Europea, accesible en: <https://github.com/ec-europa>

<sup>10</sup> Servicio EDCI Issuer accesible en: <https://webgate.acceptance.ec.europa.eu/europass/edci-issuer/#/home>

de los usuarios de la plataforma Europass.

Dentro del proceso de emisión de las credenciales, podemos destacar tres fases:

- **Fase de preparación (Prepare):** durante esta primera fase para la emisión de las credenciales digitales, deberemos obtener un sello electrónico (**e-Seal**) para sellar nuestras credenciales digitalmente y estudiar el formato que deben tener los datos que forman parte de la credencial para poder exportarlos en EDCI.
- **Fase de Construcción (Build):** durante esta fase, llevaremos a cabo la generación del lote de credenciales que se quieren emitir, junto con la creación del formato visual que tendrá la credencial (aspecto de la credencial).
- **Fase de Subida (Upload):** En esta última fase, subiremos el lote de credenciales, se comprobará que el formato es válido, se sellarán con el **e-Seal** y finalmente serán enviadas tanto por correo electrónico como al **Wallet** de cada uno de los usuarios de la plataforma Europass.

### 2.3. Solicitud del e-Seal para la emisión de las credenciales EDCI.

Para poder llevar a cabo la emisión de credenciales **EDCI** [8], las organizaciones necesitarán de un sello electrónico calificado o avanzado (**Qualified or Advanced e-Seal**). Este sello electrónico calificado cumple con el reglamento **eIDAS**, pudiendo considerarse como equivalentes digitales a los sellos de personas jurídicas en papel y permitiendo verificar e identificar al emisor de una credencial durante largos períodos de tiempo.

Estos sellos solo pueden ser generados o creados por un dispositivo electrónico calificado y basado en un certificado calificado para un sello electrónico. Debido a la premisa anterior, la organización deberá solicitar este sello a algún proveedor de servicios de confianza (**Qualified Trust Services**) en su país. Para ello, dentro de la plataforma Europass se ofrece un listado de proveedores<sup>11</sup> que ofrecen estos servicios en cada país.

---

<sup>11</sup> Listado de proveedores de Trust Services accesible en: <https://webgate.ec.europa.eu/tl-browser/#/>

Los **Trust Services** o **servicios de confianza** en español, es un tipo de servicio que permite crear, verificar y validar firmas y sellos electrónicos, marcas temporales, servicios de entrega registrados electrónicamente, certificados electrónicos y autenticación de sitios web.

### 2.3.1 Diferencias entre QES (Qualified Electronic Seal) y AES (Advanced Electronic Seal).

El **eIDAS** establece diferentes niveles tanto de firmas como de sellos electrónicos en su reglamento. Tanto **QES** y **AES** son sellos electrónicos y cumplen con las características básicas de los mismos. Dentro de esta jerarquía de niveles, **AES** es un sello de nivel inferior que **QES**, de modo, que este primero ofrece menos características. En la tabla 2.1<sup>12</sup>, podemos observar las principales diferencias entre estos dos niveles de **e-Seal**.

	QES	AES
<b>Permite la integridad de los datos y exactitud de su origen, incluso entre Estados miembros.</b>	SI	NO
<b>Permite la integridad de los datos y la exactitud de su origen para el ámbito político o legislativo en un Estado Miembro.</b>	SI	SI
<b>Es admisible en casos, donde se necesitaría una firma notarial para confirmar la autenticidad.</b>	SI	NO
<b>Es probable que sea aceptado por una administración pública.</b>	SI	QUIZAS

<sup>12</sup> Fuente: Traducción propia de [8]

<b>Es probable que sea aceptado por un empleador u organización.</b>	SI	SI
----------------------------------------------------------------------	----	----

Tabla 2.1. Principales características del sello electrónico avanzado y calificado.

### 2.3.2 Diferencias entre e-Signature (Electronic Signature) y e-Seal (Electronic Seal).

Actualmente, tenemos la posibilidad, en multitud de procesos, de firmar electrónicamente, logrando los mismos efectos que cuando utilizamos la firma manuscrita. Estas firmas electrónicas también están bajo el amparo del reglamento del **eIDAS**, del mismo modo que los sellos electrónicos.

En la tabla 2.2<sup>13</sup>, podemos observar que la principal diferencia entre una firma y un sello electrónico es que, un sello autentica el origen de una persona jurídica (organización), mientras que una firma autentica el origen de una persona física.

Servicio de Confianza	Integridad de los datos	Confidencialidad	Autentica el origen (Persona física)	Autentica el origen (Persona Jurídica)	Autentica el tiempo
e-Signature	SI	NO	SI	NO	NO
e-Seal	SI	NO	NO	SI	NO

Tabla 2.2. Principales características de la firma y sello electrónico.

<sup>13</sup> Fuente: Traducción propia de [8]

---

## 2.4. Especificación del formato de los datos para EDCI.

Para poder llevar a cabo la emisión de las credenciales digitales de una organización en EDCI, deberemos adaptar nuestros datos a la plantilla **EDC Excel**.

La plantilla **EDC Excel** [8], es un libro habilitado para macros de Excel (extensión .xlsm), con un formato específico de los datos, que nos permite preparar conjuntos de datos de lotes de credenciales. Esta plantilla se divide en 8 hojas para llevar a cabo el registro de los datos de las credenciales de manera sistemática:

- La hoja 1 (**Credentials**), es la hoja principal de la plantilla EDC Excel y en ella, se recoge datos referentes a la credencial que vamos a emitir.
- La hoja 2 (**Persons**), esta hoja recoge los datos referentes a la persona que va a recibir la credencial.
- La hoja 3 (**Organisations**), esta hoja recoge la información referente a la organización o institución registrada que emite la credencial.
- La hoja 4 (**Achievements**), esta hoja recoge informaciones referentes a los logros del título o credencial.
- La hoja 5 (**Learning Outcomes**), esta hoja recoge la información referente a los resultados de aprendizaje del título o credencial.
- La hoja 6 (**Activities**), esta hoja recoge la información referente a las actividades que se referenciarán en la credencial.
- La hoja 7 (**Assessments**), esta hoja recoge la información referente a las evaluaciones que se referenciarán en la credencial.
- La hoja 8 (**Entitlements**), esta hoja recoge la información referente a los derechos que otorga esta credencial.

En los siguientes subapartados, profundizaremos en cada una de las hojas anteriores y en la información que recogen o necesitan para poder llevar a cabo la emisión de las credenciales.

### 2.4.1. Hoja 1: Credentials

La primera hoja, denominada “**Credentials**”, recoge los datos referentes a la credencial que se va a emitir (como lenguaje oficial del título, título de la credencial, descripción, etc). Esta hoja también podremos establecer el formato visual que tendrá nuestra credencial, mediante el lenguaje de etiquetas **HTML** [9].

Dentro de esta hoja, tenemos una serie de campos que son obligatorios para poder emitir la credencial y que toda organización deberá cumplimentar mínimamente. En la tabla 2.3<sup>14</sup>, podemos observar cuales son estos campos y que información recogen.

Nombre del campo	Obligatorio	Información
<b>Primary language</b>	SI	Idioma principal de la credencial que va a ser emitida.
<b>Title</b>	SI	El título oficial que recibirá la credencial.
<b>Description</b>	NO	Descripción de la credencial (Máximo 140 palabras).
<b>Valid from</b>	SI	Fecha de emisión de la credencial (el formato es dd/mm/yyyy, ejemplo 20/06/2021).
<b>Expiry date</b>	NO	Fecha de expiración de la credencial (el formato es dd/mm/yyyy, ejemplo 20/06/2021).
<b>Type</b>	SI	Tipo de la credencial, podemos elegir entre: <b>1-Generic,</b> <b>2-Qualification Award</b> <b>3-Diploma Supplement.</b>
<b>Owner</b>	SI	La persona que va a recibir la credencial. Esta información referencia a la hoja 2 ( <b>Persons</b> ).
<b>Issuer</b>	SI	La organización que emite y sella digitalmente la credencial. Esta información referencia a la hoja 3 ( <b>Organisations</b> ).

Tabla 2.3. Campos de la Hoja Credentials de la plantilla Excel EDC I.

<sup>14</sup> Fuente: Elaboración propia basada en [8].

En esta misma hoja, también figura un apartado llamado **Credential Wildcards**. Este apartado contiene una serie de campos, todos opcionales, que nos permiten modificar el estilo visual de nuestra credencial. En la tabla 2.4<sup>15</sup>, podemos observar los nombres y una descripción de los campos.

<b>Nombre del campo</b>	<b>Información</b>
<b>Background</b>	Permite establecer el fondo de la credencial mediante una URL a una imagen. Formato DIN A4.
<b>HTML<sup>16</sup></b>	Podemos establecer en HTML la disposición de los elementos de la credencial.
<b>Thymeleaf Template</b>	Podemos modificar el formato de la credencial digital mediante el motor de plantillas Java Thymeleaf [10].

Tabla 2.4. Campos de la Hoja Credentials de la plantilla Excel EDC II.

<sup>15</sup> Fuente: Elaboración propia basada en [8].

<sup>16</sup> Información completa sobre la estructura HTML de la credencial digital accesible en: <https://europa.eu/europass/system/files/2020-11/EDCI-Diplomawildcards-171120-1131-2172.pdf>

## 2.4.2. Hoja 2: Persons

La segunda hoja, denominada “**Persons**”, recoge los datos referentes a la persona que va a recibir la credencial (como nombre, apellidos, etc..). Dentro de esta hoja, tenemos una serie de campos que la organización tendrá que cumplimentar mínimamente. En la tabla 2.5<sup>17</sup>, podemos observar cuales son estos campos y la información que recoge cada uno.

Nombre del campo	Obligatorio	Información
<b>Given Name</b>	SI	Nombre de pila de la persona.
<b>Family Name</b>	SI	Apellido de la persona.
<b>Date of Birth</b>	NO	Fecha de nacimiento. Formato dd/mm/aaaa, ejemplo 20/06/2000.
<b>Gender</b>	NO	Genero de la persona, tenemos 4 opciones: <b>-male (masculino).</b> <b>-female (femenino).</b> <b>-not aplicable (no aplica).</b> <b>-not known (no conocido).</b>
<b>National ID Number</b>	NO	Número del documento nacional de identificación de la persona.
<b>National ID Country</b>	NO	País referente al documento nacional de identidad. (Campo seleccionable)
<b>Place of Birth</b>	NO	Lugar de nacimiento. (Campo seleccionable).

<sup>17</sup> Fuente: Elaboración propia basada en [8].



<b>Country of citizenship</b>	NO	País que ha concedido los derechos de la ciudadanía a la persona.
<b>E-Mail Address</b>	SI	Dirección de correo electrónico de la persona. Necesaria para enviar la credencial.
<b>Wallet Address</b>	SI	Dirección de la cartera del perfil Europass de la persona. Necesaria para depositar la credencial en su wallet automáticamente.
<b>Full Address</b>	NO	Dirección completa del domicilio de residencia de la persona.
<b>Country of Residence</b>	NO	País de residencia de la persona. (Campo seleccionable)
<b>Learning Achievements</b>	NO	Informaciones de los logros del aprendizaje de la credencial, referenciados en la hoja 4 ( <b>Achievements</b> ).  (Campo múltiple seleccionable)
<b>Learning Activities</b>	NO	Informaciones de las actividades del aprendizaje de la credencial, referenciados en la hoja 6 ( <b>Activities</b> ).  (Campo múltiple seleccionable)
<b>Learning Entitlements</b>	NO	Información de los derechos de aprendizaje de la credencial, referenciados en la hoja 8 ( <b>Entitlements</b> ). (Campo múltiple seleccionable)
<b>Grade</b>	NO	Evaluaciones o Calificaciones que la persona ha tenido.

Tabla 2.5. Campos de la Hoja Persons de la plantilla Excel EDC.

### 2.4.3. Hoja 3: Organisations

La tercera hoja, denominada “**Organisations**”, recoge los datos referentes a la organización o persona jurídica que va a emitir y sellar la credencial (como nombre legal, identificador legal, etc..). Dentro de esta hoja, tenemos una serie de campos que la organización tendrá que cumplimentar mínimamente. En la tabla 2.6<sup>18</sup>, podemos observar cuales son estos campos y la información que recoge cada uno.

Nombre del campo	Obligatorio	Información
<b>Legal Name</b>	SI	El nombre legal completo de la organización, registrado por las autoridades nacionales. Este nombre solo podrá traducirse a aquellos lenguajes donde exista una traducción oficial.
<b>Common Name</b>	NO	Nombre alternativo que la organización usa normalmente en documentos o en credenciales.
<b>Legal Identifier</b>	SI	El número de identificación oficial de la organización, que ha sido otorgado por una autoridad nacional relevante
<b>Logo</b>	NO	URL pública del logo de la organización en formato png o jpg.
<b>Homepage</b>	NO	Página principal de la organización.
<b>Location Name</b>	NO	Nombre apropiado para situar la organización (Como Dublin, Gent, etc..)
<b>Full Address</b>	NO	Dirección completa de la organización (Mala cesta 12, 21000)

<sup>18</sup> Fuente: Elaboración propia basada en [8].

<b>Country</b>	SI	País de la organización (Campo seleccionable).
<b>E-Mail Address</b>	NO	Dirección de correo electrónico de la organización.
<b>VAT Number Country Code</b>	NO	El Código de País establecido oficialmente en el VAT.
<b>VAT Number digist</b>	NO	Los dígitos que identifican el anterior Código de País. (Normalmente entre 8 -12 dígitos).
<b>Tax/Fiscal Identifier Spatial ID</b>	NO	Identificación del país que forma parte del identificador fiscal oficial.
<b>Tax/Fiscal Identifier Number</b>	NO	Número de identificación fiscal perteneciente al identificador del país anterior.

Tabla 2.6. Campos de la Hoja Organisations de la plantilla Excel EDC.

## 2.4.4. Hoja 4: Achievements

La cuarta hoja, denominada “**Achievements**” recoge la información sobre los logros que se referencian en la credencial. Todos los campos de esta hoja son opcionales. En la tabla 2.7<sup>19</sup>, podemos observar cuales son estos campos y la información que recoge cada uno.

Nombre del campo	Información
<b>Title</b>	Título del logro. En caso de que el tipo de la credencial sea <b>Generic</b> o <b>Qualification Awards</b> , suele ser el mismo título de la credencial.
<b>Identifier Scheme</b>	Nombre del identificador del esquema.
<b>Identifier</b>	Identificador alternativo asignado al logro por la organización que lo concede.
<b>Awarding Date</b>	La fecha en la que se concedió el logro.
<b>Awarding Organisation</b>	Organización que concede el logro. Este campo es obligatorio si se rellenó el campo anterior.
<b>Description</b>	Descripción del logro.
<b>Proven by</b>	Evaluación que demuestra la adquisición de los resultados de aprendizaje que componen el logro. Si procede, facilite primero los detalles en la hoja " <b>Assessments</b> " y, a continuación, seleccione la evaluación que demuestra este logro.
<b>Influenced by</b>	Actividades que han contribuido a la adquisición de los resultados de aprendizaje que componen el logro. Si procede, facilite primero los detalles en la hoja de " <b>Activities</b> " y, a continuación, seleccione la(s) actividad(es) asociada(s) a este logro.

<sup>19</sup> Fuente: Elaboración propia basada en [8].

<b>Entitles Owner to</b>	Derechos que el propietario ha recibido como resultado de este logro. Si procede, facilite primero los detalles en la hoja "Entitlements" y, a continuación, seleccione los derechos asociados a este logro.
<b>Sub-achievements</b>	Unidades de logro más pequeñas que, combinadas, componen este logro. Si procede, introduzca todos los logros pertinentes en esta hoja y, a continuación, seleccione los que lo componen. (Sub-logros).

Tabla 2.7. Campos de la Hoja Achievements de la plantilla Excel EDC.

### 2.4.5. Hoja 5: Learning Outcomes

La quinta hoja, denominada “**Learning Outcomes**”, recoge información sobre todos los conocimientos, habilidades y competencias que sus alumnos han logrado al completar con éxito una oportunidad de aprendizaje. Todos los campos de esta hoja son opcionales. En la tabla 2.8<sup>20</sup>, podemos observar cuales son estos campos y la información que recoge cada uno.

<b>Nombre del campo</b>	<b>Información</b>
<b>Title</b>	Nombre legible y descriptivo para el resultado del aprendizaje.
<b>Learning Outcome Identifier Scheme</b>	Nombre del esquema del identificador. (Ejemplo LOID-73a)
<b>Learning Outcome Identifier</b>	Identificador alternativo del resultado de aprendizaje.

<sup>20</sup> Fuente: Elaboración propia basada en [8].

<b>Description</b>	Descripción de lo que el alumno puede hacer como resultado del aprendizaje, con una indicación del nivel de logro, y las condiciones o contexto en el que se puede realizar.
<b>Learning Outcome Type</b>	Tipo de resultado de aprendizaje. Dos tipos: <b>Knowledge</b> (Conocimiento) o <b>Skill</b> (Habilidad).
<b>Reusability Level</b>	Nivel de reutilización. Cuatro tipos: <b>Transversal</b> (Level 1) <b>Cross-Sector</b> (Level 2) <b>Sector-Specific</b> (Level 3) <b>Occupation-Specific</b> (Level 4)
<b>Related ESCO Skill URL</b>	URL directa a la habilidad de la ESCO referenciada.
<b>Related Skill Framework URI</b>	La identificación del vocabulario regulado (la lista de códigos). (por ejemplo, un URI)
<b>Related Skill Title</b>	el título de la habilidad referenciada (no de la UNESCO).
<b>Related Skill URL</b>	Un enlace a una habilidad relacionada o el nivel de una habilidad relacionada en un marco de habilidades (excepto ESCO).

Tabla 2.8. Campos de la Hoja Learning Outcomes de la plantilla Excel EDC.

## 2.4.6. Hoja 6: Activities

La sexta hoja, denominada “**Activities**” recoge la información sobre las actividades que se referencian en la credencial. Todos los campos de esta hoja son opcionales. En la tabla 2.9<sup>21</sup>, podemos observar cuales son estos campos y la información que recoge cada uno.

<sup>21</sup> Fuente: Elaboración propia basada en [8].

Nombre del campo	Información
<b>Title</b>	Título de la actividad de aprendizaje.
<b>Learning Activity Identifier Scheme</b>	Nombre del esquema que identifica la actividad.
<b>Learning Activity Identifier</b>	Identificador alternativo de la actividad de aprendizaje asignado por la organización que dirige la actividad.
<b>Description</b>	Texto descriptivo de la actividad de aprendizaje.
<b>Start-Date</b>	Fecha en la que el alumno comenzó la actividad. Formato: dd/mm/aaaa
<b>End-Date</b>	Fecha en la que el alumno finalizó la actividad. Formato: dd/mm/aaaa
<b>Workload in Hours</b>	Número real de horas que el alumno ha dedicado a la actividad. Solo se aceptan números enteros positivos (sin decimales).  Esto incluiría: Número estimado de horas en clase, en trabajo en grupo, en prácticas, así como las horas dedicadas al estudio no presencial.
<b>Directed by</b>	Organización que dirigió la actividad. Los datos de esta deben estar también introducidos en la hoja 3 ( <b>Organisations</b> ).
<b>Location Name</b>	Lugar donde se desarrolló la actividad.
<b>Full Address</b>	Dirección completa donde se desarrolló la actividad.

<b>Country</b>	País donde se desarrolla la actividad. Este campo es obligatorio si se estableció el campo anterior.
<b>Sub-Activities</b>	Actividades más pequeñas que, combinadas, conforman esta actividad. Estas actividades deben estar especificadas previamente en esta misma hoja.
<b>Specification Title</b>	Título de la especificación de la actividad de aprendizaje.
<b>Learning Activity Specification Identifier Scheme</b>	Nombre del esquema que identifica la especificación.
<b>Learning Activity Specification Identifier</b>	Un identificador alternativo de la Actividad de Aprendizaje, tal y como lo asignó la organización que diseñó la especificación.
<b>Specification Description</b>	Texto descriptivo de la especificación de la actividad de aprendizaje.
<b>Homepage</b>	Página web que describe la especificación de la actividad de aprendizaje.
<b>Learning Activity Type</b>	<p>Tipo de actividad de aprendizaje:</p> <ul style="list-style-type: none"> <li>-<b>educational programme</b></li> <li>-<b>workshop, seminar or conference</b></li> <li>- <b>internship</b></li> <li>- <b>apprenticeship</b></li> <li>- <b>self-motivated study</b></li> <li>- <b>volunteering</b></li> <li>- <b>job experience</b></li> <li>- <b>classroom coursework</b></li> <li>- <b>lab / simulation / practice coursework</b></li> <li>- <b>e-learning coursework</b></li> <li>- <b>research</b></li> </ul>



<p><b>Mode of Learning</b></p>	<p>Modo de aprendizaje y/o evaluación:</p> <ul style="list-style-type: none"> <li>- <b>Online</b></li> <li>- <b>Presential</b></li> <li>- <b>Blended</b></li> <li>- <b>Workbased</b></li> <li>- <b>Project based</b></li> <li>- <b>Research-Lab based</b></li> </ul>
<p><b>Volume of Learning</b></p>	<p>El número estimado de horas que se espera que el alumno dedique a la actividad. Esto incluiría el número teórico de horas en clase, en el trabajo en grupo, en las prácticas, así como las horas dedicadas al estudio auto-motivado. Sólo se aceptan números enteros positivos (sin decimales).</p>

Tabla 2.9. Campos de la Hoja Activities de la plantilla Excel EDC.

### 2.4.7. Hoja 7: Assessments

La séptima hoja, denominada “**Assessments**” recoge la información sobre las valoraciones que se referencian en la credencial. Todos los campos de esta hoja son opcionales. En la tabla 2.10<sup>22</sup>, podemos observar cuales son estos campos y la información que recoge cada uno.

<p><b>Nombre del campo</b></p>	<p><b>Información</b></p>
<p><b>Title</b></p>	<p>El título de la evaluación.</p>

<sup>22</sup> Fuente: Elaboración propia basada en [8].

<b>Assessment Identifier Scheme</b>	Nombre del esquema que identifica la evaluación.
<b>Assessment Identifier</b>	Identificador alternativo asignado a la evaluación por la organización que la califica.
<b>Description</b>	Texto descriptivo sobre la evaluación.
<b>Assessed by</b>	Organización que ha realizado la evaluación. La información de la organización debe estar recogida en la hoja 3 ( <b>Organisations</b> ).
<b>Method of assessment, supervision and ID verification</b>	Método de supervisión de la evaluación y verificación: <ul style="list-style-type: none"> <li>- <b>Unsupervised with ID Verification</b></li> <li>- <b>Unsupervised without ID Verification</b></li> <li>- <b>Supervised with ID Verification</b></li> <li>- <b>Supervised without ID Verification</b></li> </ul>
<b>Assessment Date</b>	Fecha de concesión de la nota. Formato: dd/mm/aaaa
<b>Sub-Assessments</b>	Evaluaciones más pequeñas, que cuando se combinan conforman esta evaluación. Si procede, añada primero nuevos elementos en esta hoja y, a continuación, seleccione una evaluación relacionada que haya influido en el resultado de esta evaluación.

Tabla 2.10. Campos de la Hoja Assessments de la plantilla Excel EDC.

## 2.4.8. Hoja 8: Entitlements

La octava hoja, denominada “**Entitlements**” recoge la información sobre los derechos que otorga la credencial. Todos los campos de esta hoja son opcionales. En la tabla 2.11<sup>23</sup>, podemos observar cuales son estos campos y la información que recoge cada uno.

<sup>23</sup> Fuente: Elaboración propia basada en [8].

Nombre del campo	Información
<b>Title</b>	Título del derecho.
<b>Entitlement Identifier Scheme</b>	Nombre del esquema que identifica el derecho
<b>Identifier</b>	Identificador alternativo del derecho.
<b>Date of Issue</b>	Fecha de emisión del derecho. Formato: dd/mm/aaaa
<b>Expiry Date</b>	Fecha de caducidad del derecho. Formato: dd/mm/aaaa
<b>Description</b>	Descripción de los derechos específicos que ha adquirido el titular de la credencial.
<b>Sub-Entitlements</b>	Derechos más pequeños, que cuando se combinan conforman este derecho. Estos derechos deberán estar establecidos previamente en esta misma hoja.

