

Proyecto Fin de Grado
Grado en Ingeniería de las Tecnologías de
Telecomunicación

Aplicación web para la educación mediante gamificación sobre el proyecto AJDA (Aplicación de Juegos Didácticos en el Aula) con funcionalidades de gestión de ficheros de preguntas

Autor: Guillermo Mejías Climent

Tutor: Francisco Javier Muñoz Calle

Dpto. Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Proyecto Fin de Carrera
Grado en Ingeniería de las Tecnologías de Telecomunicación

**Aplicación web para la educación mediante
gamificación sobre el proyecto AJDA (Aplicación
de Juegos Didácticos en el Aula) con
funcionalidades de gestión de ficheros de preguntas**

Autor:

Guillermo Mejías Climent

Tutor:

Francisco Javier Muñoz Calle

Profesor Colaborador

Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Grado: Aplicación web para la educación mediante gamificación sobre el proyecto AJDA
(Aplicación de Juegos Didácticos en el Aula) con funcionalidades de gestión de ficheros de preguntas

Autor: Guillermo Mejías Climent

Tutor: Francisco Javier Muñoz Calle

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Sevilla, 2021

A mi familia

A mis amigos

Agradecimientos

La posibilidad de estar hoy en día haciendo este trabajo es gracias a muchas personas que han estado a lo largo de estos últimos años, por ello, les quiero dedicar un pequeño párrafo a algunos de ellos.

En primer lugar, quiero darle gracias a mi tutor, Javier Muñoz, por permitirme hacer este fantástico trabajo, darme las herramientas necesarias, estar siempre respondiendo a mis dudas y dedicándome el tiempo que he necesitado.

También quiero agradecer a su hermano Jesús y a su alumno interno José Manuel por ayudarme en aquellos aspectos que eran más técnicos.

En tercer lugar, quiero agradecer a mi familia el apoyo incondicional durante todos estos años que han sido tan complicados para mí, no podría estar donde estoy ahora de no ser por ellos.

También he de agradecer a mi pareja su apoyo y consejos para poder seguir adelante. Tu apoyo y perspectiva en muchas de las cosas que me suponen mayor dificultad me ha facilitado mucho la tarea.

Por último, quiero darles las gracias a mis compañeros del grado, especialmente a Antonio Ángel y Jesús. Hemos pasado por esto juntos y no han sido pocas cosas. Tengo la certeza de que no habría llegado hasta aquí sin vuestra inestimable ayuda a lo largo de estos años.

Guillermo Mejías Climent
Escuela Técnica Superior de Ingeniería
Sevilla, 2021

Resumen

La utilización de juegos didácticos en el aula se ha mostrado como una excelente herramienta docente, con claras repercusiones positivas cuando su planteamiento, enfoque y ejecución son adecuados.

Este proyecto es una ampliación de uno anterior, en el cual se han añadido nuevas funciones, así como mejorado las ya existentes. Pretende así mejorar una plataforma que permita hacer un uso sencillo de dichos juegos y fomentar su uso.

Se han añadido una variedad de funciones: la posibilidad de importar y exportar ficheros de preguntas desde los distintos formatos más comúnmente utilizados por los docentes hoy en día, la creación de una base de datos que permita un fácil acceso a todos los tipos de ficheros desde un único lugar, la posibilidad de visualizar de forma fácil los ficheros disponibles en dicha base de datos y, por último, la creación de nuevos ficheros de preguntas haciendo uso de otros anteriores.

Abstract

Didactical games are a tool that is starting to become more frequent among teachers, since it has clear advantages when its approach and planning are adequate.

This project is an extension of a previous one, with the purpose of adding new functions and improving the ones that already exist. The objective is, therefore, to improve a platform that has the intention of easing and promoting the usage of aforementioned games.

A variety of new functions have been added: the possibility to import and export files that contain questions from and to the most common formats that are used by teachers nowadays. The usage of a database that gives access to every type of file from anywhere, as well as being able to visualize those files. Finally, it is also possible to combine already existing files to make new ones.

Contenido

Agradecimientos	IX
Resumen	XI
Abstract	XIII
Índice	XV
Índice de Tablas	XIX
Índice de Figuras	XXI
1 Introducción	1
1.1 <i>Motivación y objetivos</i>	1
1.2 <i>Descripción del problema</i>	2
1.3 <i>Antecedentes</i>	2
1.3.1 <i>Aplicación de Juegos Didácticos en el Aula</i>	2
1.3.2 <i>Aplicación web para la creación y modificación de ficheros de juegos para la Aplicación de Juegos Didácticos en el Aula (AJDA)</i>	4
1.4 <i>Descripción de la solución</i>	5
1.4.1 <i>Introducción</i>	5
1.4.2 <i>Objetivos</i>	5
1.4.3 <i>Arquitectura</i>	6
1.5 <i>Composición de la memoria</i>	7
2 Metodología	9
3 Tecnologías Utilizadas	11
3.1 <i>Tecnologías ya explicadas</i>	11
3.1.1 <i>Uso de XML</i>	11
3.2 <i>Nuevas tecnologías</i>	11
3.2.1 <i>SQL (Structured Query Language)</i>	12
3.2.2 <i>PostgreSQL</i>	12
3.3 <i>XPath</i>	12
4 Herramientas Utilizadas	13