Tabla 2.11. Campos de la Hoja Entitlements de la plantilla Excel EDC.

## 2.5. Proceso de subida, validación, sellado, emisión y envío de las credenciales EDCL.

Tras haber conseguido nuestro **QES (Qualification Electronic Seal)** y haber cumplimentado la plantilla **Excel EDC**. Debemos proceder con los pasos de testeo del sello, subida y validación del lote de credenciales, previsualización de las credenciales, sellado y envío de las mismas mediante correo y a la **Wallet** del alumno.

### 2.5.1. Testeo del Sello electrónico.

En primer lugar, deberemos comprobar que nuestro **e-Seal** es válido para la emisión de credenciales mediante **EDCI**. Para poder llevar a cabo el testeo del sello, necesitaremos tener instalado, en nuestro equipo o en el equipo donde se vaya a probar el sello, **NexU**<sup>24</sup>. Previamente, el equipo debe tener en su cartera o almacén de llaves o firmas, el **e-Seal** anteriormente mencionado.

**NexU** [7] es una herramienta de firma remota de código abierto, multiplataforma multinavegador de código abierto con el propósito de comunicarse con tarjetas inteligentes. Dentro de sus características destacamos:

- Desarrollada en **Java**.
- La herramienta se comunica con la tarjeta inteligente a través del lenguaje **Javascript**.
- Basado en **SD-DSS (Digital Signature Service)**, la implementación de referencia de la UE para la firma electrónica.

Una vez instalada la herramienta y lanzada, deberemos apreciar un icono circular con una llave en su interior, en la parte inferior derecha de nuestro monitor, para asegurarnos que está funcionando correctamente. En la figura 2.5<sup>25</sup>, podemos observar este icono.

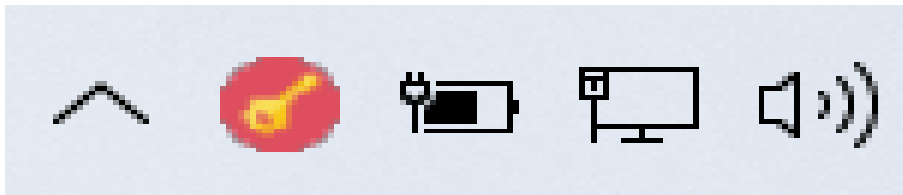


Figura 2.5. NextU en ejecución.

Tras esto, procederemos a probar nuestro sello, accedemos al **EDCI Issuer** de la plataforma Europass y accedemos al apartado de “**Pruebe su sello electrónico**”. En la figura 2.6<sup>26</sup>, podemos observar la sección donde podemos probar nuestro sello electrónico dentro del **EDCI Issuer**.

<sup>24</sup> Herramienta NexU accesible en: <https://github.com/nowina-solutions/nexu/releases/download/nexu-1.22/nexu-bundle-1.22.zip>

<sup>25</sup> Fuente: Elaboración propia.

<sup>26</sup> Fuente: Elaboración propia.

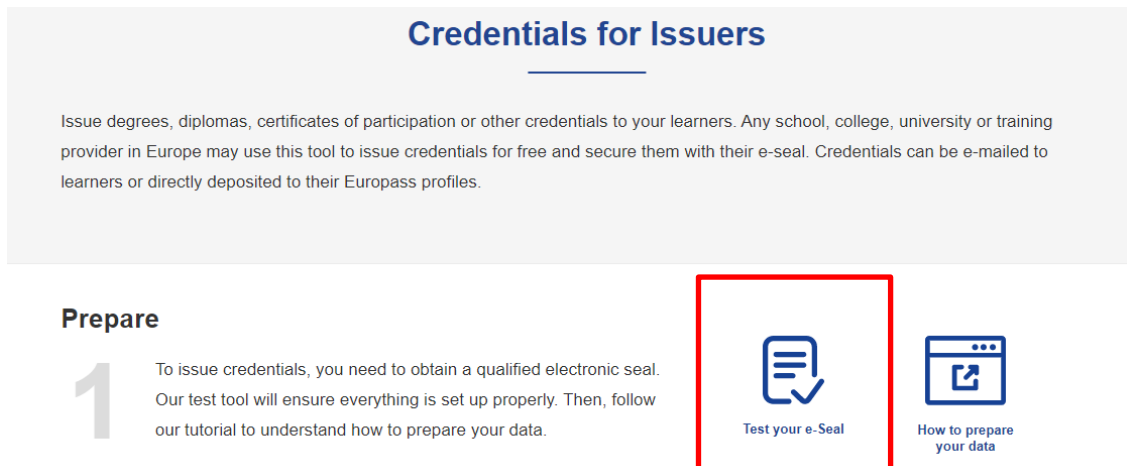


Figura 2.6. Apartado para probar nuestro sello electrónico.

Según accedemos, si tenemos activo **NextU**, se establecerá la conexión, se nos pedirá el sello que vamos a probar y finalmente, chequeará mediante una serie de credenciales internas, que sella la plataforma automáticamente, si nuestro sello es válido. En la figura 2.7<sup>27</sup>, podemos observar los pasos del testeo del sello.

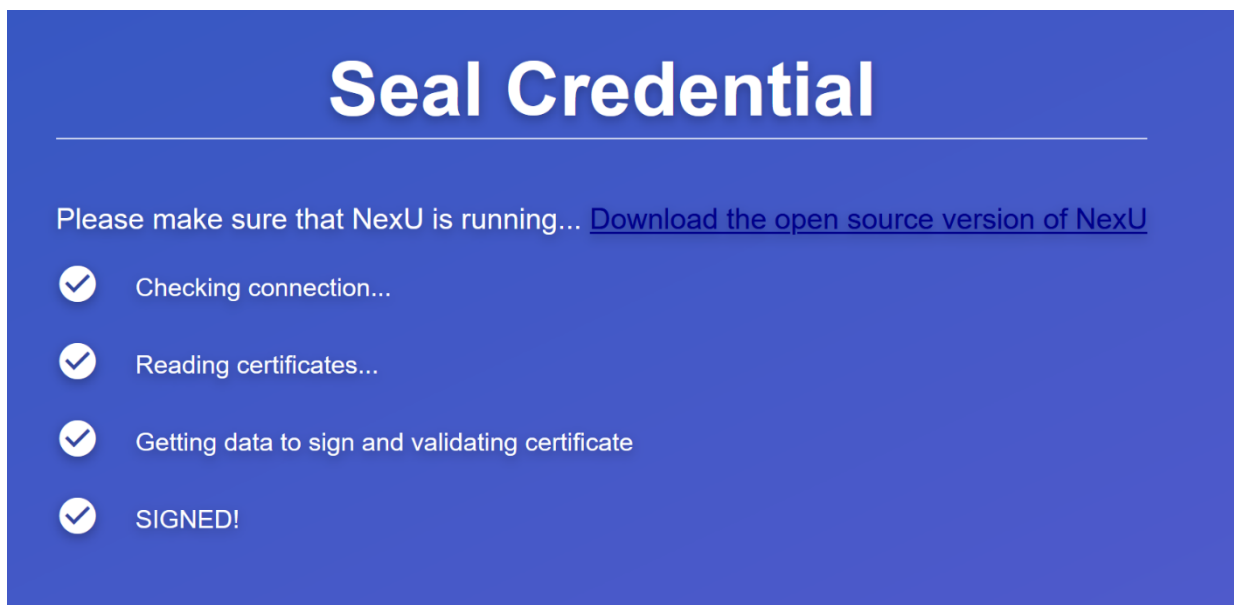


Figura 2.7. Probando nuestro e-Seal mediante EDCI Issuer.

<sup>27</sup> Fuente: Elaboración propia.

## 2.5.2. Subida y validación del lote de credenciales.

Tras verificar que nuestro **e-Seal** es válido. Podemos subir nuestro lote de credenciales contenidos en el fichero **Excel EDC**. Para ello, accedemos al apartado “**Upload**” del **EDCI Issuer**. En la figura 2.8<sup>28</sup>, podemos observar el apartado de subida del lote de credenciales.

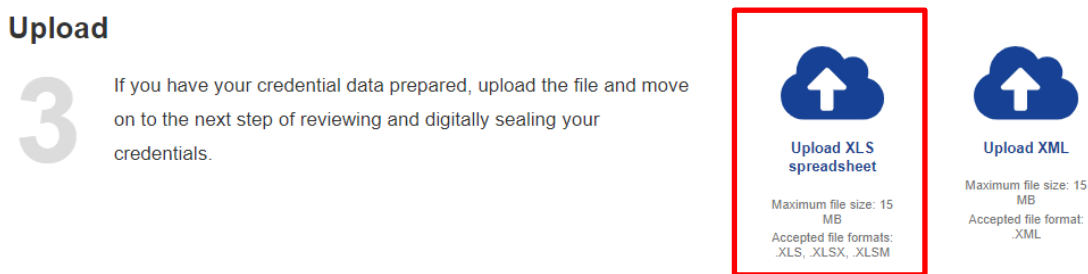


Figura 2.8. Apartado para subir nuestro lote de credenciales.

Tras esto, seleccionamos nuestro archivo **Excel EDC** con las credenciales que queremos emitir. Tras subirla obtendremos si el formato de nuestras credenciales es válido o no. En caso de no ser válidas, se debería revisar el contenido del **Excel EDC** y modificar el fallo. En la figura 2.9<sup>29</sup>, podemos observar una credencial con un formato correcto y otra con un formato inválido.

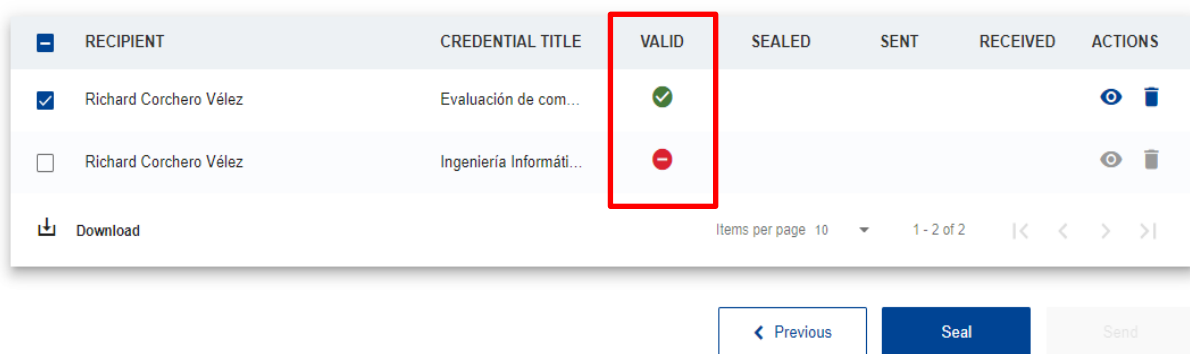


Figura 2.9. Validación del lote de credenciales mediante EDCI Issuer.

<sup>28</sup> Fuente: Elaboración propia.

<sup>29</sup> Fuente: Elaboración propia.

### 2.5.3. Previsualización de las credenciales digitales.

Una vez tengamos las credenciales con un formato válido, podemos previsualizar la credencial y ver su representación visual. Esta credencial posteriormente también podrá obtenerse en formato PDF e incluso imprimirse, si el usuario de destino, lo considera oportuno. Podemos previsualizar la credencial digital pulsando sobre el icono del “ojo” situado en el apartado “**Actions**”. En la figura 2.10<sup>30</sup>, podemos observar la ubicación del icono de previsualización de las credenciales digitales.

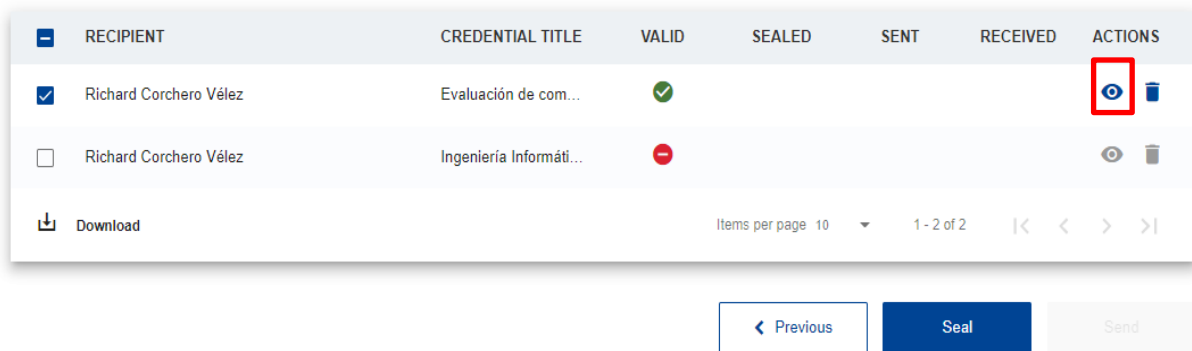


Figura 2.10. Previsualizando la credencial digital I.

Por otro lado, en la figura 2.11<sup>31</sup>, podemos observar la previsualización, por defecto, de la credencial digital que va a ser emitida.

<sup>30</sup> Fuente: Elaboración propia.

<sup>31</sup> Fuente: Elaboración propia.

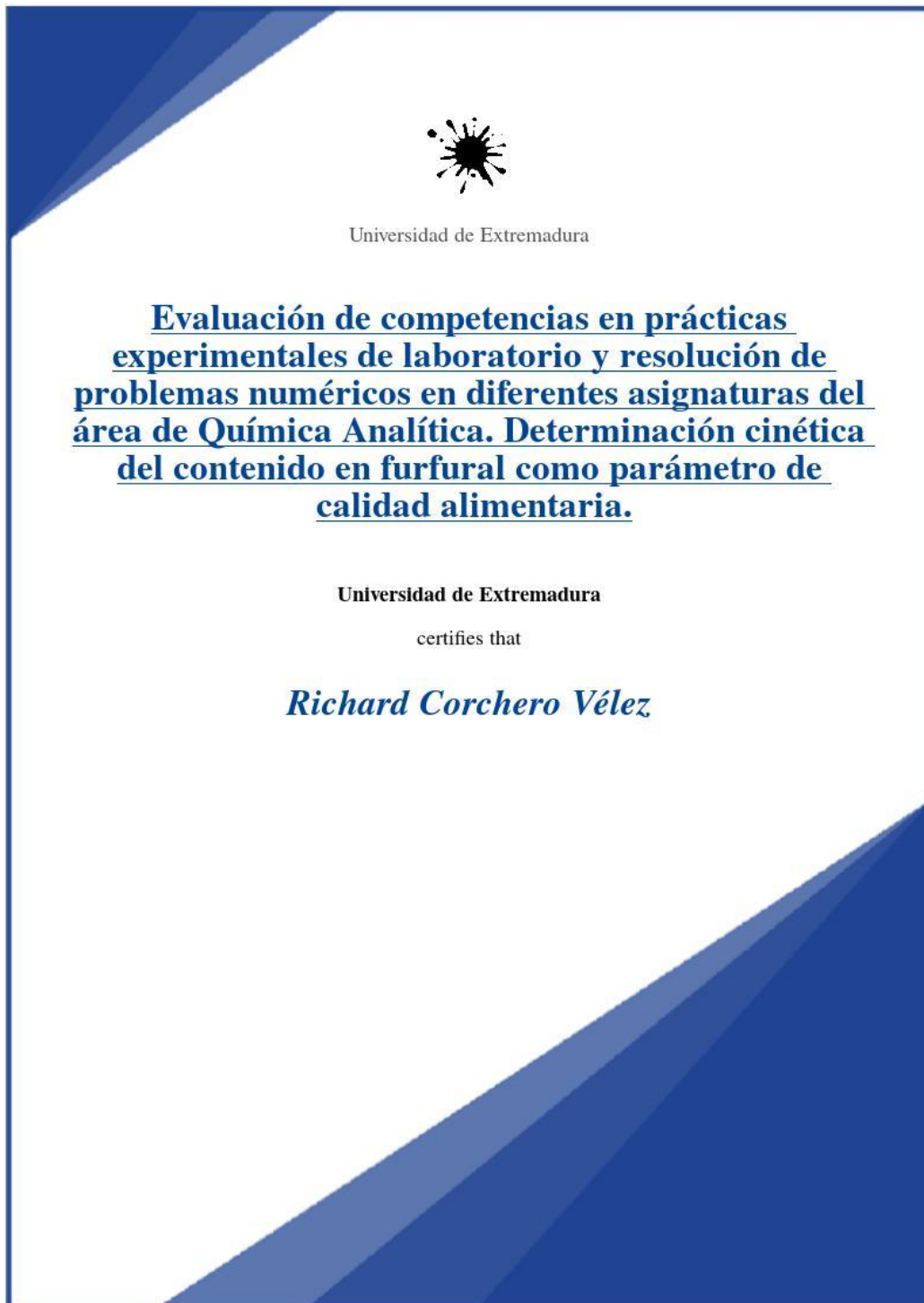


Figura 2.11. Previsualizando la credencial digital II.



## 2.5.4. Sello de las credenciales digitales mediante e-Seal.

Una vez la credencial tiene un formato válido y hemos observado, mediante la previsualización de esta, que tiene el estilo visual que deseamos, podemos seguir con el sellado de las credenciales. En la figura 2.12<sup>32</sup>, podemos ver como seleccionar las credenciales y pasar al proceso de sellado.

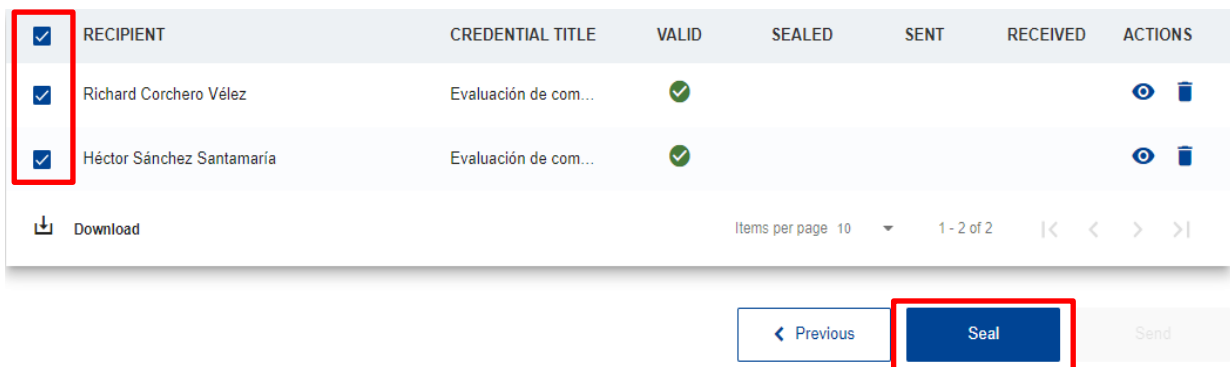


Figura 2.12. Selección y sellado de las credenciales.

Para ello, seleccionamos las credenciales y pulsamos sobre el botón “**Seal**” (Importante tener arrancada la herramienta **NextU** previamente). Posteriormente, se nos pedirá que seleccionemos nuestro **e-Seal**. Finalmente, si no hubo ningún problema durante el proceso de sellado, deberíamos obtener un “**tic verde**” en la casilla “**Sealed**”. En la figura 2.13<sup>33</sup>, podemos observar el resultado final del sellado.

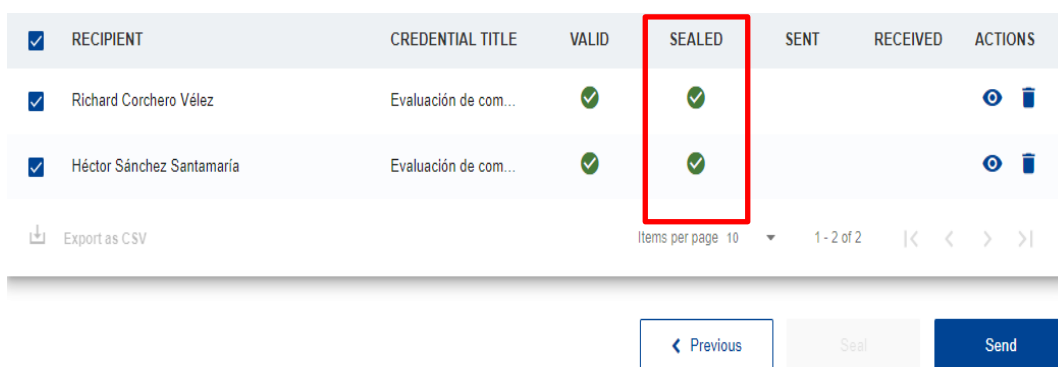


Figura 2.13. Resultado final del sellado de las credenciales.

<sup>32</sup> Fuente: Elaboración propia.

<sup>33</sup> Fuente: Elaboración propia.

## 2.5.5. Emisión y envío de las credenciales por correo electrónico y al wallet del alumno.

Por último, nos queda enviar las credenciales digitales validadas y selladas a sus destinatarios. Para ello solamente, debemos hacer clic sobre el botón “**Sent**”. En la figura 2.14<sup>34</sup>, podemos observar la opción para realizar el envío de las credenciales mediante EDCI Issuer.

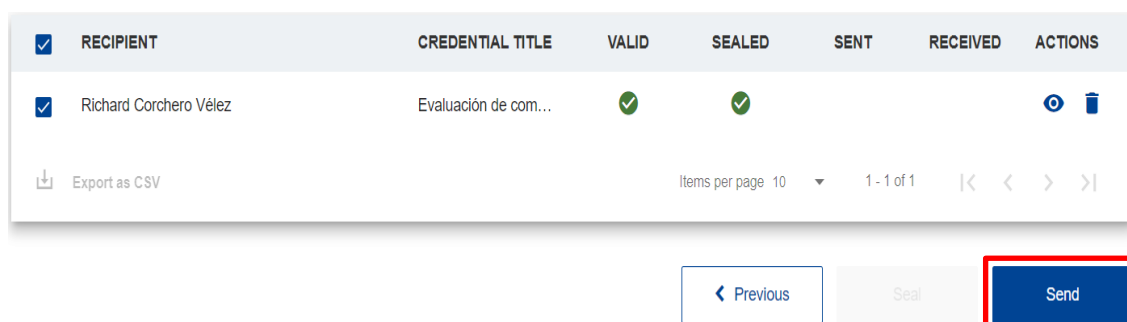


Figura 2.14. Envío de las credenciales digitales selladas en Europass I.

Si la operación se realizó exitosamente, deberíamos visualizar un “**tic verde**” en el campo “**Sent**” y “**Received**”. En la figura 2.15<sup>35</sup>, podemos observar el resultado exitoso de la operación.

RECIPIENT	CREDENTIAL TITLE	VALID	SEALED	SENT	RECEIVED	ACTIONS
<input checked="" type="checkbox"/> D. RICHARD CORCHERO VÉLEZ	JORNADA DE FOR...	✓	✓	✓	✓	
<input checked="" type="checkbox"/> D. HÉCTOR SÁNCHEZ SANTAMARÍA	JORNADA DE FOR...	✓	✓	✓	✓	

Export as CSV      Items per page: 10      1 - 2 of 2      << < > >>

Figura 2.15. Envío de las credenciales digitales selladas en Europass II.

Tras esto, habremos recibido un correo electrónico, con un enlace para ver nuestra credencial en nuestra cartera de la plataforma Europass, y la credencial en nuestra cartera o wallet de Europass. En la figura 2.16<sup>36</sup>, podemos observar la credencial en la wallet del usuario.

<sup>34</sup> Fuente: Elaboración propia.

<sup>35</sup> Fuente: Elaboración propia.

<sup>36</sup> Fuente: Elaboración propia.

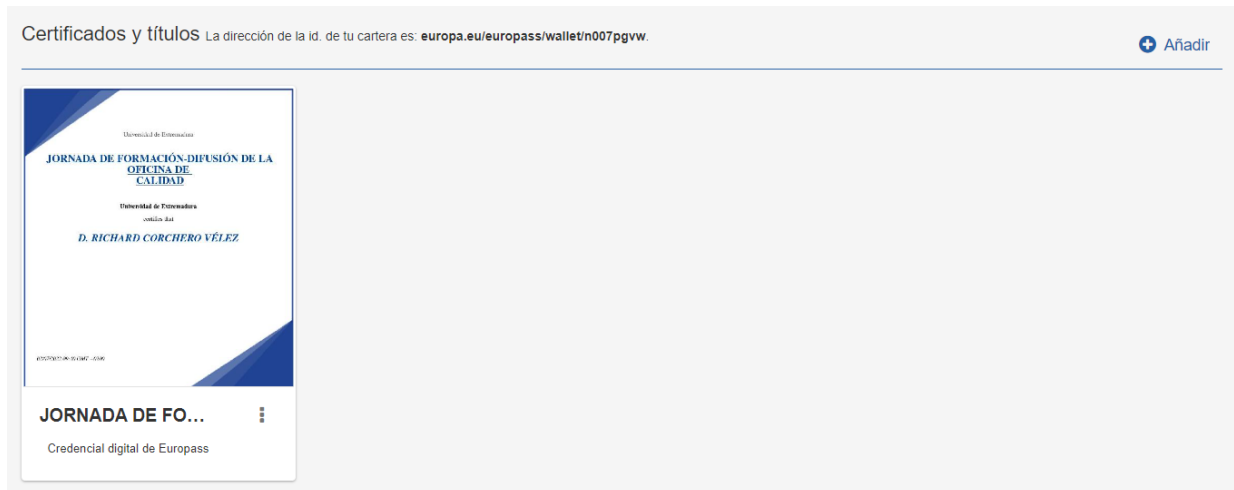


Figura 2.16. Credencial Digital sellada depositada en la cartera o wallet de un perfil Europass.



# Capítulo 3

## Metodología

### Contenidos

---

<b>3.1. Metodología Iterativa Incremental</b> .....	<b>41</b>
3.1.1 Introducción .....	<b>41</b>
3.1.2 Etapas .....	<b>42</b>
3.1.3 Beneficios e Inconvenientes.....	<b>44</b>
<b>3.2. Metodología Orientada a Objetos</b> .....	<b>44</b>

---

Este tercer capítulo se detallará la metodología aplicada en este TFG. Podemos destacar dos tipos de metodologías empleadas:

- **Metodología Iterativa Incremental**, como marco para estructurar, planificar y controlar el desarrollo del proyecto.
- **Metodología Orientada a Objetos**, necesaria para el desarrollo del aplicativo o servicio.

En la sección 3.1 se detallará el concepto de metodología iterativa incremental y sus etapas.

En la sección 3.2 se describe el concepto de metodología orientada a objetos y sus fases.

### 3.1. Metodología Iterativa Incremental

Para cubrir los objetivos que pretende alcanzar este TFG, se aplica una metodología iterativa incremental, como marco para estructurar, planificar y controlar el desarrollo.

#### 3.1.1 Introducción

La metodología iterativa incremental [11] divide el proyecto en un conjunto de bloques temporales llamados “**iteraciones**”.

Cada una de estas iteraciones, puede entenderse como subproyectos que forman el proyecto principal. En cada iteración, se repite un proceso de trabajo similar para proporcionar un resultado completo, generando u obteniendo un entregable o conjunto de entregables funcionales para el cliente. De este modo, se genera incrementalmente el producto, dando la posibilidad al cliente de obtener beneficios del proyecto de manera incremental.

Para poder ejecutar esta metodología, cada requisito debe completarse en una única iteración, finalizando todas las tareas establecidas en la iteración e incluyendo pruebas e información por parte del equipo de desarrollo. En la figura 3.1, podemos observar una ejemplificación de la metodología iterativa incremental.

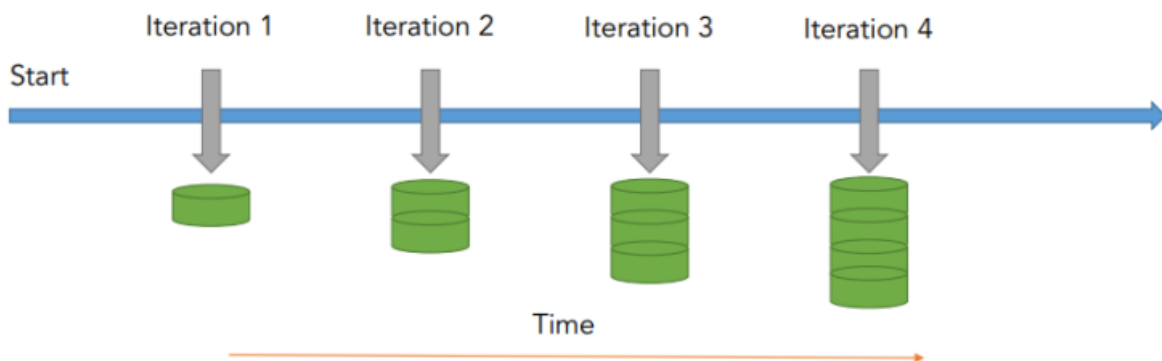


Figura 3.1. Ejemplificación de la metodología iterativa incremental.

De esta manera, el equipo de desarrollo va evolucionando el producto a partir de los resultados de la iteración anterior, mejorando o añadiendo nuevos requisitos. Un aspecto fundamental para guiar el desarrollo iterativo incremental es la priorización de los objetivos o requisitos en función del valor que aportan al cliente.

### 3.1.2 Etapas

Dentro del modelo iterativo incremental, podemos distinguir un conjunto de siete etapas [12] que debemos tener en cuenta:

- **Primera fase (Identificación de requisitos)**, durante esta fase se recogen los objetivos principales y específicos que persigue el proyecto.
- **Segunda fase (Definición de tareas e iteraciones)**, durante esta fase se establecen una lista de tareas, que posteriormente, se asignan o agrupan en las diferentes iteraciones que tendrá el proyecto.

- **Tercera fase (Diseño de los incrementos)**, durante esta fase se define la evolución del producto (incremento) en cada una de las iteraciones. Cada resultado final de una iteración deberá superar a la anterior.
- **Cuarta fase (Desarrollo del incremento)**, durante esta fase se realizan las tareas e incrementos anteriormente establecidos.
- **Quinta fase (Validación de incrementos)**, tras finalizar cada iteración se deben dar por buenos los incrementos que han generado. Si no hemos obtenidos los resultados esperados deberemos volver al inicio de la iteración y encontrar el problema.
- **Sexta fase (Integración de incrementos)**, tras haber validado los incrementos, estos forman la denominada “**línea incremental**” o evolución del proyecto en su conjunto. Cada incremento ha contribuido al resultado final.
- **Séptima fase (Entrega del producto)**, tras validar el producto en su conjunto, se procede con su entrega final al cliente.

En la figura 3.2, podemos observar un esquema del flujo de las etapas de la metodología iterativa incremental.

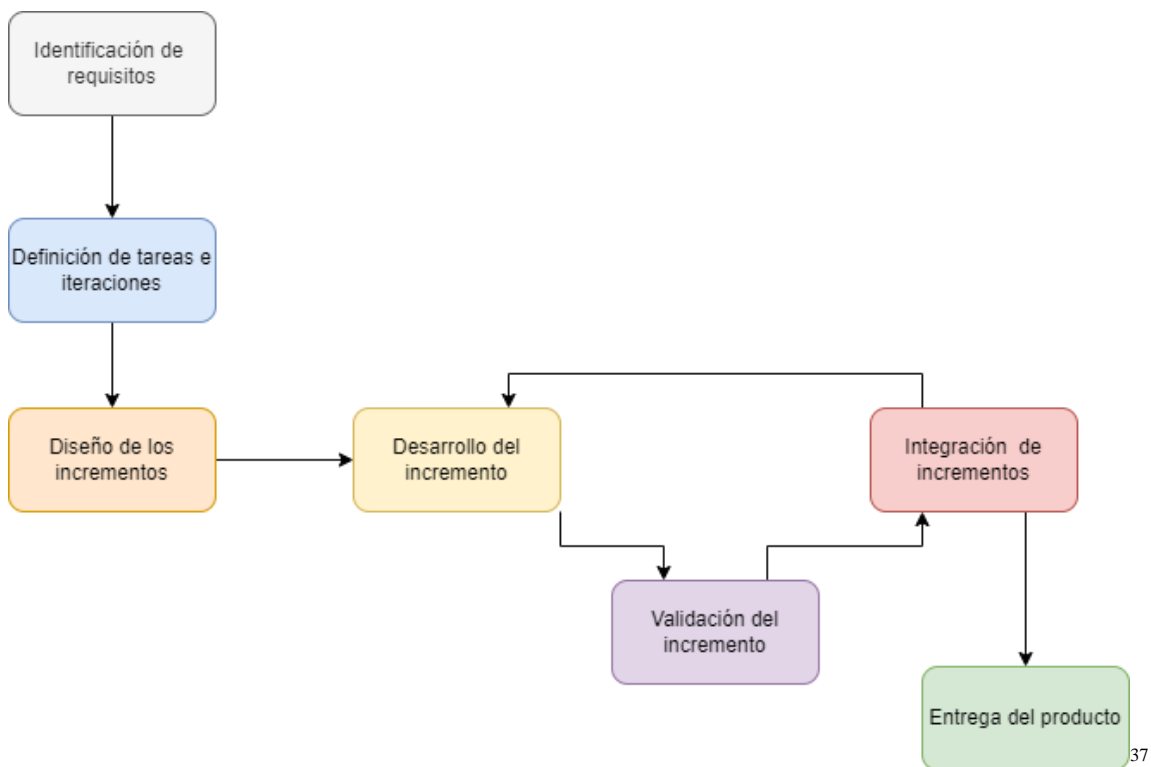


Figura 3.2. Etapas de la metodología iterativa incremental.

<sup>37</sup> Fuente: Elaboración original basada en [12]

### 3.1.3 Beneficios e Inconvenientes

La metodología iterativa incremental nos aporta una serie de ventajas y desventajas [11] al desarrollo de nuestro proyecto. Destacamos las principales ventajas:

- Capacidad para poder gestionar las expectativas del cliente de forma regular, tomando decisiones en cada iteración (realmente útil, cuando el cliente no sabe lo que necesita, cambios a corto plazo, etc.).
- Capacidad para comenzar el proyecto sin necesidad de conocer todos los requisitos al inicio (conocer los requisitos de las primeras iteraciones e ir completando el resto de los requisitos en las iteraciones consecutivas).
- El cliente obtiene de manera regular versiones funcionales y usables desde las primeras iteraciones.
- Mayor flexibilidad antes los cambios del proyecto.
- El cliente como máximo puede perder los recursos dedicados a una iteración, no los de todo el proyecto o iteraciones anteriores [11].
- Minimiza el número de errores y aumenta la calidad del producto final.

No obstante, las principales restricciones son:

- La disponibilidad del cliente debe ser alta, debido a que participa de manera continua en el inicio [11] y finalización de cada una de las fases (muchas veces el cliente no tiene tiempo o no quiere involucrarse en el desarrollo).
- La relación con el cliente ha de estar basada en los principios de colaboración y ganar/ganar [11].
- Cada iteración debe dar como resultado requisitos terminados.
- Técnicas y herramientas que nos permitan diseñar cada iteración de manera incremental (de modo que, de una iteración a otra, el incremento sea inabarcable o requiera un esfuerzo adicional).

## 3.2. Metodología Orientada a Objetos

Para el desarrollo de cada una de las iteraciones del proyecto, definidas mediante la metodología iterativa incremental, se empleará la metodología orientada a objetos.



La metodología orientada a objetos [13] es un enfoque de desarrollo de sistemas sobre una base de componentes (objetos y clases de los lenguajes de programación orientados a objetos como **Java**), que fomenta la reutilización efectiva de los componentes software existentes y facilita el intercambio entre componentes del sistema. La principal premisa de esta metodología es representar todo elemento del mundo real como un objeto.

Esta metodología exige al analista que determine cuáles son los objetos del sistema, así como sus responsabilidades, relaciones con otros objetos y comportamiento a largo plazo. Para llevar a cabo, el modelo y diseño de nuestro sistema se ha empleado el estándar internacional **UML (Unified Modeling Language)** [14].

Podemos establecer 4 etapas [15] principales dentro de esta metodología:

- **Análisis**, durante esta fase, se analizan y entienden los requisitos del sistema.
- **Diseño**, durante esta fase, se modelan y diseñan los componentes del sistema que tienen que dar solución a los requisitos de la fase anterior.
- **Implementación o codificación**, durante esta fase, se traduce el diseño, de la fase anterior, a través de una aplicación o lenguaje de programación orientado a objetos, adaptando la solución a un entorno concreto.
- **Pruebas**, durante esta fase, se realizan pruebas sobre los diferentes componentes software implementados y se valida si cumplen los requisitos del sistema.



# Capítulo 4

## Análisis del sistema

### Contenidos

---

<b>4.1. Análisis de viabilidad .....</b>	<b>47</b>
4.1.1 Viabilidad técnica.....	48
4.1.2 Viabilidad económica.....	48
4.1.3 Viabilidad operacional .....	48
<b>4.2. Planificación estimada .....</b>	<b>49</b>
4.2.1 Estructura de desglose de trabajo (EDT).....	49
4.2.2 Diccionario de la EDT.....	51
4.2.3 Cronograma.....	55

---

En este capítulo, se concretará el análisis del sistema que va a ser desarrollado, determinando su viabilidad, planificación y principales características funcionales o de comportamiento.

En la sección 4.1, se expondrá un estudio de viabilidad sobre el apartado técnico, operacional y económico, así como, su estudio de oportunidad.

En la sección 4.2, se mostrará la planificación del proyecto, realizando una **EDT (Estructura de desglose de trabajo)**, que nos permita dividir el proyecto en diferentes paquetes de trabajo, y posteriormente, estableciendo el cronograma temporal.

### 4.1. Análisis de viabilidad

En este apartado estudiaremos la viabilidad del proyecto desde tres perspectivas: **técnica, económica y operacional.**

### 4.1.1 Viabilidad técnica

Para poder llevar a cabo la consecución de esta TFG, es necesario tener conocimientos técnicos sobre:

- **Lenguajes de programación orientados a objetos**, donde destacamos el principal uso del lenguaje **Java**.
- **Lenguajes de scripting**, donde destacamos uso principal de **Javascript** y **Microsoft Visual Basic Script**.
- **Tecnologías web**, como **HTML** y **CSS**.
- **Servicios web e infraestructura cliente – servidor**, necesario para el despliegue de nuestra **API REST**.
- **Formato para intercambio de datos**, como ficheros **JSON**.

Todas estas tecnologías son bien conocidas y poseen gran cantidad de documentación, facilitando la resolución de problemas y cuestiones. Debido a todo lo anterior, podemos concluir que desde el punto de vista técnico es viable.

### 4.1.2 Viabilidad económica

Para la realización de este TFG, todas las tecnologías software serán gratuitas y por tanto no supondrá ningún tipo de coste. Los principales costes podrían darse en el despliegue y mantenimiento del servicio API REST en un entorno real. Debido a lo anterior, podemos concluir que el proyecto es viable desde el punto de vista económico.

### 4.1.3 Viabilidad operacional

Dentro de nuestro proyecto, el usuario final son las instituciones que quieran automatizar el proceso de generación de los lotes de credenciales digitales Europass mediante el uso de nuestra **API-REST**. Las instituciones se comunican con nuestro servicio, mediante solicitudes **HTTP** para interactuar con nuestro servidor y realizar operaciones **POST**.

Las organizaciones deberán enviar como cuerpo de esa operación **POST**, un archivo **JSON** (Con un formato específico), que nuestro servidor pueda entender y procesar, generando, finalmente, una nueva plantilla **Excel EDC** con la información de las credenciales a sellar.

Las organizaciones generarán el fichero **JSON** a partir de sus propios sistemas de información, los cuales, tendrán que modificar o ampliar para poder emitir credenciales sobre **EDCI**.

La creación de una **API REST** es vital para todo el desarrollo del proyecto, debido a que, para generar las credenciales en **EDCI**, necesitamos partir de una plantilla específica (ofrecida por la plataforma Europass) y que ha sido reprogramada en ciertos aspectos para poder automatizar el proceso de generación. La plantilla se encuentra en el servidor y es utilizada en todas las peticiones **HTTP** que se realizan sobre el mismo.

Debido a todo lo anterior, podemos concluir que el proyecto también es viable desde el punto de vista operacional.

## 4.2. Planificación estimada

En esta sección se mostrará:

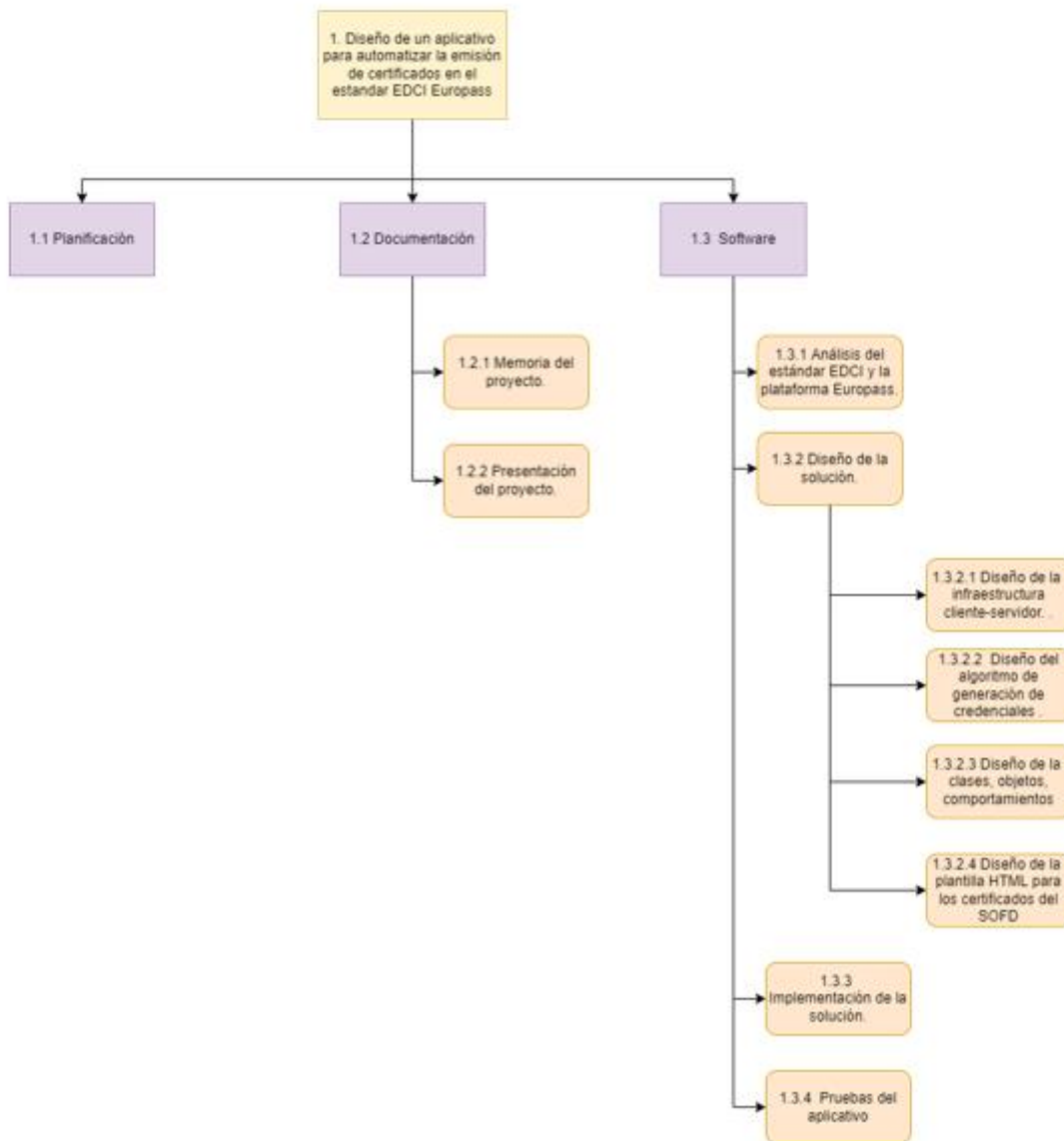
- **EDT** y paquetes de trabajo en los que se ha dividido el proyecto.
- Tareas asociadas a cada uno de los paquetes de trabajo.
- Cronograma.

### 4.2.1 Estructura de desglose de trabajo (EDT)

La **EDT** [16] nos permite descomponer jerárquicamente el alcance total del trabajo que debe llevar a cabo el equipo de proyecto para alcanzar los objetivos del mismo y crear los entregables requeridos. Esta jerarquía está formada por niveles, siendo el nivel más bajo conocido como “**paquete de trabajo**”.

Un paquete de trabajo [16] es un producto entregable o componente del trabajo del proyecto, que cuenta con el nivel de detalle suficiente para ser estimado en términos de alcance, tiempo y costo. A partir de estos paquetes se obtienen las diferentes tareas que lo forma.

En la figura 4.1, podemos observar la estructura de desglose de trabajo de nuestro proyecto.



38

Figura 4.1. Estructura de desglose de trabajo.

<sup>38</sup> Fuente: Elaboración propia.

## 4.2.2 Diccionario de la EDT

Toda EDT debe incluir un “**Diccionario EDT**”. Este diccionario recoge una descripción precisa de los componentes de la EDT. En la tabla 4.1<sup>39</sup>, podemos observar el diccionario EDT referente a los componentes de la EDT de nuestro proyecto.

Identificador	Nombre	Descripción
1.1	Planificación	En este paquete de trabajo se lleva a cabo una planificación estimada del proyecto a desarrollar.
1.2	Documentación	Componente que integra todo el análisis y diseño previo del sistema, redacción de la memoria y presentación de esta.
1.2.1	Memoria del proyecto	En este paquete de trabajo se lleva a cabo la redacción de la memoria de nuestro TFG, así como las revisiones de este.
1.2.2	Presentación del proyecto	En este paquete de trabajo se lleva a cabo la elaboración de la presentación del TFG, así como las horas de ensayo de este.
1.3	Software	Este componente engloba todo el desarrollo software del proyecto, desde el análisis previo del sistema hasta su fase de pruebas y validación.

<sup>39</sup> Fuente: Elaboración propia.

<b>1.3.1</b>	Análisis del estándar EDCI y la plataforma Europass.	En este paquete de trabajo, se recoge todo el estado del arte del proyecto, así como, componentes que sostienen y estructuran EDCI y la plataforma Europass.
<b>1.3.2</b>	Diseño de la solución	Este componente agrupa todas las cuestiones referidas al diseño de la solución. Desde su infraestructura hasta las tecnologías empleadas.
<b>1.3.2.1</b>	Diseño de la infraestructura cliente-servidor.	En este paquete de trabajo, se recogen el estudio y diseño de la infraestructura cliente – servidor a desarrollar.
<b>1.3.2.2</b>	Diseño del algoritmo de generación de credenciales.	En este paquete de trabajo, se recogen todo el estudio de las tecnologías y diseño de estas para poder llevar a cabo la elaboración del algoritmo.
<b>1.3.2.3</b>	Diseño de las clases, objetos y comportamientos.	En este paquete de trabajo, se recoge el estudio y diseño del modelado de los datos que vamos a utilizar en nuestro servicio, así como el comportamiento de este,
<b>1.3.2.4</b>	Diseño de la plantilla HTML para los certificados del SOFD.	En este paquete de trabajo se recoge el diseño de la plantilla HTML para el formato visual de las credenciales a emitir por el SOFD (Servicio de Orientación y Formación Docente) de la Universidad de Extremadura.



<b>1.3.3</b>	Implementación de la solución.	En este paquete de trabajo se recogen todos los aspectos de la implementación de la solución.
<b>1.3.4</b>	Pruebas del aplicativo	En este paquete de trabajo se recogen las pruebas de comunicación con el servicio API REST y la validación del lote de credenciales generado.

Tabla 4.12. Diccionario EDT componentes.

En la tabla 4.2<sup>40</sup>, podemos observar las tareas que componen los paquetes de trabajo anteriormente descritos.

<b>Identificador</b>	<b>Nombre</b>	<b>Identificador del paquete</b>
<b>1.1.1</b>	Desarrollo de la EDT	1.1
<b>1.1.2</b>	Diccionario de la EDT	1.1
<b>1.1.3</b>	Identificación de las actividades de los paquetes	1.1
<b>1.1.4</b>	Planificación y elaboración del cronograma.	1.1
<b>1.2.1.1</b>	Redacción	1.2.1
<b>1.2.1.2</b>	Revisiones	1.2.1
<b>1.2.2.1</b>	Elaboración	1.2.2
<b>1.2.2.2</b>	Revisión	1.2.2
<b>1.2.2.3</b>	Ensayos	1.2.2
<b>1.3.1.1</b>	Estudio del estándar EDCI	1.3.1

<sup>40</sup> Fuente: Elaboración propia.

<b>1.3.1.2</b>	Estudio de la plataforma Europass y sus servicios.	1.3.1
<b>1.3.1.3</b>	Estudio del formato de las credenciales EDC	1.3.1
<b>1.3.2.1.1</b>	Estudio de las tecnologías API REST	1.3.2.1
<b>1.3.2.1.2</b>	Diseño de la comunicación entre cliente y servidor.	1.3.2.1
<b>1.3.2.2.1</b>	Estudio de las tecnologías para interactuar con la plantilla Excel EDC	1.3.2.2
<b>1.3.2.2.2</b>	Diseño del sistema.	1.3.2.2
<b>1.3.2.3.1</b>	Diagrama de clases	1.3.2.3
<b>1.3.2.3.2</b>	Análisis de los requisitos del sistema y diseño de la solución.	1.3.2.3
<b>1.3.2.4.1</b>	Estudio del formato HTML de la plantilla Excel EDC.	1.3.2.4
<b>1.3.3.1</b>	Implementación del algoritmo de generación de credenciales.	1.3.3
<b>1.3.3.2</b>	Implementación del formato HTML para las credenciales del SOFD.	1.3.3
<b>1.3.3.3</b>	Implementación del servicio API REST	1.3.3
<b>1.3.3.4</b>	Integración de las implementaciones.	1.3.3
<b>1.3.4.1</b>	Validación de la comunicación cliente servidor.	1.3.4
<b>1.3.4.2</b>	Validación de la generación correcta del lote de credenciales por el servicio REST	1.3.4

<b>1.3.4.3</b>	Validación, sellado y envío del lote de credenciales en la plataforma Europass	1.3.4
----------------	--------------------------------------------------------------------------------	-------

Tabla 4.13. Diccionario EDT tareas.

### 4.2.3 Cronograma

Tras el desglose de los diferentes paquetes y tareas, se ha diseñado un cronograma para planificar la estimación temporal en función de los objetivos mencionados en el primer capítulo de este documento para la realización de este Trabajo Fin de Grado. En la tabla 4.3<sup>41</sup>, podemos observar el cronograma mencionado con anterioridad.

OBJETIVOS / MESES	Marzo		Mayo		Junio		Julio	
	P M	S M	P M	S M	P M	S M	P M	S M
Objetivo 1								
Iteración 1: Estudio estándar EDCI y la plataforma Europass.								
Objetivo 2								
Iteración 1: Estudio de las tecnologías (Apache POI, API REST, etc..)								
Objetivo 3								
Iteración 1: Diseño								
Iteración 2: Implementación								
Iteración 3: Validación								
Objetivo 4								
Iteración 1: Diseño								
Iteración 2: Implementación								

<sup>41</sup> Fuente: Elaboración propia.

Iteración 3: Validación									
Objetivo 5									
Iteración 1: Creación de la plantilla personalizada para el SOFD									
Objetivo 6									
Iteración 1: Emisión y Validación de certificados									
Redacción de la Memoria TFG									
Iteración 1									
Iteración 2									
Iteración 3									
Iteración 4									
Iteración 5									
Presentación TFG									
Horas	45	45	25	40	30	40	35	40	Total 300

**PM = Primera Mitad del Mes.**  
**SM = Segunda Mitad del Mes.**

Tabla 4.14. Cronograma.



# Capítulo 5

## Diseño

### Contenidos

---

<b>5.1. Plantilla Excel EDC.....</b>	<b>59</b>
5.1.1 JExcel .....	59
5.1.2 Apache POI .....	60
5.1.3 Microsoft Visual Basic Script .....	61
5.1.4 Conclusiones .....	63
<b>5.2. API REST.....</b>	<b>63</b>
5.2.1 JAX-RS .....	65
5.2.2 Jersey .....	65
5.2.3 Conclusiones .....	65
<b>5.3. HTML EDC Wildcard .....</b>	<b>66</b>
<b>5.4. Formato de envío de datos .....</b>	<b>67</b>
5.4.1 XML .....	68
5.4.2 JSON .....	69
5.4.3 Conclusiones .....	70
<b>5.5. Ecosistema del proyecto.....</b>	<b>70</b>

---

En este capítulo se llevará acabo el diseño del sistema, tras todo el análisis y planificación mostrada anteriormente. En este apartado analizaremos las distintas partes que forman el proyecto, viendo diferentes alternativas que se han tenido en cuenta a la hora de implementar el aplicativo, siendo siempre el lenguaje de programación **Java**, centro del ecosistema, debido a su versatilidad y gran fuente de documentación.

En la sección 5.1, estudiaremos las tecnologías necesarias para poder manipular la plantilla Excel EDC y poder generar los lotes de credenciales.

En la sección 5.2, estudiaremos las tecnologías subyacentes para llevar a cabo la implementación de nuestra API REST.

En la sección 5.3, estudiaremos el formato para poder crear plantillas personalizadas para la creación de credenciales.

En la sección 5.4, estudiaremos los diferentes tipos de formato de archivo de intercambio de datos y su uso dentro del sistema.

En la sección 5.5, podemos visualizar una representación del ecosistema completo.

## 5.1. Plantilla Excel EDC

Como se ha explicado en el capítulo 2 sección 2.4, para poder llevar a cabo la emisión de lotes de credenciales mediante el **EDCI Issuer**, necesitamos cumplimentar correctamente la plantilla **Excel EDC**.

La plantilla **Excel EDC** [8] es un libro habilitado para macros de Excel (extensión .xslm), con un formato específico de los datos, divididos a lo largo de 8 hojas (formato definido en la sección 2.4 del capítulo 2) que nos permiten estructurar la información de la credencial digital que vamos a generar.

Partiendo de la selección de **Java** como lenguaje de programación para la implementación de este sistema, planteamos utilizar algunas de sus múltiples librerías para manejo de archivos Excel. Destacando principalmente dos opciones: **JExcel** y **Apache POI**.

### 5.1.1 JExcel

**JExcel** [17] es una librería de Java que facilita la generación de informes de Excel. Esta librería nos brinda una manera rápida y sencilla de manipular archivos Excel, mediante muy pocas líneas de código.

Si bien, **JExcel** es fácil de usar, gratuita y con buenas fuentes de documentación, no cumple con una característica vital para el desarrollo del proyecto. **JExcel** no soporta, de momento, los archivos Excel habilitado para macros (.xslm) y, por tanto, tampoco la plantilla **Excel EDC**.

En la figura 5.1<sup>42</sup>, podemos observar un fragmento de código para leer un fichero Excel con **JExcel**.

---

<sup>42</sup> Fuente: [17]

```

public class JExcelHelper {

    public Map<Integer, List<String>> readJExcel(String fileLocation)
        throws IOException, BiffException {

        Map<Integer, List<String>> data = new HashMap<>();

        Workbook workbook = Workbook.getWorkbook(new File(fileLocation));
        Sheet sheet = workbook.getSheet(0);
        int rows = sheet.getRows();
        int columns = sheet.getColumns();

        for (int i = 0; i < rows; i++) {
            data.put(i, new ArrayList<String>());
            for (int j = 0; j < columns; j++) {
                data.get(i)
                    .add(sheet.getCell(j, i)
                        .getContents());
            }
        }
        return data;
    }
}

```

Figura 5.1. Uso de la librería JExcel en Java.

## 5.1.2 Apache POI

**Apache POI** [18], [19] es una API para Java gratuita que nos permite manipular varios formatos de archivos basados en los estándares **Office Open XML (OOXML)** y el formato **OLE2** de **Microsoft**.

**OOXML** [20] es un formato de archivos abierto, cuyos datos están basados principalmente en el lenguaje de marcado **XML**. Se utiliza para representar y almacenar hojas de cálculo (**.xlsx**), graficas, presentaciones (**.pptx**) y documentos de texto (**.docx**). Este formato de archivo está recogido a lo largo de dos estándares: **ECMA-376** [21] y **ISO/IEC 29500** [22].

**Apache POI** ofrece un conjunto de funcionalidades más completo que **JExcel**, siendo por consiguiente, también más complejo de utilizar que el anterior, pero siendo todavía una solución de alto nivel. Además, debido a que está basado en **OOXML**, nos permite trabajar con ficheros Excel con macros habilitadas sin corromperlas.



En la tabla 5.1<sup>43</sup>, podemos observar los diferentes componentes que componen **Apache POI** y las aplicaciones con las que nos permite interacciones.

Component	Application type	Maven artifactId
<a href="#">POIFS</a>	OLE2 Filesystem	<i>poi</i>
<a href="#">HPSF</a>	OLE2 Property Sets	<i>poi</i>
<a href="#">HSSF</a>	Excel XLS	<i>poi</i>
<a href="#">HSLF</a>	PowerPoint PPT	<i>poi-scratchpad</i>
<a href="#">HWPF</a>	Word DOC	<i>poi-scratchpad</i>
<a href="#">HDGF</a>	Visio VSD	<i>poi-scratchpad</i>
<a href="#">HPBF</a>	Publisher PUB	<i>poi-scratchpad</i>
<a href="#">HSMF</a>	Outlook MSG	<i>poi-scratchpad</i>
DDF	Escher common drawings	<i>poi</i>
HWMF	WMF drawings	<i>poi-scratchpad</i>
<a href="#">OpenXML4J</a>	OOXML	<i>poi-ooxml</i> plus either <i>poi-ooxml-lite</i> or <i>poi-ooxml-full</i>
<a href="#">XSSF</a>	Excel XLSX	<i>poi-ooxml</i>
<a href="#">XSLF</a>	PowerPoint PPTX	<i>poi-ooxml</i>
<a href="#">XWPF</a>	Word DOCX	<i>poi-ooxml</i>
<a href="#">XDGF</a>	Visio VSDX	<i>poi-ooxml</i>
<a href="#">Common SL</a>	PowerPoint PPT and PPTX	<i>poi-scratchpad</i> and <i>poi-ooxml</i>
<a href="#">Common SS</a>	Excel XLS and XLSX	<i>poi-ooxml</i>

Tabla 5.15. Componentes de Apache POI.

A pesar la gran API que nos ofrece **Apache POI**, no nos permite ejecutar las macros de la plantilla **Excel EDC** desde Java, condición indispensable para el correcto funcionamiento del aplicativo que va a ser desarrollado.

### 5.1.3 Microsoft Visual Basic Script

**Visual Basic Script (VBS)** [23] es un lenguaje interpretado, mediante **Windows Script**, que nos permite crear scripts que pueden interactuar con las aplicaciones host de Microsoft. VBS es un lenguaje derivado de **Visual Basic** [24], el cual, es una lenguaje de programación orientado a objetos , desarrollado también por Microsoft.

**Windows Script (WScript)** [25] proporciona un entorno en el que los usuarios pueden ejecutar scripts de una variedad de lenguajes (entre los que se incluye VBS). Este entorno está integrado en el sistema operativo Windows y podemos acceder a él mediante la consola de comandos o cmd.

<sup>43</sup> Fuente: [19]

En la figura 5.2<sup>44</sup>, podemos observar un ejemplo de ejecución del entorno **WScript** mostrando la ayuda de este.

```
C:\Users\ricor>WScript /?
```

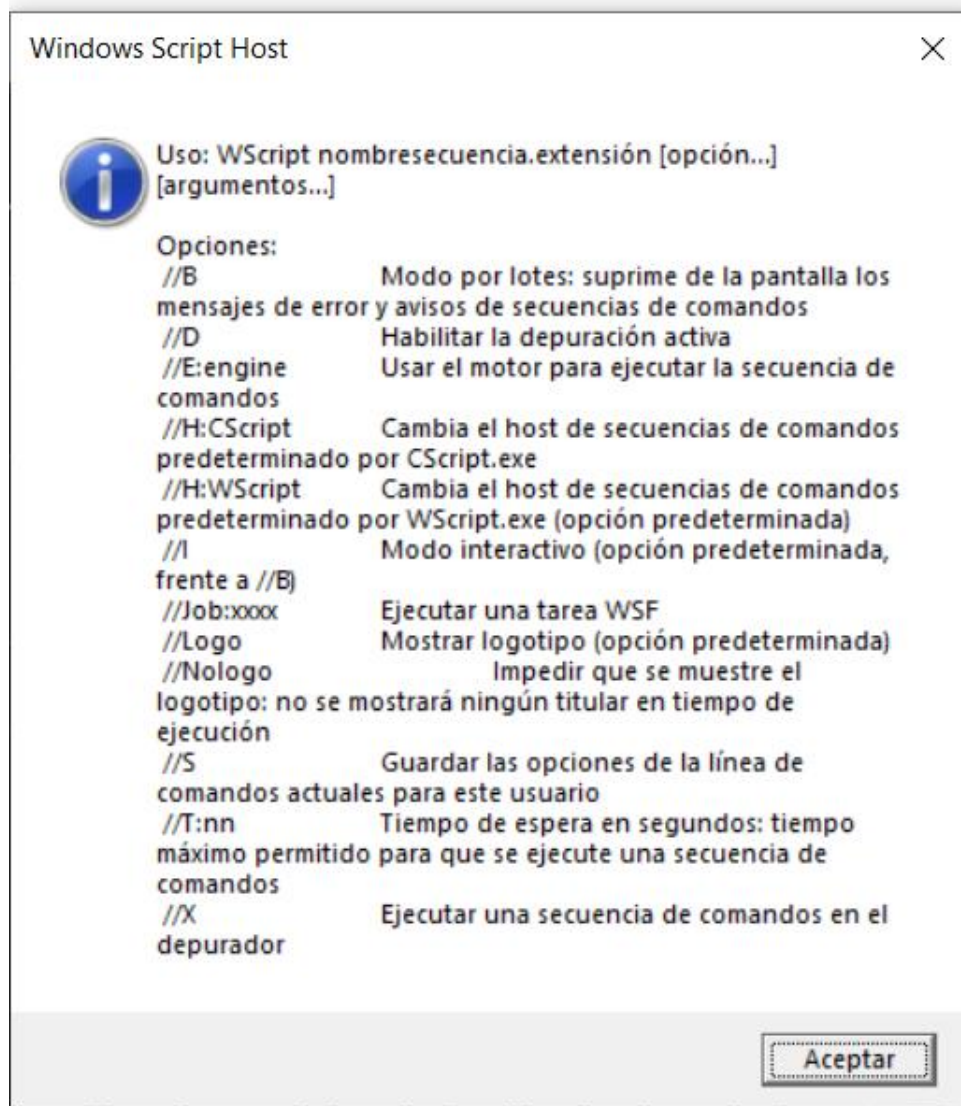


Figura 5.2. Ayuda de WScript.

Gracias a este entorno, podemos crear objetos desde VBS que nos permitan automatizar el uso de las aplicaciones de Microsoft. Con VBS podemos tener un control absoluto sobre los ficheros de Microsoft Excel, debido a que directamente nos comunicamos con la aplicación principal. Pudiendo incluso llamar a las macros contenidas dentro de un archivo Excel.

<sup>44</sup> Fuente: Elaboración propia.

---

## 5.1.4 Conclusiones

Finalmente, la única alternativa válida para llevar a cabo el desarrollo del aplicativo era utilizar **VBScript**. Debido a que:

- **JExcel**, no soportaba archivos Excel habilitados para macros (.xlsm).
- **Apache POI**, si soporta archivos Excel habilitados para macros (.xlsm), pero no puede ejecutar o interactuar con las macros contenidas en el mismo.

El uso de VBScript ocasiona una aplicación más compleja que con las librerías de Java, pero, nos aporta un control absoluto sobre la plantilla **Excel EDCI** y sus macros.

## 5.2. API REST

Una de las principales partes del diseño de nuestro sistema, será el desarrollo de una **API REST**, de modo que podamos agilizar el proceso de comunicación entre la entidad que quiera emitir sus certificados en el formato **EDCI**.

Para probar el funcionamiento de la emisión de las credenciales digitales en la plataforma Europass, se ha colaborado con el **SOFD (Servicio de Orientación y Formación Docente)** de la **Universidad de Extremadura**.

Una API REST [26] es un mecanismo que permite a una aplicación o servicio acceder a un recurso o conjunto de recursos dentro de otro aplicación o servicio.

Destacamos dos principales interlocutores en este mecanismo:

- El **cliente**, aplicación o servicio que realiza o solicita el acceso al recurso.
- El **servidor**, aplicación o servicio que contiene el recurso que solicita el cliente.

Dentro de nuestro marco de desarrollo, el cliente haría referencia a las entidades o instituciones que quieren obtener sus credenciales en formato EDC y el servidor sería nuestro aplicativo que facilitaría el recurso tras recibir la petición con los datos del cliente. El cliente y el servidor se comunicarán principalmente mediante solicitudes **HTTP** [27].

Dentro del protocolo **HTTP**, podemos destacar 4 solicitudes:

- Solicitud **GET**, el cliente solicita un recurso. Esta solicitud no tiene cuerpo y los parámetros se envían en la cabecera.
- Solicitud **POST**, el cliente envía información al servidor, para que este genere o cree el recurso solicitado.
- Solicitud **PUT**, el cliente envía información al servidor, para que este modifique el recurso creado.
- Solicitud **DELETE**, el cliente solicita la eliminación de un recurso situado en el servidor.

Dentro de nuestro marco de desarrollo, la institución realizará una solicitud **POST** al servidor, la cual, contendrá la información de las credenciales a generar en formato **EDC**. El servidor procesará la información, generará el lote de credenciales y finalmente, será devuelto en la respuesta de la solicitud a la institución, listo para subirlo y sellarlo mediante el **EDCI Issuer** de la plataforma **Europass**. En la figura 5.3<sup>45</sup>, se nos muestra la ejemplificación de una posible API REST.

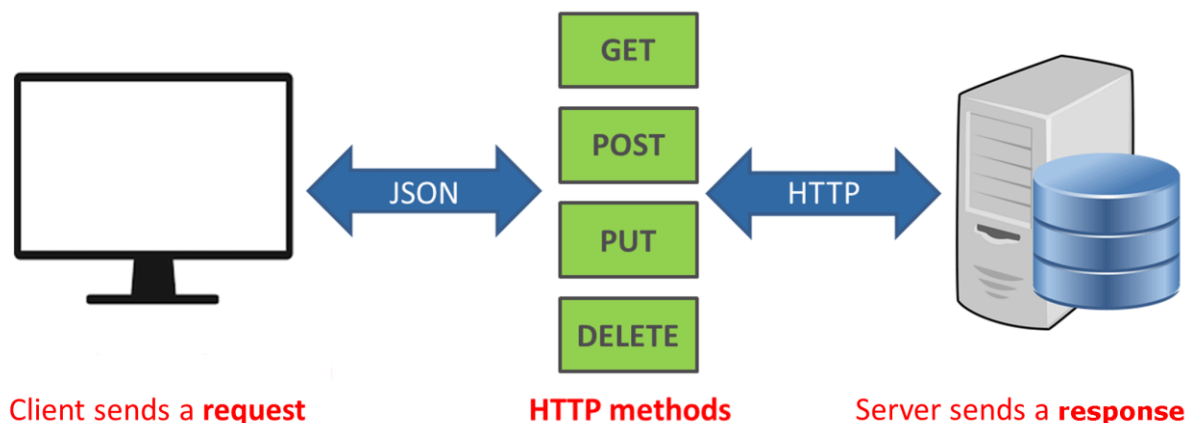


Figura 5.3. Ejemplificación de una API REST.

Durante el desarrollo de nuestro proyecto, haremos uso del lenguaje de programación **Java** y de las tecnologías **JAX-RS** y **JERSEY** para la construcción de nuestro servicio **REST**.

<sup>45</sup> Fuente: [28]

## 5.2.1 JAX-RS

**JAX-RS (Java API for RESTful Web Services)** [29] es una especificación para el desarrollo de servicios web REST de manera sencilla y natural. Esta especificación está basada en el uso de “**anotaciones**” para simplificar el desarrollo y el despliegue de los servicios.

Estas anotaciones permiten:

- Identificar componentes de la aplicación.
- Enrutar solicitudes a un método apropiado en una clase seleccionada
- Extraer datos de una solicitud en los argumentos de un método.
- Proporcionar metadatos utilizados en las respuestas.

## 5.2.2 Jersey

**Jersey** [30] es un marco de desarrollo para servicios **REST**, que nos facilita más el desarrollo de este tipo de servicios. Este marco proporciona una implementación de la especificación **JAX-RS**, junto con una **API** propia que amplía las características de la anterior.

## 5.2.3 Conclusiones

La **API REST** permite desacoplar las aplicaciones clientes servidor (sistemas de la institución que quiera emitir las credenciales), permitiendo agilizar el proceso de comunicación entre ambos y facilitando a los clientes la automatización de la generación de las credenciales, sin necesidad, de que se preocupen del formato de la plantilla **Excel EDC**.

Los únicos requisitos que el cliente necesitará para poder comunicarse con el servidor serán:

- La **URL** o dirección web del servicio, sobre el que se realizan las solicitudes **HTTP**.
- La información para enviar al servidor, que deberá seguir un criterio o formato específico para ser reconocido por el servidor.

---

## 5.3. HTML EDC Wildcard

La plantilla **Excel EDC**, nos ofrece la posibilidad de modificar el formato visual de la credencial digital que vamos a generar. Esto es posible gracias a los campos mencionados en la sección 2.4 del capítulo 2 de este mismo documento. Entre estos campos destacamos:

- Campo “**Background**”, este campo contiene una URL a la imagen de fondo del certificado. Esta URL debe ser pública y la imagen referenciada debe tener un formato DIN A4 (210 x 297 mm), regulado por la **ISO 216:2007** [31].
- Campo “**HTML**”, este campo contiene la definición del formato visual de la información, contenida en la credencial, mediante HTML y CSS (no está permitido incluir **JavaScript**).

Dentro del marco de desarrollo de nuestro TFG, tendremos que diseñar una plantilla personalizada para la emisión de las credenciales digitales para el SOFD de la Universidad de Extremadura.

El formato **HTML** de la plantilla **Excel EDC** [32] nos permite utilizar expresiones **XPath** [33], que nos permiten navegar a través de elementos y atributos en un documento **XML** (formato en el que se representa el modelo de datos de las credenciales Europass). De este modo, podemos acceder a la información embebida dentro del archivo **XML** de la credencial y representarlo en el formato **HTML**.

Dentro de estas expresiones **XPath**, destacamos:

- “[\$/eup:europassCredential/eup:title\$]”, **XPath** que nos permite obtener el título de la credencial recogido en la plantilla Excel EDC.
- “[\$/eup:europassCredential/eup:agentReferences/eup:organization[0]/eup:logo\$]”, **XPath** que nos permite visualizar el logo de la Entidad que va a emitir la credencial.
- “[\$/eup:europassCredential/eup:credentialSubject/eup:fullName\$]”, **XPath** que nos permite visualizar el nombre completo de la persona que va a recibir la credencial.
- “[\$/eup:organization[@id=/eup:europassCredential/cred:issuer/@idref]/eup:prefLabel\$]”, **XPath** que nos permite acceder al nombre de la organización que va a emitir la credencial.

En la figura 5.4<sup>46</sup>, podemos observar la estructura HTML por defecto de una credencial digital Europass.

```
<div style="font-style: normal; padding-bottom: 2rem; padding-top: 6rem;">
  <div style="flex: 0 0 100%; max-width: 100%; display: block; text-align: center;">
    
    <br/>
    <p style="font-size: 18px; color: #525252;">
      [$//eup:organization[@id=eup:europassCredential/cred:issuer/@idref]/eup:prefLabel$]
    </p>
  </div>
  <div style="flex: 0 0 100%; max-width: 100%;">
    <p style="padding-left: 0.5rem !important; margin: 3rem 1em; text-align: center !important;
font-weight: 700 !important; color: #004494 !important; text-decoration: underline; font-size: 31px;">
      [$//eup:europassCredential/eup:title$]
    </p>
  </div>
  <div style="flex: 0 0 100%; max-width: 100%;">
    <p style="text-align: center; font-weight: 700 !important; font-size: 18px;">
      [$//eup:organization[@id=eup:europassCredential/cred:issuer/@idref]/eup:prefLabel$]
    </p>
    <p style="text-align: center; font-size: 18px;">
      ($diploma.msg.certifiesThat$)
    </p>
  </div>
  <div style="flex: 0 0 100%; max-width: 100%;">
    <p style="font-weight: 700 !important; font-size: 31px; margin-top: 2rem; margin-bottom: 2rem;
color: #004494 !important; text-align: center; font-style: italic !important;">
      [$//eup:europassCredential/eup:credentialSubject/eup:fullName$]
    </p>
  </div>
  <div style="flex: 0 0 100%; max-width: 100%; height: 7cm; margin: 0 auto; width: 87%; padding-bottom:
1rem !important;">
    <p style="font-size: 12px; text-align: center; font-weight: 300 !important;">
      [$//eup:europassCredential/eup:description$]
    </p>
  </div>
  <div style="flex: 0 0 100%; max-width: 100%; margin-left: 2.5rem;">
    <p style="font-style: oblique; text-align: left; font-weight: 300 !important;">
      [$//eup:europassCredential/cred:issued$]
    </p>
  </div>
</div>
```

Figura 5.4. Estructura HTML por defecto de EDC.

## 5.4. Formato de envío de datos

Una de las partes fundamentales del sistema es el formato de envío de los datos de las credenciales desde la institución (cliente) hasta el servicio (servidor) de conversión de las credenciales a EDC que va a ser desarrollado. En la actualidad, podemos destacar dos tipos de formato de envío de datos: **XML** y **JSON**.

<sup>46</sup> Fuente: [32]

## 5.4.1 XML

**XML (Extensible Markup Language)** [6], [34] es un formato de texto simple y muy flexive, utilizado para el intercambio de datos en la web y en otros entornos.

En la figura 5.5<sup>47</sup>, podemos observar un ejemplo de archivo de datos XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<vp:verifiablePresentation xmlns="http://data.europa.eu/snb" xmlns:vp="http://data.europa.eu/snb/vp" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:cred="http://data.europa.eu/europass/model/credentials/w3c#" id="urn:verifiable:aa504b0f-54e9-4d78-9368-2adc5b48a14e">
  <vp:type targetFrameworkUri="http://data.europa.eu/snb/verifiable-presentation/25831c2" uri="http://data.europa.eu/snb/verifiable-presentation/c_825cddc8">
    <targetName>
      <text lang="en" content-type="text/plain">Mandated Issue</text>
    </targetName>
    <targetFrameworkName>
      <text lang="en" content-type="text/plain">Verifiable Presentation Type</text>
    </targetFrameworkName>
  </vp:type>
  <vp:issuer id="urn:repass:org:1">
    <registration>VATES-Q06180018</registration>
    <prefLabel>
      <text lang="es" content-type="text/plain">UNIVERSIDAD DE EXTREMADURA</text>
    </prefLabel>
    <altLabel>
      <text lang="es" content-type="text/plain">SelloUEx2022</text>
    </altLabel>
    <hasLocation>
      <id urn:repass:location:vp:1</id>
      <hasAddress>
        <country targetFrameworkUri="http://publications.europa.eu/resource/authority/country" targetNotation="country" uri="http://publications.europa.eu/resource/authority/country/ESP">
          <targetName>
            <text lang="es" content-type="text/plain">España</text>
          </targetName>
          <targetFrameworkName>
            <text lang="en" content-type="text/plain">Country</text>
          </targetFrameworkName>
        </country>
      </hasAddress>
    </hasLocation>
  </vp:issuer>
  <europassCredential xsdVersion="0.10.0" cred:id="urn:credential:6ec8cea9-7e1d-4b1e-a8ed-ee7161de7800">
    <type targetFrameworkUri="http://data.europa.eu/snb/credential/25831c2" uri="http://data.europa.eu/snb/credential/e34929035b">
      <targetName>
        <text lang="en" content-type="text/plain">Generic</text>
        <text lang="es" content-type="text/plain">Genérica</text>
      </targetName>
      <targetFrameworkName>
        <text lang="en" content-type="text/plain">Europass Standard List of Credential Types</text>
        <text lang="es" content-type="text/plain">Lista normalizada de tipos de credenciales de Europass</text>
      </targetFrameworkName>
    </type>
    <primaryLanguage>es</primaryLanguage>
    <availableLanguages>
      <language>es</language>
    </availableLanguages>
    <cred:validFrom>2022-03-07T00:00:00+01:00</cred:validFrom>
    <cred:issued>2022-07-03T00:51:12+02:00</cred:issued>
    <cred:issuer idref="urn:repass:org:1"/>
    <title>
      <text lang="en" content-type="text/plain">JORNADA DE FORMACIÓN-DIFUSIÓN DE LA OFICINA DE CALIDAD</text>
    </title>
  </europassCredential>
</verifiablePresentation>
```

Figura 5.5. Estructura XML de una credencial emitida en la plataforma Europass.

Como podemos observar, es un metalenguaje, que requiere de etiquetas de apertura y cierre, del mismo modo que **HTML**. **XML** nos ofrece ventajas como el uso de **XPath**, mencionado en la sección anterior, y el soporte de metadatos en forma de atributos dentro de las etiquetas, permitiéndonos definir etiquetas personalizadas para describir y organizar los datos de un documento.

Además, trabaja sobre texto plano, lo que conlleva a su rápido procesamiento en el envío entre cliente y servidor.

<sup>47</sup> Fuente: Elaboración propia.



## 5.4.2 JSON

Por otro lado, tenemos **JSON (JavaScript Object Notation)** [34], [35] es un formato ligero de intercambio de datos, mucho más simple que **XML**, lo que conlleva, que sea mucho más fácil leer y escribir para los humanos. **JSON** forma parte de un subconjunto del lenguaje de programación **JavaScript**, lo que permite, encajar de forma muy natural con el mismo.

JSON, no aporta tanta información sobre los datos como los archivos **XML** y, por tanto, su velocidad de procesamiento en la recepción y envío es mayor que un archivo **XML**, ambos con los mismos datos.

Dentro de los archivos JSON, principalmente, nos encontramos pares nombre/valor agrupados u ordenados, formando una lista de valores. Actualmente, muchos lo autodenominan como el formato de intercambio preferido para servicios web. En la figura 5.6<sup>48</sup>, podemos observar un ejemplo del formato de un archivo JSON.

```
{
  "nombre": "Richard",
  "primerapellido": "Corchero",
  "segundoapellido": "Vélez",
  "amigos": [
    {
      "nombre": "Mario",
      "ape1": "Acedo",
      "ape2": "Conejo"
    },
    {
      "nombre": "Héctor",
      "ape1": "Sánchez",
      "ape2": "Santamaría"
    },
    {
      "nombre": "Álvaro José",
      "ape1": "Navarrete",
      "ape2": "Pozueco"
    }
  ]
}
```

Figura 5.6. Estructura de un archivo en formato JSON.

<sup>48</sup> Fuente: Elaboración propia.

### 5.4.3 Conclusiones

Tanto **XML** como **JSON**, son dos formatos totalmente válidos para el envío de información entre aplicaciones web, no obstante, para el desarrollo de este TFG, se hará uso de archivos **JSON**. Las principales cuestiones que nos han hecho tomar esta decisión han sido:

- La mayor aceptación de **JSON**, actualmente, en el envío de información por parte de la mayoría de los servicios web.
- Su mayor velocidad de procesamiento frente a **XML**.
- Su interoperabilidad natural con el lenguaje **JavaScript**.

### 5.5. Ecosistema del proyecto

Tras detallar en los apartados anteriores, las distintas tecnologías que se utilizarán para el desarrollo del servicio expuesto en este TFG. En la figura 5.7<sup>49</sup>, se puede observar una ejemplificación de la estructura del sistema a desarrollar.

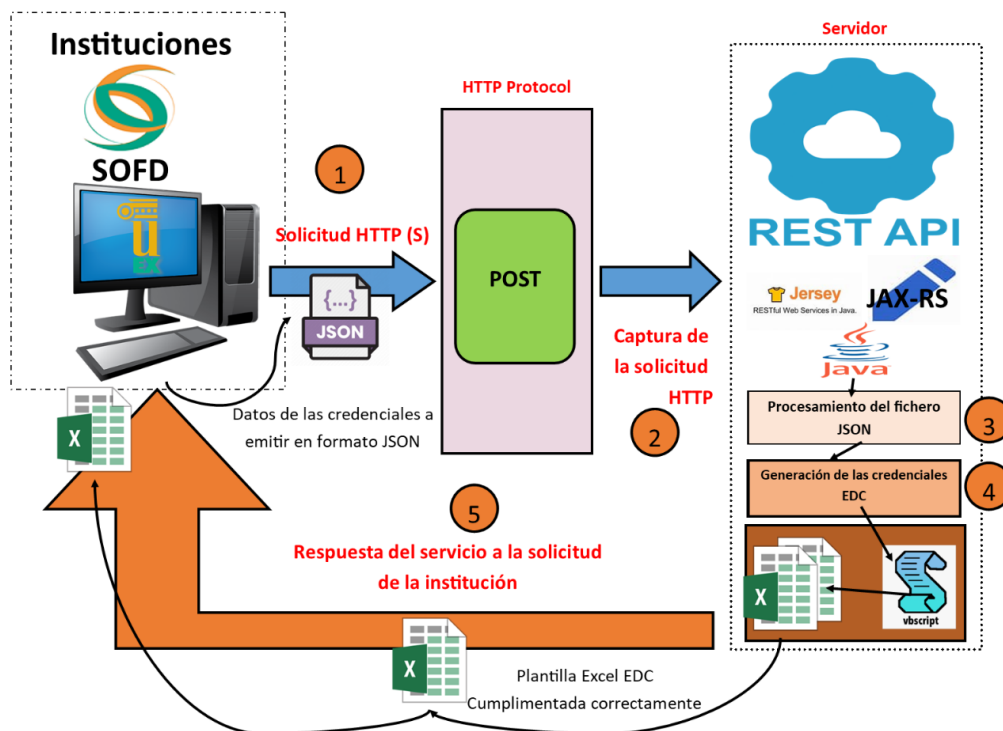


Figura 5.7. Estructura del sistema a desarrollar.

<sup>49</sup> Fuente: Elaboración propia.

Como podemos observar, las instituciones u organizaciones que quieran emitir sus credenciales deberán empaquetar las mismas en un archivo **JSON** y enviar el mismo en el cuerpo de una solicitud **HTTP POST** hacia la **URL** del servicio.

Si el formato del **JSON** es correcto, el servicio aceptará la solicitud y comenzará a procesar el fichero **JSON**. Tras este procesado, la información se enviará como argumentos al script de **Visual Basic**, siendo este último, el encargado de generar la nueva plantilla **Excel EDC** cumplimentada con la información de los certificados enviados. Para generar esa nueva plantilla, hará uso de una plantilla por defecto preparada en el mismo servidor.

Finalmente, el servicio devolverá el nuevo archivo Excel generado a la institución que lo solicitó. Por último, la institución solo deberá acceder al apartado **EDCI Issuer** del portal **Europass** y proceder con la validación, sellado y envío de los certificados.



# Capítulo 6

## Implementación

### Contenidos

---

<b>6.1. Configuración de la plantilla Excel EDC .....</b>	<b>74</b>
6.1.1 Macros en Excel .....	74
6.1.2 Adaptación de la Macro AddCredentials. ....	75
6.1.3 Adaptación de la Macro AddPersons. ....	76
<b>6.2. Automatización del cumplimentado de la plantilla Excel EDC. ....</b>	<b>77</b>
<b>6.3. Creación de la HTML EDC Wildcard para el SOFD. ....</b>	<b>79</b>
<b>6.4. Creación del servicio API REST mediante Jersey y JAX-RS. ....</b>	<b>80</b>
6.4.1. Maven y Eclipse EE .....	80
6.4.2. Incluir dependencia Jersey en nuestro proyecto. ....	80
6.4.3. Desarrollo del servicio mediante las anotaciones JAX-RS. ....	81
<b>6.5. Procesamiento del JSON y llamada a VBScript.....</b>	<b>85</b>
<b>6.6. Despliegue, Pruebas y validación del servicio REST .....</b>	<b>87</b>

---

Tras las fases de diseño y análisis del servicio a desarrollar, en este capítulo se describirán los detalles de la implementación de este. Podemos dividir este capítulo en seis secciones.

En la sección 6.1, se expondrá los cambios realizados sobre la plantilla **Excel EDC**, por defecto, para poder automatizar el proceso de generación de las credenciales en formato **EDCI**.

En la sección 6.2, se desarrollará el script en **Visual Basic** que cumplimente de manera automática el fichero **Excel EDC**, mediante los datos enviados por la organización.

En la sección 6.3, llevaremos a cabo el desarrollo del formato visual personalizado para los certificados del **Servicio de Orientación y Formación Docente (SOFD)**.

En la sección 6.4, se comentará la configuración y pasos para desarrollar el servicio **API REST** mediante las tecnologías mencionadas anteriormente.

En la sección 6.5, se detallarán el algoritmo para el procesamiento del JSON y la invocación del Script de VB desde nuestro servicio REST.

Finalmente, en la sección 6.6, se llevarán a cabo pruebas para comprobar que el servicio responde a las peticiones del cliente y que la plantilla **Excel EDC** devuelta es aceptada correctamente por el servicio **EDCI Issuer** de la plataforma Europass.

## 6.1. Configuración de la plantilla Excel EDC

Para poder llevar a cabo el servicio correctamente, el servidor tendrá situado, en un directorio específico, la plantilla Excel EDC por defecto ofrecida por el servicio EDCI Issuer, la cual, cumplimentada correctamente, se podrá validar, sellar y enviar a las carteras o wallets de los usuarios del portal Europass. Dentro de este archivo Excel, las **macros** juegan un papel fundamental.

### 6.1.1 Macros en Excel

Las **macros de Excel** [36], son fragmentos de código, escritos en Visual Basic, que permiten configurar tareas complejas, de modo, que puedas repetir las todas las veces que quieras sin necesidad de realizarlo manualmente.

La plantilla Excel EDC contiene una serie de macros que agilizan y facilitan el proceso de cumplimentado del Excel manualmente. En la figura 6.1, podemos observar el código contenido dentro de una de las macros de la plantilla Excel EDC.

```
Sub UnProtect_All_Sheets()  
Dim ws As Worksheet  
    For Each ws In ActiveWorkbook.Worksheets  
        ws.Unprotect Password:="EuropassBear2020"  
    Next ws  
End Sub
```

Figura 6.1. Macro contenida en la plantilla Excel EDC que desprotege todas las hojas del archivo.

Para poder automatizar el proceso mediante VBScript, deberemos modificar algunas de estas macros, para que reciban la información como argumentos y no por diálogos gráficos donde tengamos que introducir la información manualmente.

### 6.1.2 Adaptación de la Macro AddCredentials.

La macro **AddCredentials**, situada en la hoja “**Credentials**” de la plantilla **Excel EDC**, nos permite ir desbloqueando las diferentes filas para poder cumplimentar la información referente a las credenciales. Si no utilizamos esta macro para desbloquear las filas, desde **VBScript** no podremos acceder a esas filas y por tanto no podremos llevar a cabo el proceso de completado del fichero.

Para automatizar el proceso desde **VBScript**, debemos poder pasar la información, sobre el número de filas que vamos a destapar, mediante argumentos a la macro.

No obstante, esta macro no está programada para recibir la información como argumentos o parámetros, sino que lo realiza mediante cuadro de diálogos para introducir la información. En la figura 6.2<sup>50</sup>, podemos observar la ejecución de la macro por defecto en la plantilla **Excel EDC**.

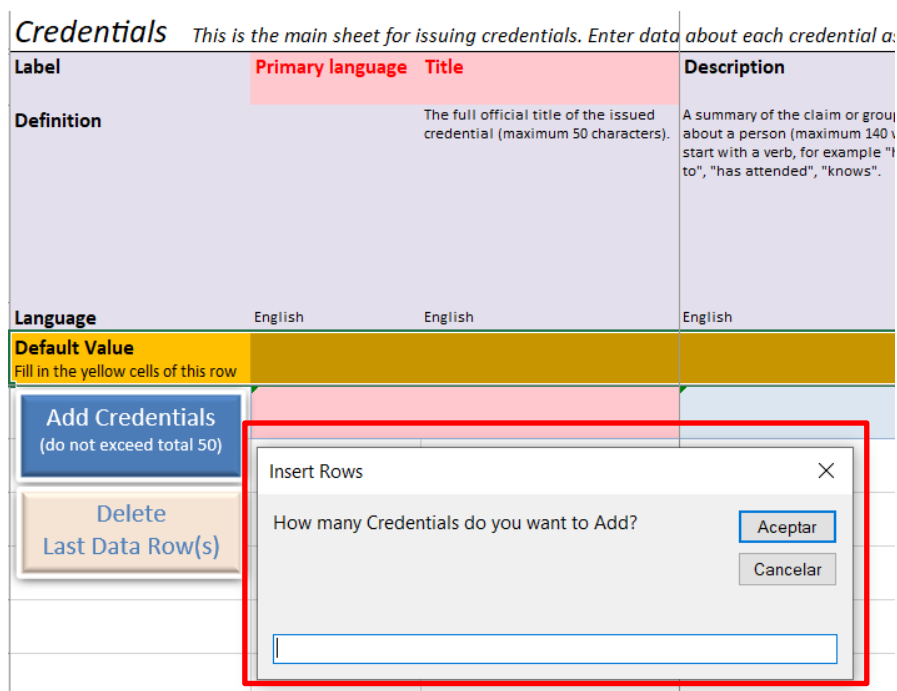


Figura 6.2. Ejecución de la macro AddCredentials por defecto.

<sup>50</sup> Fuente: Elaboración propia.

Adaptar la macro anterior, para poder ser ejecutada desde VBScript, es relativamente sencillo.

Para ello, deberemos acceder al código de la macro implementada, añadirle un parámetro a la función de tipo entero y asignar ese parámetro como valor de las filas a destacar (eliminado el dialogo gráfico que se mostraba anteriormente) de la función macro. En el algoritmo 6.1, podemos observar el código de la macro **AddCredentials** adaptado.

```

Sub AddCredentials(number As Integer)
ActiveSheet.Unprotect Password:="EuropassBear2020"
Dim numRows As Integer
Dim counter As Integer

'Select the current row
ActiveCell.EntireRow.Select
On Error GoTo Last
numRows = number

'Keep on inserting rows until we reach the desired number
For counter = 1 To numRows
ActiveSheet.ListObjects("Credentials").ListRows.Add AlwaysInsert:=True
Next counter
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True, AllowFormattingColumns:=True, Password:="EuropassBear2020"
Last: Exit Sub
End Sub

```

Parámetro de la función macro que será enviado desde el script VB

Eliminación del dialogo gráfico y asignación del parámetro enviado.

Algoritmo 6.1. Código adaptado para nuestro servicio contenido en la macro AddCredentials.

### 6.1.3 Adaptación de la Macro AddPersons.

La macro **AddPersons**, situada en la hoja “**Persons**”, de la plantilla Excel EDC, nos permite ir desbloqueando las diferentes filas para poder cumplimentar la información referente a las personas que van a obtener la credencial. Esta macro presenta el mismo problema inicial que la anterior y su adaptación para el correcto funcionamiento del servicio es muy similar. En el algoritmo 6.2, podemos observar el código de la macro **AddPersons** adaptado.

```

Private Sub AddPersons(number As Integer)
ActiveSheet.Unprotect Password:="EuropassBear2020"
Dim numRows As Integer
Dim counter As Integer
Application.EnableEvents = False
'Select the current row
ActiveCell.EntireRow.Select
On Error GoTo Last
numRows = number

'Keep on inserting rows until we reach the desired number
For counter = 1 To numRows
ActiveSheet.ListObjects("Persons").ListRows.Add AlwaysInsert:=True
Next counter
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True, AllowFormattingColumns:=True, Password:="EuropassBear2020"
Application.EnableEvents = True
Last: Exit Sub
End Sub

```

Parámetro de la función macro que será enviado desde el script VB

Eliminación del dialogo gráfico y asignación del parámetro enviado.

Algoritmo 6.2. Código adaptado para nuestro servicio contenido en la macro AddPersons.



Mismo proceso debería realizarse para las macros que permiten añadir filas en cada una de las hojas del fichero Excel, no obstante, para el funcionamiento de nuestro servicio, en esta primera versión, no será necesario, debido a que, el SOFD del Universidad de Extremadura no aportará información sobre las mismas.

## 6.2. Automatización del cumplimentado de la plantilla Excel EDC.

Una vez hemos configurado las macros de la platilla **Excel EDC**. Podemos implementar el algoritmo **VBScript** que hará uso de la anterior, para generar una nueva plantilla **EDC**, lista para subir a Europass.

Este script será llamado desde el servicio **REST** en Java, mediante una llamada al sistema **Windows** (Durante este proyecto se ha utilizado **Windows 10** como sistema operativo del servidor que sostiene el servicio). Para poder ejecutar el script, correctamente, deberemos pasarle una serie de argumentos desde el servicio **API REST**. Estos argumentos, se pasan todos como texto o cadenas de caracteres, que posteriormente el script interpreta.

El script recibe un total de seis argumentos:

- **Primer argumento (args (0))**, directorio de trabajo donde se encuentra la plantilla Excel EDC, la cual, es utilizada para generar las nuevas plantillas.
- **Segundo argumento (args (1))**, nombre de la plantilla genérica Excel EDC.
- **Tercer argumento (args (2))**, nombre de la nueva plantilla que va a ser generada y enviada en la respuesta al cliente.
- **Cuarto argumento (args (3))**, número de credenciales que vamos a emitir.
- **Quinto argumento (args (4))**, datos referentes a las credenciales que vamos a emitir, personas, etc.
- **Sexto argumento (args (5))**, formatos HTML EDC wildcard para cada una de las credenciales que vamos a emitir.

En el algoritmo 6.3, podemos observar el código del script VB que permite el cumplimentado de las credenciales EDC.

```

' Definicion de variables
Dim args, objExcel, objWorkbook, objSheet, dic

' Creamos un diccionario con las columnas de las celdas según cada hoja
set dic = CreateObject("Scripting.Dictionary")
dic(2) = Array(2, 4, 62, 64, 66, 68) ' Columnas Celdas hoja 2
dic(3) = Array(3, 4, 21, 22) ' Columnas Celdas hoja 3
dic(4) = Array(2, 60, 121) ' Columnas Celdas hoja 4

html_cell = 72 ' Posición de la celda del formato HTML de la credencial
row_index_init = 13 ' Posición de la fila inicial a partir de la que empezamos a rellenar el fichero Excel.

'Recuperamos los argumentos pasados en la llamada del Script.
set args = WScript.Arguments

' Creamos un objeto de tipo Excel.Application para interactuar con estos fichero
set objExcel = CreateObject("Excel.Application")
' Abrimos la plantilla genérica ,a partir de la cual, generaremos el fichero cumplimentado.
set objWorkbook= objExcel.workbooks.open(args(0) & args(1))

objExcel.visible = False ' No queremos que se nos abra la interfaz de Excel.
objExcel.Run "Sheet10.UnProtect_All_Sheets" ' Ejecutamos la macro que desprotege todas las hojas del archivo
objExcel.Run "Sheet1.AddCredentials", CInt(args(3)) ' Ejecutamos la macro que nos desbloquea las filas que le pasemos por parámetro de la hoja de credenciales
objExcel.Run "Sheet2.AddPersons", CInt(args(3)) ' Ejecutamos la macro que nos desbloquea las filas que le pasemos por parámetro de la hoja de personas.

formatos_html = Split(args(5), "&SEPARADOR&") ' Conseguimos los diferentes formatos HTML Wildcard
credenciales = Split(args(4), "|") ' Conseguimos los diferentes datos referentes a las credenciales que vamos a emitir.

' Vamos recorriendo los datos de las diferentes credenciales y rellenado el nuevo fichero Excel.

FOR EACH credencial IN credenciales
    lineas = Split(credencial, "+")
    FOR EACH linea IN lineas
        line_content = Split(linea, ";")
        sheet_index = CInt(line_content(0))
        set objSheet = objWorkbook.Worksheets(sheet_index)
        cells_index = dic(sheet_index)
        data = Split(line_content(1), "&")
        FOR i = 0 TO UBound(cells_index) step 1
            objSheet.cells(row_index_init, cells_index(i)).Value = data(i)
        NEXT
    NEXT
    row_index_init = row_index_init + 1
NEXT

' Asignamos el formato visual en html de las credenciales (HTML e imagen de fondo)
set objSheet = objWorkbook.Worksheets(2)
FOR i= 0 TO CInt(args(3))
    objSheet.cells(13+i, 72).Value = formatos_html(i)
    objSheet.cells(13+i, 71).Value = "https://github.com/ricorcher/imagenes/blob/main/fondo_certificados.jpg?raw=true"
NEXT

' Finalizamos y guardamos el fichero con un nuevo nombre, de manera que no modifiquemos la plantilla.
objExcel.DisplayAlerts = False
objExcel.ActiveWorkbook.SaveAs args(0) & args(2)
objExcel.DisplayAlerts = True
objExcel.ActiveWorkbook.Close(0)
objExcel.Quit

```

Inicialización de las variables y objetos de scripting necesarios para trabajar con la aplicación Excel.

Recuperamos los argumentos, establecemos comunicación con la aplicación de Microsoft Excel y abrimos la plantilla Excel EDC.

Parseamos y recorremos los datos recibidos desde Java. Vamos hoja por hoja, línea por línea, introduciendo la información en las celdas indicadas para cada una de las hojas.

Asignamos el formato HTML Wildcard personalizado para el SOFD y la imagen de fondo del certificado.

Guardamos la nueva plantilla, cerramos el libro y finalizamos la aplicación de Excel.

Algoritmo 6.3. Código de VBScript que genera el lote de credenciales EDC que será enviado de vuelta al cliente.

## 6.3. Creación de la HTML EDC Wildcard para el SOFD.

Dentro del marco de desarrollo de este TFG, se ha desarrollado una plantilla HTML personalizada para la emisión de las credenciales del Servicio de Orientación y Formación Docente de la Universidad de Extremadura. En la figura 6.3<sup>51</sup>, podemos observar la estructura HTML de las credenciales del SOFD.

```
<div style='font-style: normal; padding-bottom: 2rem; padding-top: 6rem;'>
  <div style='flex: 0 0 100%; max-width: 100%; display: block; text-align: center;'>
    <img style='max-height: 5.5rem; width: auto; src='["$//eup:organization[@id=eup:europassCredential/cred:issuer/@idref]/eup:logo$"]/'>
    <br/>
  </div>
  <div style='flex: 0 0 100%; max-width: 70%; display: block; text-align: justify; margin-left: 10%; margin-right: 10%;>
    <p style='font-size: 18px; color: #000000;'>
      <b>{$nombre_firma1}, {$cargo_firma1}</b>de la ["$//eup:organization[@id=eup:europassCredential/cred:issuer/@idref]/eup:prefLabel$"].</b>
    </p>
  </div>
  <div style='flex: 0 0 100%; max-width: 100%;text-align: center;'>
    <p style='font-size: 31px;'><b>CERTIFICA</b></p>
    <p style='font-size: 18px;'><b>Que</b></p>
    <p style='font-size: 28px; color: #127832;'> ["$//eup:europassCredential/eup:credentialSubject/eup:fullName$"] </p>
  </div>
  <br/>
  <br/>
  <div style='flex: 0 0 100%; max-width: 100%; margin-left: 10%; margin-right: 10%; text-align: justify;'>
    <p style='font-size: 22px;'>
      Ha realizado el taller <b>["$//eup:europassCredential/eup:title$"]</b>, organizado por
      el Servicio de Orientación y Formación Docente de la Universidad de Extremadura,
      dentro del plan {$anio}, se celebró en {$zona}, {$dia_imparte}, siendo su duración
      de {$numero_horas}.
    </p>
    <br/>
    <br/>
    <br/>
    <br/>
    <p style='float:left'><b>{$cargo_firma2}, {$nombre_firma2}</b></p>
    <p style='float:right'><b>{$cargo_firma1}, {$nombre_firma1}</b></p>
  </div>
  <div style='flex: 0 0 100%; max-width: 100%; margin-left: 2.5rem;'>
    <p style='font-style: oblique; text-align: left; font-weight: 300 !important;'>["$//eup:europassCredential/cred:issued$"]</p>
  </div>
</div>
```

Figura 6.3. Estructura HTML de las credenciales del SOFD.

De la figura anterior, podemos destacar elementos **Xpath**, que nos permiten, mostrar información de la estructura XML de la credencial al procesarla por Europass, como, por ejemplo, el título de la credencial, logo de la institución, nombre de la persona, etc.

Además, destacamos también una serie de campos propios que serán utilizado para mostrar información propia del SOFD y que serán reemplazados por la información oportuna durante el procesamiento del archivo JSON en el servicio REST.

<sup>51</sup> Fuente: Elaboración propia.

## 6.4. Creación del servicio API REST mediante Jersey y JAX-RS.

Para la implementación del servicio **REST**, se ha llevado a cabo la creación de un proyecto **Java** con **Maven** con arquitectura de aplicación web mediante el entorno de desarrollo integrado gratuito **Eclipse EE**.

### 6.4.1. Maven y Eclipse EE

**Maven** [37] es una herramienta para la gestión y construcción de proyectos **Java**, basado en **XML**, siendo su principal herramienta de configuración el archivo **POM.xml**. En este archivo se incluye toda la información necesaria para poder construir el proyecto.

Una de las funciones más utilizadas de Maven es la gestión de dependencias. **Maven** nos permite establecer un listado de todas las dependencias externas utilizadas por nuestro proyecto (como **Jersey**) en nuestro archivo **POM.xml**, encargándose incluso, a su vez, de las dependencias que tengan esas dependencias externas (dependencias transitivas).

**Maven** nos va a permitir simplificar la gestión de dependencias y la construcción de aplicaciones web basadas en **Frameworks**, como es nuestro proyecto.

Por otro lado, hemos utilizado **Eclipse EE** [38] como entorno de desarrollo integrado para el desarrollo de nuestro servicio **REST**. **Eclipse EE** nos ofrece un gran conjunto de herramientas (incluida **Maven**) para el desarrollo de aplicaciones **Web** mediante Java.

### 6.4.2. Incluir dependencia Jersey en nuestro proyecto.

En primer lugar, hemos incorporado las dependencias de Jersey de Maven en nuestro archivo **POM.xml**. De este modo, podremos acceder a las funcionalidades de Jersey. En la figura 6.4<sup>52</sup>, podemos observar cómo se han establecido las dependencias en nuestro **POM.xml**.

---

<sup>52</sup> Fuente: Elaboración propia.

```

<dependencies>

  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.glassfish.jersey.containers/jersey-container-servlet-core -->
  <dependency>
    <groupId>org.glassfish.jersey.containers</groupId>
    <artifactId>jersey-container-servlet-core</artifactId>
    <version>3.0.0</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.glassfish.jersey.media/jersey-media-json-binding -->
  <dependency>
    <groupId>org.glassfish.jersey.media</groupId>
    <artifactId>jersey-media-json-binding</artifactId>
    <version>3.0.0</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.glassfish.jersey.inject/jersey-hk2 -->
  <dependency>
    <groupId>org.glassfish.jersey.inject</groupId>
    <artifactId>jersey-hk2</artifactId>
    <version>3.0.0</version>
  </dependency>

</dependencies>

```

Figura 6.4. Dependencias incluidas en nuestro fichero POM.xml.

Incluyendo esas dependencias, tendríamos todo lo necesario para poder empezar a desarrollar nuestro servicio REST.

### 6.4.3. Desarrollo del servicio mediante las anotaciones JAX-RS.

A continuación, definimos la clase que responderá a las peticiones ejecutadas sobre nuestra **API REST**. Esta clase definirá los métodos **HTTP (GET, POST, PUT, DELETE)** necesarios para la funcionalidad del sistema.

Referentes a los métodos anteriores, también podremos especificar el tipo de datos que van a recibir en la solicitud, la ruta para solicitar ese método, el tipo de respuesta que el método va a realizar. Nuestro servicio solo soportará un método, este método presenta las siguientes características:

- Tipo: **POST**
- Ruta servicio: **http://localhost:8080/edcirest\_api/rest/convert\_edci**

- Formato de datos de envío: **Archivo JSON**.
- Formato de datos de respuesta: **Archivo Excel**

En la figura 6.5<sup>53</sup>, podemos observar la implementación de este servicio mediante las anotaciones JAX-RS.

```
package es.unex.cum.edci.rest.service;

import java.io.File;

import es.unex.cum.edci.rest.VO.CredencialVO;
import es.unex.cum.edci.rest.module.EDCIConverter;
import jakarta.ws.rs.Consumes;
import jakarta.ws.rs.POST;
import jakarta.ws.rs.Path;
import jakarta.ws.rs.Produces;
import jakarta.ws.rs.core.MediaType;
import jakarta.ws.rs.core.Response;
import jakarta.ws.rs.core.Response.ResponseBuilder;

@Path("/")
public class EdciconverterService {

    private EDCIConverter edciconverter = new EDCIConverter();

    @POST
    @Path("/convert_edci")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces("application/vnd.ms-excel")
    public Response getJson(CredencialVO json) { // Objeto contenedor del JSON.

        System.out.println("Hemos recibido una petición");
        edciconverter.generarPlantilla(json);

        // Después de generar la plantilla devolvemos el fichero Excel generado
        File edci_credentials = new File("C:/Europass_apirest/plantillas/EDC_template.xlsx");
        ResponseBuilder rb = Response.ok(edci_credentials);
        rb.header("Content-Disposition", "attachment; filename=new-excel-file.xlsx");
        return rb.build();
    }
}
}
```

Componente encargado de procesar e invocar al VBScript

Configuración del método POST que atenderá nuestro servicio.

Figura 6.5. Implementación del servicio de generación de credenciales EDC mediante JAX-RS.

**JAX-RS** nos permite automáticamente transformar los campos de un fichero **JSON** a una clase Java (En nuestro caso, la clase contenedor recibe el nombre de **CredencialVO**).

Esta clase debe tener los mismos atributos y del mismo tipo que están presentes en el fichero **JSON**. Cuando se reciba el **JSON**, los valores de los campos automáticamente se transferirán a los atributos de nuestra clase contenedor.

<sup>53</sup> Fuente: Elaboración propia.

En la figura 6.6<sup>54</sup>, podemos observar un ejemplo del formato de archivo JSON que emite el sistema de la información del SOFD de la Universidad de Extremadura para la emisión de las credenciales.

```
{
  "numcert": 2,
  "certificados": [
    {
      "dni": "28960000",
      "nombre": "D. RICHARD CORCHERO VÉLEZ",
      "denominacion": "JORNADA DE FORMACIÓN-DIFUSIÓN DE LA OFICINA DE CALIDAD",
      "campus": "CÁCERES",
      "zona": "CÁCERES",
      "trato": "D.",
      "dias_imparte": "el día 13 de marzo de 2015",
      "anio": "2015",
      "numero_horas": "4 horas presenciales",
      "curso": "2014-2015",
      "email": "ricorcher@gmail.com",
      "wallet": "n007pgvw"
    },
    {
      "dni": "28960001",
      "nombre": "D. HÉCTOR SÁNCHEZ SANTAMARÍA",
      "denominacion": "JORNADA DE FORMACIÓN-DIFUSIÓN DE LA OFICINA DE CALIDAD",
      "campus": "CÁCERES",
      "zona": "CÁCERES",
      "trato": "D.",
      "dias_imparte": "el día 13 de marzo de 2015",
      "anio": "2015",
      "numero_horas": "4 horas presenciales",
      "curso": "2014-2015",
      "email": "sasah@unex.es",
      "wallet": "n0061wmx"
    }
  ],
  "firma1": {
    "nombre": "Jesús Sánchez Martín",
    "cargo": "Director del Servicio de Orientación y Formación Docente"
  },
  "firma2": {
    "nombre": "José Luis Gurria Gascón",
    "cargo": "Vicerrector de Planificación Académica"
  },
  "reverso": {
    "denominacion": "LOS PROCEDIMIENTOS DE ACREDITACIÓN DE LA ANECA PARA EL ACCESO A PLAZAS DE PROFESORADO UNIVERSITARIO (PEP: PAD, PCD, PUP)",
    "apellido1": "BLANCO",
    "apellido2": "FERNÁNDEZ",
    "nombre": "GERARDO",
    "numero_horas": "6 HORAS ( 5 horas presenciales/ 1 hora no presencial)",
    "metodologia": "Detección inicial de necesidades.\r\n\r\nExposición más taller práctico.\r\n\r\nDiseño de itinerario de acreditación favorable.\r\n\r\n",
    "objetivos": "1. Conocer el procedimiento de acreditación de profesorado de la ANECA para los Cuerpos de Funcionarios Docentes Universitarios (ACADEMIA),",
    "contenidos": "1. Legislación vigente.\r\n\r\n2. Procedimiento de acreditación (evaluación previa, documental, plataforma, solicitud).\r\n\r\n3. Indicios"
  }
}
```

Figura 6.6. Formato de envío JSON de los datos referentes al lote de credenciales a generar.

En la figura 6.7<sup>55</sup>, podemos observar un diagrama de la clase **CredencialVO**, encargada de contener el **JSON** enviado por el cliente, y las subclases que la componen, junto con sus atributos.

<sup>54</sup> Fuente: Elaboración propia.

<sup>55</sup> Fuente: Elaboración propia.

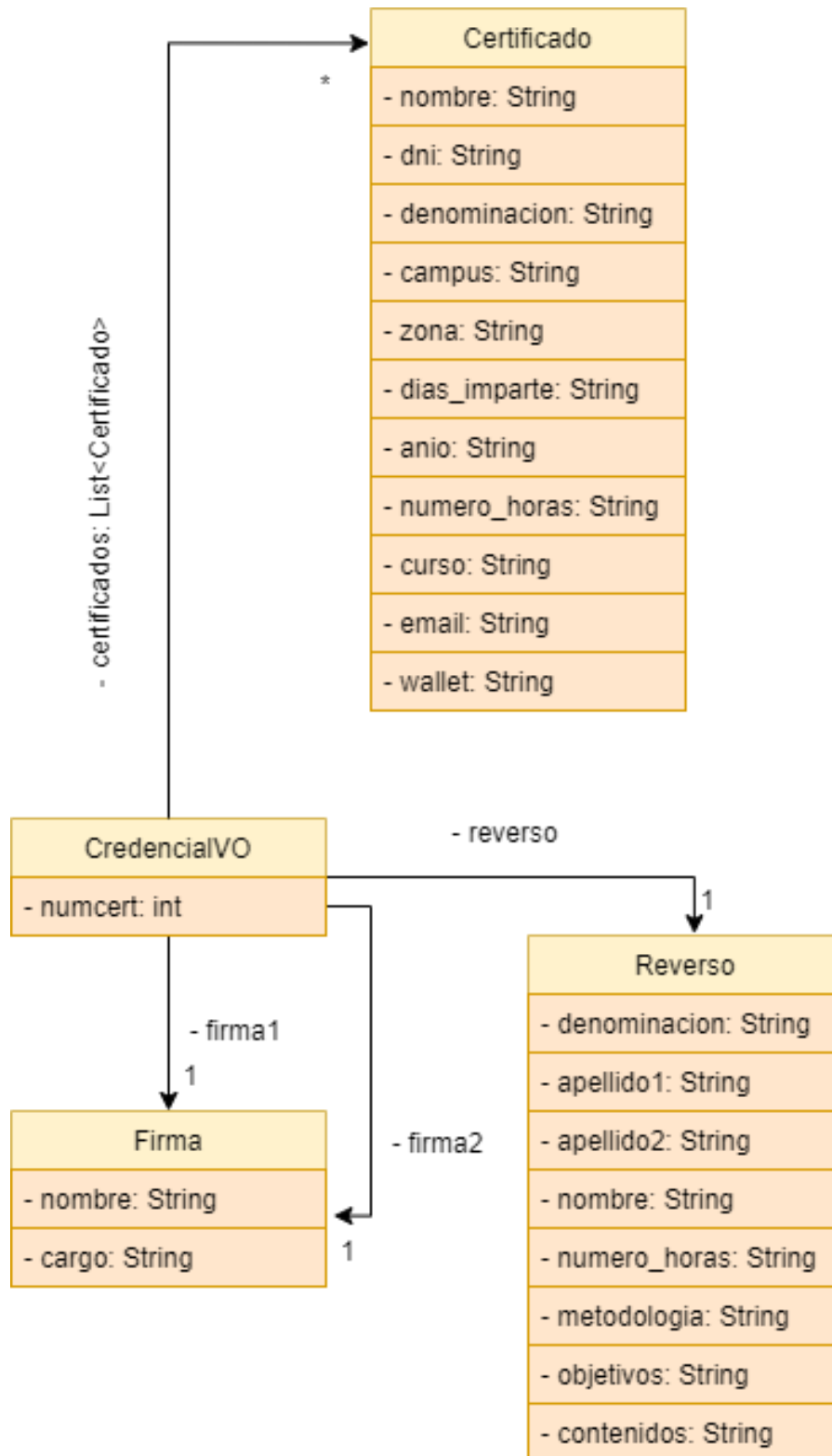


Figura 6.7. Diagrama de clase, del conjunto de componentes, necesitados para recoger el fichero JSON de la solicitud del cliente.



Para finalizar la parte de implementación del servicio REST con Jersey. Debemos establecer en el fichero **web.xml** de nuestro proyecto una serie de condiciones.

El fichero **web.xml** [39], proporciona información sobre configuración y despliegue de los componentes web que componen una aplicación web. En este archivo debemos especificar el Servlet Jersey, encargado de despachar las solicitudes http, y la ruta hacia el mismo.

En la figura 6.8<sup>56</sup>, podemos observar cómo quedaría nuestro fichero web.xml tras la configuración anterior.

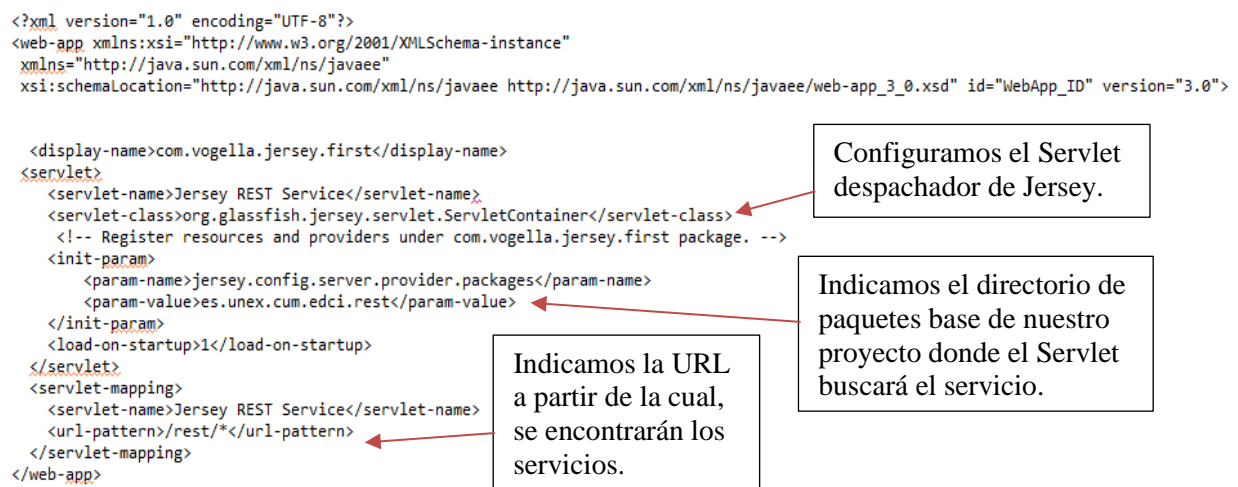


Figura 6.8. Configuración del archivo web.xml con Jersey.

## 6.5. Procesamiento del JSON y llamada a VBScript

El principal componente del que hace uso nuestro servicio es **EDCICConverter**. Este componente es el encargado de procesar el **JSON** y ejecutar la llamada al sistema para invocar el **VBScript**. El comportamiento de este componente se detalla extensamente en el anexo I.

El principal método de este componente recibe el nombre de “**generarPlantilla**”. Este método recibe como parámetro el **JSON** contenido dentro del objeto **CredencialVO**, procesa ese **JSON** obteniendo los datos de las credenciales y el formato **HTML** de las mismas y finalmente, invoca al **VBScript** para generar el nuevo lote de credenciales **EDC**. En el algoritmo 6.4 podemos observar la implementación de este método.

<sup>56</sup> Fuente: Elaboración propia.

```

public void generarPlantilla(CredencialVO json){

    // Procesamos el JSON recibido y obtenemos los datos de las credenciales
    // y los formatos de estas.
    String[] data = procesarJSON(json);

    // Invocamos el script que ejecuta la macro.
    String[] cmd = {"WScript",
        workingdirectory+"ScriptMacroExcell.vbs",
        workingdirectory,
        generic_file,
        new_file,
        Integer.toString(json.getNumcert()-1), // Por defecto, ya tenemos una fila desbloqueada.
        data[0],
        data[1]};

    // Ejecutamos el comando Visual Basic Script
    Process p = null;
    try {
        p = Runtime.getRuntime().exec(cmd);
        // Esperamos a que el comando finalice para poder devolver el fichero generado.
        p.waitFor();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

```

Algoritmo 6.4. Implementación de la función que genera el proceso de construcción del lote de credenciales EDC.

Dentro de esta función podemos destacar dos partes:

- El procesamiento del **JSON**, en un formato que pueda ser entendido por **Visual Basic Script**.
- La invocación del **VBScript**, mediante una llamada al sistema, para generar el lote de credenciales.

El método encargado de procesar el **JSON** recibe el nombre de “**procesarJSON**”. Este método devuelve un array con dos cadenas de información en formato texto, delimitado o marcados por diferentes caracteres, de modo, que el script de **Visual Basic** pueda separarlos y recorrerlos para completar la plantilla **Excel**. En el anexo I, se detalla de manera más exacta los métodos anteriores y los métodos que son invocados desde estos.

## 6.6. Despliegue, Pruebas y validación del servicio REST

Finalmente, probaremos si el servicio cumple con su cometido, mediante un despliegue en local, haciendo uso de **Apache Tomcat** en su versión 10. **Apache Tomcat** [40] es un contenedor de Java Servlet , desarrollado por la Fundación Apache, que nos permite crear un servidor web.

Además, se ha implementado un cliente web básico que nos permite introducir el archivo **JSON** en el formato expedido por el **SOFD** y enviarlo mediante una solicitud **HTTP POST** al servicio **REST**. En la figura 6.8, podemos observar el cliente web implementado.

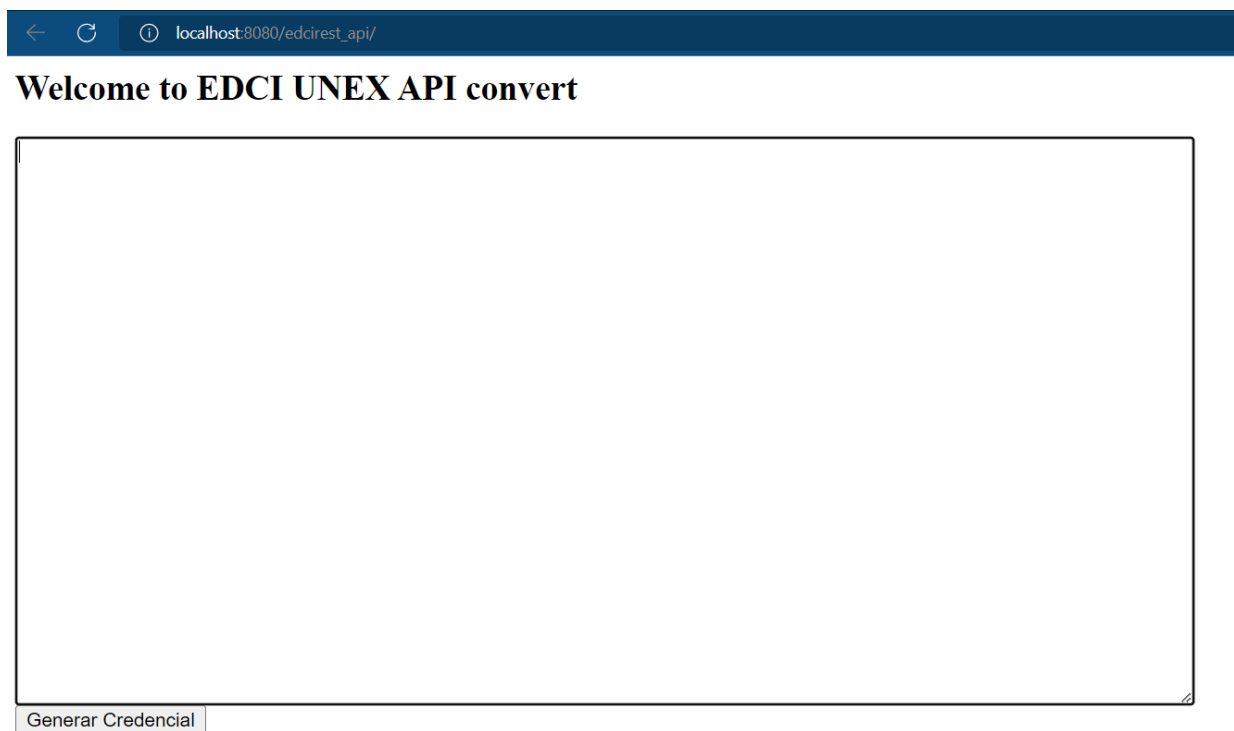


Figura 6.9. Cliente web implementado para envío de los datos de las credenciales en formato JSON.

Este cliente web, tiene embebida en su código HTML, una función JavaScript asociada al botón “**Generar Credenciales**”. En la figura 6.10, podemos observar la estructura HTML del cliente y el código JavaScript utilizado para la solicitud **HTTP POST**.

```

<html>
<script>
function myfuncion(){

    var textarea = document.getElementById("json_input"); // Accedemos al textarea
    var json = JSON.parse(textarea.value) // Recuperamos el valor y lo pasamos de texto_plano a JSON
    var fileName = 'edci_credenciales.xlsx' // Establecemos el nombre con que queremos guardar el nuevo lote de credenciales.

    // Comenzamos a preparar la solicitud HTTP
    var request = new XMLHttpRequest();
    request.open('POST', 'http://localhost:8080/edci_rest_api/rest/convert_edci',true) // Método POST a la dirección de nuestro servicio REST.
    request.setRequestHeader('Content-Type', 'application/json') // Establecemos que los datos van a ir en formato JSON
    var body = JSON.stringify(json) // JSON como cuerpo de la solicitud.
    // Podemos visualizarlo para ver el formato
    console.log(body)
    request.responseType = 'blob' // Establecemos que la respuesta va a ser de tipo blob (Archivo)

    // Programamos las operaciones que realizaremos tras recibir la respuesta.
    request.onload = function(e) {
        if (this.status === 200) { // Si el servidor nos responde
            var blob = this.response;
            if(window.navigator.msSaveOrOpenBlob) { // Pediremos al navegador que guarde el fichero con el nombre del principio.
                window.navigator.msSaveBlob(blob, fileName);
            }
            else{
                // En caso de que el navegador no soporte la función anterior, lo haremos manualmente.
                var downloadLink = window.document.createElement('a');
                var contentTypeHeader = request.getResponseHeader("Content-Type");
                downloadLink.href = window.URL.createObjectURL(new Blob([blob], { type: contentTypeHeader }));
                downloadLink.download = fileName;
                document.body.appendChild(downloadLink);
                downloadLink.click();
                document.body.removeChild(downloadLink);
            }
        }
    };
    request.send(body); // Enviamos la solicitud y el archivo JSON.
}
</script>
<body>
<h2>Welcome to EDCI UNEX API convert</h2>
<textarea id="json_input" rows="4" cols="50"></textarea>
<br>
<button onclick="myfuncion()" > Generar Credencial</button>
</body>
</html>

```

Figura 6.10. Estructura HTML y código JavaScript de nuestro cliente web.

Finalmente, pegamos el fichero JSON en el área de texto y pulsamos el botón **“Generar credencial”**. El servicio **REST**, debería poder devolver el lote de credenciales generado a nuestro cliente web y este lo descargara automáticamente. En la figura 6.11, podemos observar como el servicio responde al cliente, devolviendo el lote de credenciales Excel EDC.

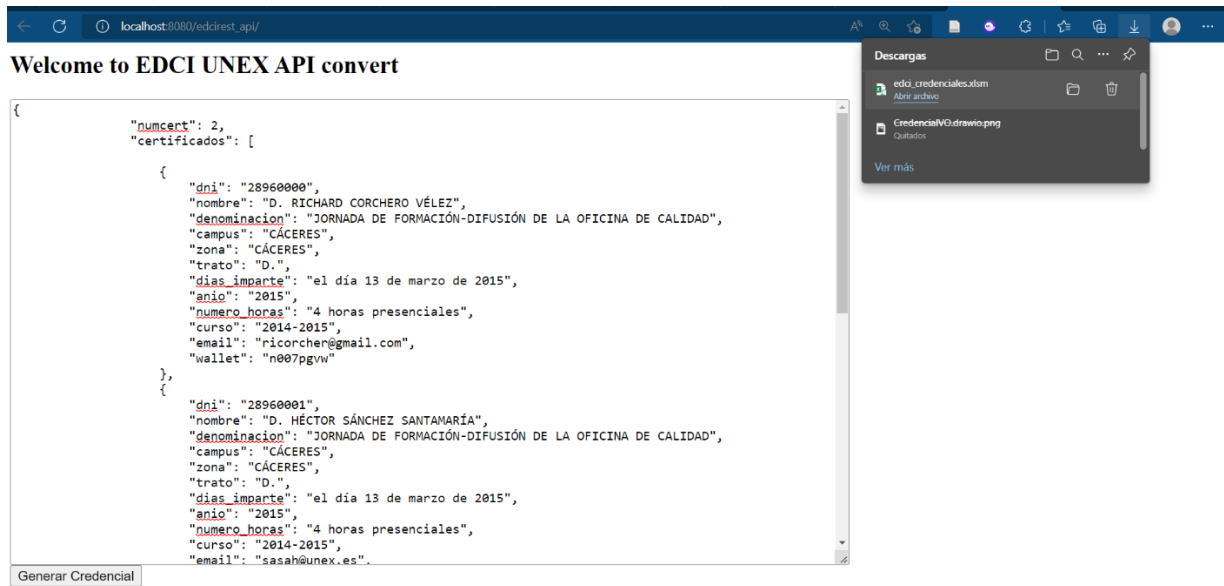


Figura 6.11. Respuesta con éxito de la solicitud a nuestro servicio Rest.

Por último, el usuario de la aplicación cliente, podría emitir el lote de credenciales mediante el servicio EDCI Issuer del portal Europass. En la figura 6.12<sup>57</sup>, se muestra el formato de la credencial obtenida del SOFD tras ser sellada, valida y enviada a la wallet del usuario.

<sup>57</sup> Fuente: Elaboración propia.

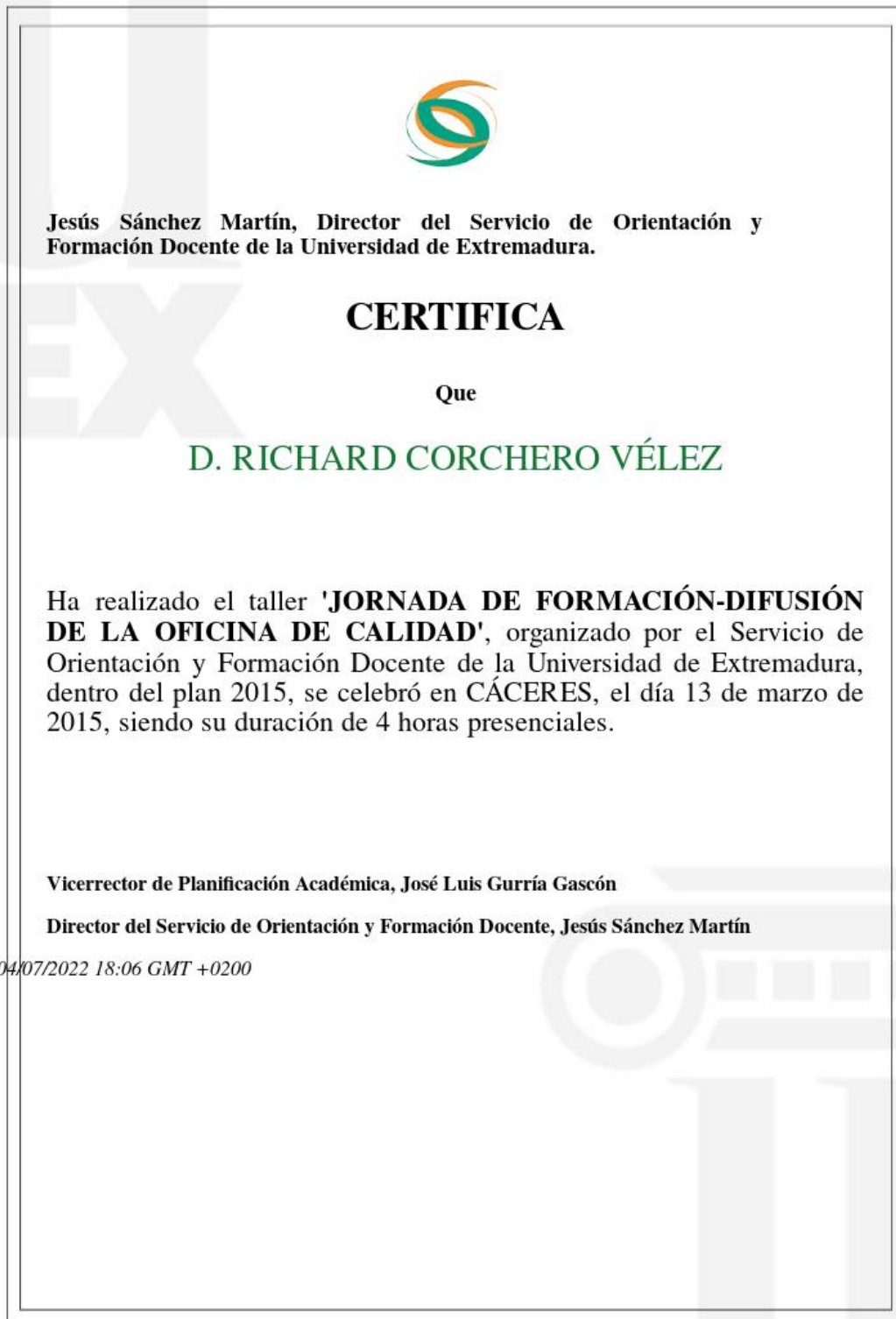


Figura 6.12. Credencial validada, sellada y emitida en la plataforma Europass del SOFD.



# Capítulo 7

## Conclusiones

### Contenidos

---

7.1. Conclusiones. ....	92
7.2. Trabajos futuros. ....	94

---

En este último capítulo, se expondrán las principales conclusiones obtenidas tras la realización de este TFG, así como las posibles vías de continuidad que pueden plantearse tras la realización de este.

En la sección 7.1 hablaremos de las conclusiones obtenidas tras la realización del proyecto.

En la sección 7.2 se plantearán algunos posibles trabajos futuros que pueden plantearse, teniendo este TFG como punto de partida.

### 7.1. Conclusiones.

El principal objetivo de este **TFG** consistía en diseñar y desarrollar un aplicativo que permita automatizar la emisión de certificados tras la realización exitosa de una formación no reglada siguiendo el estándar **EDCI (Europass Digital Credentials Infrastructure)** de la plataforma Europass. El funcionamiento del Servicio ha sido probado en local, siendo exitoso su resultado.

El Servicio **REST** es capaz de atender o despachar solicitudes **HTTP POST** que transportan en su cuerpo datos, en formato **JSON**, de las credenciales que el cliente desea emitir en el estándar **EDCI**, los cuales, deben tener una estructura específica (Formato de datos **JSON** del **SOFD** de la **Universidad de Extremadura**).



Posteriormente, el servicio es capaz de procesar esa información y generar una plantilla **Excel EDC** cumplimentada y con un formato visual personalizado, mediante los datos enviados.

Finalmente, el sistema es capaz de devolver el nuevo fichero **Excel EDC** generado y cumplimentado, listo para subir al servicio **EDCI Issuer** de la plataforma Europass, donde el cliente, puede validar, sellar y emitir las credenciales a las carteras de los alumnos, usuarios de la plataforma Europass.

Para lograr este objetivo, se han realizado los siguientes subobjetivos o objetivos específicos:

- Se ha realizado un estudio detallado del estándar **EDCI** y de la plataforma **Europass**. Se detalló los componentes y principales partes de este estándar **EDCI**, así como, el funcionamiento de la plataforma **Europass** y **EDCI Issuer**.
- Se realizó un estudio y diseño de las tecnologías necesarias para poder llevar a cabo la implementación del aplicativo, así como de las diferentes alternativas que fueron planteadas.
- Se llevó a cabo el diseño e implementación del algoritmo que permite generar las credenciales digitales, mediante el lenguaje **VBScript** para manipular la plantilla **Excel EDC**.
- Se llevó a cabo la implementación y despliegue del servicio **REST** que atiende las solicitudes de los clientes y devuelve la plantilla **Excel EDC** con la información cumplimentada. Además, se consiguió integrar el algoritmo de generación de credenciales con el servicio **REST**.
- Se consiguió diseñar un formato personalizado de plantilla **HTML** para la emisión de las credenciales del **SOFD** de la **Universidad de Extremadura**.
- Se validó que los lotes de credenciales generados por el servicio **REST** son compatibles con la plataforma Europass y pueden ser sellados y enviados a través de esta.

Tras analizar el objetivo principal como los subjetivos, podemos considerar que se han cumplido correctamente. Aunque este sistema solo es una primera versión, este **TFG** supone un acercamiento para las instituciones y en especial a la **Universidad de Extremadura** para la emisión de sus títulos o credenciales en la plataforma europea, otorgándoles un valor añadido.

## 7.2. Trabajos futuros.

Podemos plantear diferentes enfoques de trabajo futuro a partir de este TFG:

- **Integración del servicio para ser la nueva forma de emitir credenciales en la Universidad de Extremadura (UNEX):** Se podía seguir llevando a cabo pruebas e integrando de este servicio con **SOFD** de la **UNEX** y posteriormente, una vez que se tenga una versión sólida, extender este sistema al resto de servicios de emisión de credenciales de la **UNEX**.
- **Añadir la funcionalidad de poder emitir los certificados en la Blockchain:** La tecnología **Blockchain** [41] o cadena de bloques en español, está siendo cada vez más popular en el mundo tecnológico. La **Blockchain** es una lista digital compuesta por muchos bloques de datos, organizados cronológicamente y vinculados entre sí. Dentro de la **Blockchain**, las transacciones ocurren en una red de nodos, manteniendo cada uno de estos una copia de la cadena de bloques, de manera, que cada uno de estos nodos chequea las transacciones de la **Blockchain** para verificarlas y evitar modificaciones o fraudes. De esta manera, la información contenida en un bloque solo puede ser editada o modificada, si modificamos todos los bloques. Utilizar esta tecnología para emitir credenciales mediante alguna **Blockchain** pública, sería una buena forma de luchar contra el fraude y la falsificación de credenciales digitales. Siendo otro camino que permitiría generar documentos inmutables y originales.



---

## Bibliografía.

- [1] «Statistics | Europass». <https://europa.eu/europass/en/statistics> (accedido 5 de julio de 2022).
- [2] «europass\_background-info\_framework-digitally-signed-credentials.pdf». Accedido: 19 de junio de 2022. [En línea]. Disponible en: [https://ec.europa.eu/futurium/en/system/files/ged/europass\\_background-info\\_framework-digitally-signed-credentials.pdf](https://ec.europa.eu/futurium/en/system/files/ged/europass_background-info_framework-digitally-signed-credentials.pdf)
- [3] «2015 - The European Banking Union A Compendium.pdf». Accedido: 19 de junio de 2022. [En línea]. Disponible en: [https://ec.europa.eu/futurium/en/system/files/ged/eidas\\_regulation.pdf](https://ec.europa.eu/futurium/en/system/files/ged/eidas_regulation.pdf)
- [4] «European Single Market», *Wikipedia*. 12 de junio de 2022. Accedido: 19 de junio de 2022. [En línea]. Disponible en: [https://en.wikipedia.org/w/index.php?title=European\\_Single\\_Market&oldid=1092717601](https://en.wikipedia.org/w/index.php?title=European_Single_Market&oldid=1092717601)
- [5] «Verifiable Credentials Data Model v1.1». <https://www.w3.org/TR/vc-data-model/> (accedido 28 de junio de 2022).
- [6] «Extensible Markup Language (XML)». <https://www.w3.org/XML/> (accedido 28 de junio de 2022).
- [7] «The European Qualifications Framework (EQF) | Europass». <https://europa.eu/europass/en/european-qualifications-framework-eqf> (accedido 28 de junio de 2022).
- [8] «Preparación de credenciales | Credenciales digitales europeas para el aprendizaje | Europass». <https://europa.eu/europass/en/preparing-credentials-european-digital-credentials-learning> (accedido 23 de junio de 2022).
- [9] «HTML Standard». <https://html.spec.whatwg.org/#toc-introduction> (accedido 28 de junio de 2022).
- [10] «Thymeleaf». <https://www.thymeleaf.org/> (accedido 28 de junio de 2022).
- [11] «Desarrollo iterativo e incremental», *Proyectos Ágiles*, 27 de septiembre de 2008. <https://proyectosagiles.org/desarrollo-iterativo-incremental/> (accedido 2 de julio de 2022).
- [12] «Características y fases del modelo incremental», *OBS Business School*. <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental> (accedido 2 de julio de 2022).
- [13] «Metodologías orientadas a objetos». <https://www.careerride.com/page/object-oriented-methodologies-642.aspx> (accedido 2 de julio de 2022).

- 
- [14] «Que es UML | Lenguaje de modelado unificado».  
<https://www.uml.org/what-is-uml.htm> (accedido 2 de julio de 2022).
- [15] «Desarrollo de software orientado a objetos - Blog TECH México University». <https://www.techtute.com/mx/informatica/blog/desarrollo-software-orientado-objetos> (accedido 2 de julio de 2022).
- [16] «PMBOK7.pdf». Accedido: 3 de julio de 2022. [En línea]. Disponible en: <https://iehouse.org/wp-content/uploads/2021/07/PMBOK7.pdf>
- [17] baeldung, «Working with Microsoft Excel in Java | Baeldung», 7 de febrero de 2017. <https://www.baeldung.com/java-microsoft-excel> (accedido 6 de julio de 2022).
- [18] «Apache POI - the Java API for Microsoft Documents».  
<https://poi.apache.org/index.html> (accedido 19 de abril de 2022).
- [19] «Apache POI - Component Overview».  
<https://poi.apache.org/components/index.html> (accedido 19 de abril de 2022).
- [20] «Office Open XML», *Wikipedia, la enciclopedia libre*. 21 de abril de 2022. Accedido: 3 de julio de 2022. [En línea]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Office\\_Open\\_XML&oldid=143048880](https://es.wikipedia.org/w/index.php?title=Office_Open_XML&oldid=143048880)
- [21] P. Charollais, «ECMA-376», *Ecma International*. <https://www.ecma-international.org/publications-and-standards/standards/ecma-376/> (accedido 3 de julio de 2022).
- [22] 14:00-17:00, «ISO/IEC 29500-1:2016», *ISO*.  
<https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/16/71691.html> (accedido 3 de julio de 2022).
- [23] «VBScript». [https://docs.microsoft.com/en-us/previous-versions/t0aew7h6\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/t0aew7h6(v=vs.85)) (accedido 3 de julio de 2022).
- [24] dotnet-bot, «Documentos de Visual Basic: inicio, tutoriales y referencias.» <https://docs.microsoft.com/es-es/dotnet/visual-basic/> (accedido 3 de julio de 2022).
- [25] JasonGerend, «wscript». <https://docs.microsoft.com/es-es/windows-server/administration/windows-commands/wscript> (accedido 3 de julio de 2022).
- [26] «rest-apis», 26 de agosto de 2021. <https://www.ibm.com/es-es/cloud/learn/rest-apis> (accedido 3 de julio de 2022).
- [27] R. Fielding, M. Nottingham, y J. Reschke, «RFC9112», *IETF HTTP Working Group Specifications*, junio de 2022.  
<https://httpwg.org/specs/rfc9112.html> (accedido 3 de julio de 2022).
- [28] «What is REST API | PHPenthusiast».  
<https://phpenthusiast.com/blog/what-is-rest-api> (accedido 6 de julio de 2022).
- [29] «La API de Java para servicios web RESTful (JAX-RS): cree rápidamente

- 
- servicios web ligeros». <https://www.oracle.com/technical-resources/articles/java/jax-rs.html> (accedido 3 de julio de 2022).
- [30] Anonymous, «Eclipse Jersey™», *projects.eclipse.org*, 3 de enero de 2018. <https://projects.eclipse.org/projects/ee4j.jersey> (accedido 3 de julio de 2022).
- [31] «ISO 216:2007(en), Writing paper and certain classes of printed matter — Trimmed sizes — A and B series, and indication of machine direction». <https://www.iso.org/obp/ui/#iso:std:iso:216:ed-2:v1:en> (accedido 3 de julio de 2022).
- [32] «EDCI-Diplomawildcards-171120-1131-2172.pdf». Accedido: 3 de julio de 2022. [En línea]. Disponible en: <https://europa.eu/europass/system/files/2020-11/EDCI-Diplomawildcards-171120-1131-2172.pdf>
- [33] «XPath Tutorial». [https://www.w3schools.com/xml/xpath\\_intro.asp](https://www.w3schools.com/xml/xpath_intro.asp) (accedido 3 de julio de 2022).
- [34] L. Gupta, «JSON vs XML», *REST API Tutorial*, 30 de abril de 2018. <https://restfulapi.net/json-vs-xml/> (accedido 3 de julio de 2022).
- [35] «JSON». <https://www.json.org/json-en.html> (accedido 3 de julio de 2022).
- [36] Y. Fernández, «Macros de Excel: qué son, cómo funcionan y cómo crearlos», *Xataka*, 13 de febrero de 2021. <https://www.xataka.com/basics/macros-excel-que-como-funcionan-como-crearlos> (accedido 4 de julio de 2022).
- [37] «Maven – Introducción». <https://maven.apache.org/what-is-maven.html> (accedido 4 de julio de 2022).
- [38] «Eclipse IDE para desarrolladores de Java EE | Paquetes Eclipse». <https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-java-ee-developers> (accedido 4 de julio de 2022).
- [39] «IBM Docs», 21 de marzo de 2022. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/was-nd/8.5.5?topic=applications-webxml-file> (accedido 4 de julio de 2022).
- [40] «Apache Tomcat® - Welcome!» <https://tomcat.apache.org/> (accedido 4 de julio de 2022).
- [41] «Blockchain», *Binance Academy*. <https://academy.binance.com/en/glossary/blockchain> (accedido 4 de julio de 2022).



# Capítulo 8

## Anexo I. Manual del programador

### Contenidos

---

<b>8.1. Tecnologías y Herramientas empleadas .....</b>	<b>101</b>
<b>8.2. Componentes y Estructura .....</b>	<b>101</b>
8.2.1. VBScript.....	102
8.2.2. Servicio REST.....	102
<b>8.3. Variables .....</b>	<b>104</b>
8.3.1. VBScript.....	104
8.3.2. EDCIConverter .....	105
<b>8.4. Funciones y Métodos.....</b>	<b>107</b>
8.4.1. VBScript Funciones.....	107
8.4.2. VBScript Algoritmos.....	108
<b>8.4.3. EDCIConverter .....</b>	<b>109</b>

---

En este anexo se detallan, de manera más exacta, las partes fundamentales del sistema desarrollado, para futuros desarrolladores que vayan a trabajar a partir de este proyecto.

En la sección 8.1, se repasarán las tecnologías empleadas y se indicarán las versiones de estas.

En la sección 8.2, se expondrán los componentes que forman el sistema y su estructura.

En la sección 8.3, se detallan el uso de las variables más importantes dentro de los distintos componentes software implementados.

Finalmente, en la sección 8.4, se detallan los principales métodos, funciones y algoritmos que son indispensables para el funcionamiento de nuestro servicio.



## 8.1. Tecnologías y Herramientas empleadas

Para poder llevar a cabo el desarrollo de este TFG, se han utilizado las diferentes tecnologías y herramientas:

- **Java Development Kit (JDK)**, en su versión **1.8.0\_202**.
- **Eclipse IDE for Enterprise Java and Web Developers (Eclipse EE)**, en su versión **2022-06 (4.24.0)**.
- **Visual Studio Code**, en su versión **1.68.1**.
- **Apache Tomcat**, en su versión **10.0.22**.
- **Dependencias de Jersey RESTful Web Services in Java**, versión **3.0.0** (**jersey-container-servlet-core**, **jersey-media-json-binding**, **jersey-hk2**).
- **HTML5, CSS3 y JavaScript**.
- **Windows Script (WScript)** en su versión **5.812**.
- **Visual Basic Script (VBScript)** en su versión **5.8**.
- **Sistema Operativo Windows 10 Home**, versión: **21H1**

## 8.2. Componentes y Estructura

Nuestro sistema podría dividirse en dos componentes software principalmente:

- El primer componente sería el **VBScript** que interactúa con la plantilla **Excel EDC**. Este script nos permitirá generar un nuevo archivo Excel EDC, relleno con la información procedente del cliente. Esta información ha sido procesada previamente por el servicio **REST**, de modo que el script pueda entenderla y automatizar la generación del nuevo lote de credenciales.
- El segundo componente sería el servicio **REST** desarrollado con **Java** y el **Framework Jersey**. Este componente permite la comunicación entre las instituciones y nuestro servicio. Este servicio es el encargado de atender las solicitudes de los clientes que quieren generar sus credenciales en el formato **EDCI**, procesar la información enviada por el cliente en formato **JSON**, pasar la información procesada al **VBScript**, invocar el **VBScript** para que genere el nuevo lote de credenciales y finalmente, devolver el nuevo lote de credenciales generado.

### **8.2.1. VBScript**

La funcionalidad de la generación de un nuevo lote de credenciales EDC, partiendo de la información enviada por la institución y la plantilla Excel EDC con las macros adaptadas, recae sobre un único fichero escrito mediante el lenguaje Visual Basic Script (extensión del fichero **.vbs**). Este script deberá encontrarse en el mismo directorio que la plantilla Excel EDC genérica.

La estructura de este archivo podemos observarla en el algoritmo 6.3 del capítulo 6. En el resto de las secciones se describirán sus variables y métodos detenidamente.

### **8.2.2. Servicio REST**

El servicio REST es el encargado de atender las solicitudes de los clientes, procesar la información JSON en un formato entendible para el VBScript, llevar a cabo la invocación de este y finalmente, responder a lo solicitud del cliente con el nuevo lote de credenciales Excel EDC generado.

En la figura 8.1, podemos observar la estructura de directorios, paquetes y archivos de configuración que conforman este servicio.

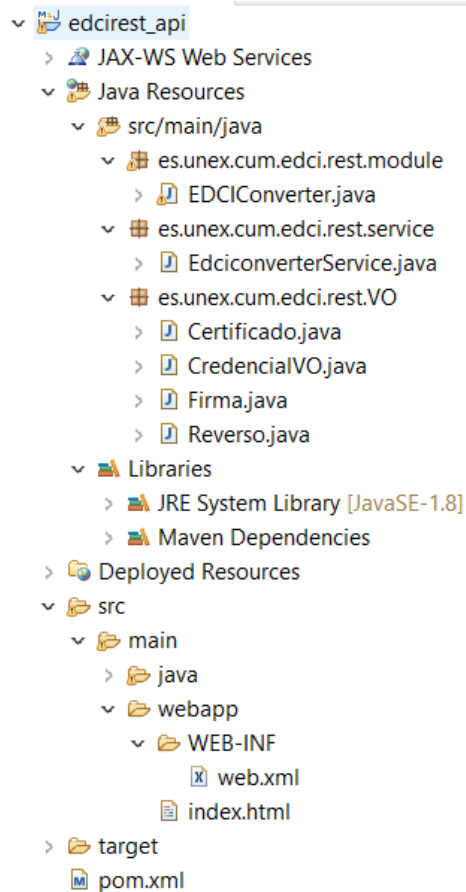


Figura 8.1. Estructura de los archivos que componen la API REST.

Podemos destacar los siguientes elementos de la imagen anterior:

- Fichero **Pom.xml**: Principal archivo de configuración de la herramienta Maven y en el cual hemos establecido las dependencias de Jersey (capítulo 6 sección 6.4).
- Fichero **Web.xml**: Archivo que contiene información los componentes web que van a ser desplegados. En este archivo, hemos configurado el Servlet de Jersey para despachar las solicitudes del servicio REST (capítulo 6 sección 6.4).
- Fichero **Index.html**: fichero que implementa el cliente utilizado para la validación del servicio Web (capítulo 6 sección 6.6).
- **Paquetes**: estructura de paquetes Java que conforman nuestro proyecto. El directorio raíz, del que cuelgan todos los paquetes, es “**es.unex.cum.edci**”

Si profundizamos en la estructura de paquetes, podemos observar los siguientes:

- Paquete “**module**”: Este paquete contiene los módulos de nuestro sistema. Estos módulos implementan las funcionalidades para cada uno de los servicios que se diseñen. En nuestro caso, tenemos un único módulo, el cual, es usado en nuestro servicio REST. Este módulo será el encargado de procesar la información en formato JSON e invocar al VBScript para la creación del nuevo lote de credenciales. Este módulo recibe el nombre de “**EDCConverter**”.
- Paquete “**service**”: Este paquete contiene los servicios que oferta nuestra API REST. Cada uno de estos servicios podrán tener sus propios métodos para responder a las solicitudes HTTP (GET, POST, PUT y DELETE). Estos servicios harán uso de los módulos para llevar a cabo su funcionalidad. En nuestro caso, solo tenemos un único servicio denominada “**EdcconverterService**”. Este servicio atenderá las solicitudes del cliente para la generación de lotes de credenciales Excel EDC, a partir de la información enviada por el cliente. Este servicio hará uso del módulo “**EDCConverter**”, para llevar a cabo su principal funcionalidad.
- Paquete “**VO**”: Este paquete contiene las clases que modelarán el formato JSON recibido en la solicitud al servicio. La clase principal que representa el modelo de datos JSON es “**CredencialVO**”. Esta clase a su vez está formado o compuesta por el resto de las clases (**Certificado.java**, **Firma.java**, **Reverso.java**), las cuales, modelan diferentes aspectos contenidos en el archivo JSON.

En la figura 6.7 del capítulo 6, podemos observar el diagrama de clases de los componentes del paquete “**VO**”, como se relación y la información que recogen.

## 8.3. Variables

A continuación, detallan las variables fundamentales del script de **Visual Basic** y del módulo “**EDCConverter**”.

### 8.3.1. VBScript

Podemos destacar las siguientes variables del script de Visual Basic para llevar a cabo la generación de las credenciales EDC:

- **Args**: esta variable contendrá el array de argumentos pasados al script. La definición de los argumentos que son enviados al script, podemos encontrarla en la sección 6.2 del capítulo 6.

- **objExcel**: Esta variable contendrá el objeto que interacciona con la aplicación Microsoft Excel. Mediante este objeto podremos abrir archivos Excel, rellenarlos y guardarlos de manera nativa.
- **objWorkbook**: Esta variable contendrá el objeto que representa el libro o archivo Excel que vamos a abrir. Este objeto también nos permitirá navegar entre las hojas del libro Excel. En nuestro caso, el libro Excel a utilizar será la plantilla de Europass para generar lotes de credenciales.
- **objSheet**: Esta variable contendrá el objeto que representa las hojas de un libro Excel. Este objeto contendrá la hoja actual sobre la que estamos trabajando dentro del Excel. Esta variable es vital para poder actuar sobre las distintas hojas que propone la plantilla Europass.
- **dic**: Esta variable es un diccionario que recoge las columnas de las celdas a rellenar, como un array de tipo entero, para cada una de las hojas del archivo Excel. Las claves del diccionario son los valores enteros de las hojas (siendo el valor 2, el de la hoja “**Credentials**”). Cada clave nos devuelve un vector o array de enteros, que representan las columnas de cada hoja.
- **html\_cell**: Variable entera que indica la columna de la celda, situada en la hoja “**Credentials**”, donde deberemos establecer el formato HTML de la credencial.
- **row\_inde\_init**: Variable entera que indica la fila de inicio, a partir de la cual, comenzamos a rellenar las filas del archivo.
- **formato\_html**: Vector o Array que contendrá en cada posición el código HTML para cada una de las credenciales que van a ser emitidas, en formato String. Esta información vendrá dada como argumento desde el servicio API REST y deberemos realizar el Split de esta, mediante un delimitador específico (En nuestro caso, “&SEPARADOR&”).
- **credenciales**: Vector o Array que contendrá en cada posición la información referente a una credencial (persona, organización, etc.), formato String. Esta información vendrá dada como argumento desde el servicio API REST y deberemos realizar el Split de esta, mediante un delimitador específico (En nuestro caso “]”).

### 8.3.2. EDCIConverter

Dentro del servicio REST, el principal componente que implementa la funcionalidad es el módulo “**EDCIConverter**”. Este módulo contiene una serie de constantes. En la figura 8.2, se detallan estas constantes.

```

/**
 *
 * Módulo encargado de llevar a cabo la generación del lote de credenciales Excel EDC.
 * Para ello, podemos destacar dos funcionalidades:
 *
 * 1. Procesado del JSON recibido, en un formato entendible por el script de Visual Basic.
 * 2. Invocación del VBScript y envío de la información como argumentos del mismo
 *
 *
 * @author Richard Corchero Velez
 * @version 1.0.0
 */
public class EDCConverter {

    // Variable que contiene la ruta absoluta al directorio donde se encuentran la plantilla Excel EDC y el script de Visual Basic.
    final private String workingdirectory = "c:\\Europass_apirest\\plantillas\\";
    // Variable que contiene el nombre la plantilla Excel EDC por defecto.
    final private String generic_file = "EDC_template-generic.xmlsm";
    // Variable que contiene el nombre de la nueva plantilla que vamos a generar.
    final private String new_file = "EDC_template.xmlsm";
    // Variable que establece el formato de la fecha de emisión de las credenciales.
    final private DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MM/yyyy");
    // Variable que contiene el formato HTML por defecto, que será modificado para cada uno de las credenciales.
    final private String html_template = "<div style='font-style: normal; padding-bottom: 2rem; padding-top: 6rem;'>"+
    "<div style='flex: 0 0 100%; max-width: 100%; display: block; text-align: center;'>"+
    "<img style='max-height: 5.5rem; width: auto;' src='[/eup:organization[@id=/eup:europassCredential/cred:issuer[@idref]/eup:logo$]'/>"+
    "<br/>"+
    "</div>"+
    "<div style='flex: 0 0 100%; max-width: 70%; display: block; text-align: justify; margin-left: 10%; margin-right: 10%;>"+
    "<p style='font-size: 18px; color: #000000;'>+

```

Figura 8.2. Declaración de constantes necesarias en el módulo EDCConverter.

El principal método de este módulo o el método que desencadena el resto de las funcionalidades recibe el nombre de “**generarPlantilla**”. Podemos observar la implementación de este método en la sección 6.5 del capítulo 6.

Dentro de este método, podemos destacar algunas variables locales de vital importancia:

- **data**: Array de tipo String que contiene la información procesada y lista para ser enviada al script de Visual Basic. Este array tiene dos posiciones. En la posición [0], contiene la información referente a los datos de las credenciales (titulo, nombre de la persona, organización, etc), mientras que, en la posición [1], contiene la información referente al formato visual **HTML** de la credencial.
- **cmd**: Array de tipo String que contiene la secuencia de comandos a ejecutar mediante la llamada al sistema operativo. En la posición [0], contiene la palabra reservada “**WScript**” para poder ejecutar el **VBScript**, la posición [1] contiene la ruta completa del script de Visual Basic que ejecutará y, en el resto de las posiciones, los argumentos necesarios para la ejecución del script.
- **p**: Objetivo de tipo Process, que permitirá la ejecución y captura de la llamada al sistema, durante la invocación del **VBScript**.

## 8.4. Funciones y Métodos

Finalmente, se detallarán las funciones y métodos implementadas dentro de **VBScript** y del módulo **EDCConverter**. Además, también se expondrán cuestiones relacionadas con la lógica de la implementación de estas.

### 8.4.1. VBScript Funciones

Las variables definidas en la sección 8.3.1, nos ofrecen una serie de funciones vitales para el desarrollo del script:

- **Open (String)**: Esta función, es uno de los métodos que nos ofrece el objeto de tipo “**Excel.Application**” contenido en la variable **objExcel**. Esta función recibe por parámetros un String que representa la ruta de acceso al archivo Excel que queremos abrir. Esta función devuelve un objeto de tipo “**Excel.Workbook**” que será contenido en la variable **objWorkbook**.
- **Run (String, [parametros])**: Esta función, es uno de los métodos que nos ofrece el objeto de tipo “**Excel.Application**” contenido en la variable **objExcel**. Esta función nos permite ejecutar las macros de un libro Excel que tengamos abierto previamente. Para ello, esta función recibe el nombre de la macro (**Sheetx.nombremacro**) y los parámetros que tenga la macro, si es que tienen. Esta función es vital para poder usar las macros de la plantilla **Excel EDC**.
- **Worksheets (int)**: Esta función es uno de los métodos que nos ofrece el objeto de tipo “**Excel.Workbook**” contenido en la variable **objWorkbook**. Esta función nos devuelve la hoja del libro, como un objeto “**Excel.Sheet**”, indicada como parámetro. La enumeración de las hojas comienza en 1. Esta función es vital para poder acceder al contenido de cada una de las hojas del libro Excel.
- **Cells(int, int)**: Esta función es uno de los métodos que nos ofrece el objeto de tipo “**Excel.Sheet**”. Esta función recibe dos valores enteros, que representan la fila y la columna de la celda que vamos a devolver. Mediante esta función podemos ir accediendo a las celdas de la hoja Excel y modificar los valores de esta.
- **CInt (String)**: Esta función nos permite transformar un String a un valor entero. Esta función es esencial para la conversión de los

argumentos que representan un valor entero pero que son recibidos en formato texto.

- **Split (String, String):** Esta función nos permite separar una cadena en formato String, mediante un delimitador o patrón de separación. El primer argumento es la cadena completa y el segundo el patrón separador. Finalmente, esta función devuelve un array de tipo String, con las diferentes cadenas generadas de la separación.
- **UBound (array []):** Esta función devuelve el índice mayor o índice de la última posición accesible del array.

## 8.4.2. VBScript Algoritmos

Dentro del VBScript, podemos destacar principalmente dos algoritmos:

- Algoritmo que rellena las hojas con la información referente a los certificados a generar.
- Algoritmo que rellena el formato visual **HTML** de cada una de las credenciales.

El primer algoritmo, va recorriendo de la información referente a cada una de las credenciales, contenidas en la variable “**credenciales**”. La información de cada una de las credenciales viene escrita en líneas, delimitadas por el carácter “+”. Mediante el separador anterior, podemos ir recorriendo línea a línea la información de la credencial. La información de cada una de esas líneas corresponde a los datos de una fila perteneciente a una hoja específica.

Cada una de las líneas anteriores, podemos dividir las mediante el separador “;”, obteniendo un array de dos posiciones:

- Posición [0]: Contiene el número de la hoja a la que pertenece la información de esa línea.
- Posición [1]: Contiene la información a escribir en la fila. Los diferentes campos contenidos en esta línea están separados mediante el carácter “&”.

Mediante el número de la hoja, accedemos a la mismas y a las columnas de las celdas. Posteriormente, vamos recorriendo cada uno de los campos y añadiéndolos a las celdas de manera secuencial. A continuación, podemos observar el algoritmo 8.1, donde se nos muestra el algoritmo implementado.



' Vamos recorriendo los datos de las diferentes credenciales y rellenado el nuevo fichero Excel.

```

FOR EACH credencial IN credenciales
  lineas = Split(credencial, "+")
  FOR EACH linea IN lineas
    line_content = Split(linea, ";")
    sheet_index = CInt(line_content(0))
    set objSheet = objWorkbook.Worksheets(sheet_index)
    cells_index = dic(sheet_index)
    data = Split(line_content(1), "&")
    FOR i = 0 TO UBound(cells_index) step 1
      objSheet.cells(row_index_init, cells_index(i)).Value = data(i)
    NEXT
  NEXT
  row_index_init = row_index_init + 1
NEXT

```

Algoritmo 8.1. Proceso que nos permite rellenar la plantilla Excel con la información de las credenciales.

El segundo algoritmo, es similar, pero solo para rellenar el formato **HTML** y la imagen de fondo de cada una de las credenciales. A continuación, podemos observar el algoritmo 8.2, donde se nos muestra el algoritmo implementado.

```

'Asignamos el formato visual en html de las credenciales (HTML e imagen de fondo)
set objSheet = objWorkbook.Worksheets(2)
For i= 0 To CInt(args(3))
  objSheet.cells(13+i, 72).Value = formatos_html(i)
  objSheet.cells(13+i, 71).Value = "https://github.com/ricorcher/imagenes/blob/main/fondo\_certificados.jpg?raw=true"
NEXT

```

Algoritmo 8.2. Proceso que nos permite rellenar la plantilla Excel con el formato HTML e imagen de fondo de cada una de las credenciales.

### 8.4.3. EDCIConverter

Dentro del componente **EDCIConverte**, una de las funcionalidades principales es el procesamiento de la información JSON enviada por el cliente. El método que implementa esta funcionalidad recibe el nombre de “**procesarCredenciales**”. A continuación, podemos observar el algoritmo 8.3, que nos muestra el algoritmo implementado.

```

* Método que procesa la información enviada en el archivo JSON, en un formato entendible para
* el VBScript. Esta función devuelve un array de tipo String con la información.
*
* El formato de la información referente a las credenciales es el siguiente:
*
*     número_de_hoja1;valorcelda1&valorcelda2&valorcelda3&
*     número_de_hoja2;valorcelda1&valorcelda2&valorcelda3&
*     .....]número_de_hoja1;valorcelda1&valorcelda2&valorcelda3&
*     número_de_hoja2;valorcelda1&valorcelda2&valorcelda3&
*     .....].....
*
* El caracter ']' delimita las diferentes credenciales.
*
* El formato de la información referente al formato visual de la credencial en HTML es el siguiente:
*
*     HTMLcredencial1&SEPARADOR&HTMLcredencial2&SEPARADOR&.....
*
*
* @param credenciales: listado de credenciales (Contiene la información referente a la credencial.
* @param fecha: Fecha de emisión del lote de credenciales.
* @param firma1: Objeto de tipo Firma : Información referente al primer firmante del certificado.
* @param firma2: Objeto de tipo Firma : Información referente al segundo firmante del certificado.
* @return String[] :
*     posicion 0 : contiene la información referente a las credenciales (nombre de la persona, titulo, etc.)
*     posicion 1 : contiene la información referente al formato visual de la credencial en HTML.
*/
public String[] procesarCredenciales(List<Certificado> credenciales, String fecha, Firma firma1, Firma firma2) {
    String data = new String();
    String html_format = new String();
    for (Certificado aux : credenciales) {
        aux.setDenominacion(transformarTitulo(aux.getDenominacion())); // Transformamos aquellos títulos superiores a 50 caracteres.
        data = data + "2;español&" + aux.getDenominacion() + "&" + fecha + "&Generic&" + aux.getNombre() + "&Universidad de Extremadura+" +
            "3;" + procesarNombre(aux.getNombre()) + "&" + aux.getEmail() + "&" + aux.getWallet() + "];";
        html_format = html_format + procesarHTMLCredencial(aux, firma1, firma2) + "&SEPARADOR&";
    }
    return new String[] {data, html_format};
}

```

Algoritmo 8.3. Método que procesa el archivo JSON y devuelve la información en un formato entendible para el VBScript.

Para poder llevar a cabo el método anterior, se han necesitado de una serie de métodos auxiliares.

La primera recibe el nombre de “**procesarNombre**”. Para cumplimentar la plantilla **Excel EDC**, necesitamos el nombre y los apellidos de la persona que va a recibir o ser poseedor del título. Sin embargo, la información que nos envían desde el servicio de información del **SOFD** contiene solo el nombre completo de la persona.

Esta función devuelve el nombre y los apellidos de la persona como información de diferentes celdas. A continuación, podemos observar el algoritmo 8.4, que nos muestra la función implementada.

```

public String procesarNombre(String nombrecompleto) {

    String nombre_dividido = new String();

    String[] partes = nombrecompleto.split(" "); // partimos el nombre completo en partes.
    // partes[partes.length-2] siempre va a ser el primer apellido.
    // partes[partes.length-1] siempre va a ser el segundo apellido.

    for(int i =0; i < (partes.length-2); i++) {
        nombre_dividido = nombre_dividido + partes[i] + " "; //Construimos el nombre
    }
    // indicamos el nombre y los apellidos como información de celdas diferentes.
    nombre_dividido+= "&" + partes[partes.length-2] + " " + partes[partes.length-1];
    return nombre_dividido;
}

```

Algoritmo 8.4. Función para procesar el nombre completo recibido en el fichero JSON como información de dos celdas diferentes.

La segunda recibe el nombre de “**procesarHTMLCredencial**”. Uno de los objetivos de este TFG, era conseguir que las credenciales tuvieran un formato personalizado. Para ello, necesitaremos modificar el formato estándar establecido en la constante **html\_template**.

Esta función va a devolver un nuevo String con los campos personalizados por la información referente a esa credencial. A continuación, podemos observar el algoritmo 8.5, que nos muestra la función implementada.

```

public String procesarHTMLCredencial(Certificado c, Firma f1, Firma f2) {
    return html_template.replace("${nombre_firma1}", f1.getNombre())
        .replace("${cargo_firma1}", f1.getCargo())
        .replace("${nombre_firma2}", f2.getNombre())
        .replace("${cargo_firma2}", f2.getCargo())
        .replace("${anio}", c.getAnio())
        .replace("${zona}", c.getZona())
        .replace("${dia_imparte}", c.getDias_imparte())
        .replace("${numero_horas}", c.getNumero_horas());
}

```

Algoritmo 8.5. Función que genera en formato String la plantilla personalizada HTML para las credenciales.

Finalmente, la tercera y última función necesitada para procesar las credenciales recibe el nombre de “**transformarTitulo**”. Dependiendo del formato de la credencial y de si la longitud del título supera los 50 caracteres (sin tener en cuenta espacios), se podían dar casos donde el nombre del título no se mostrará correctamente en una línea en la credencial digital generada.

Como el título se va a colocar sobre **HTML**, se implementó un algoritmo que fuera introduciendo saltos de líneas (etiqueta **<br>** en **HTML**) para aquellos títulos superiores a 50 caracteres. A continuación, podemos observar el algoritmo 8.6, que nos muestra la función implementada.

```
public String transformarTitulo(String titulo_original){
    String newtitle = "";
    int indice = 0;
    if (titulo_original.replace(" ", "").length() > 50) {
        String aux = "";
        String[] palabras = titulo_original.split(" ");
        while(indice < palabras.length ){
            if((aux.length()+palabras[indice].length()) <= 50){
                newtitle = newtitle + palabras[indice] + " ";
                aux = aux + palabras[indice] + " ";
                indice++;
            }else {
                aux = "";
                newtitle = newtitle + "<br>";
            }
        }
    }
    else newtitle = titulo_original;
    return newtitle;
}
```

Algoritmo 8.6. Función que transforma los títulos de las credenciales, superiores de 50 caracteres, para poder visualizarse correctamente en la credencial.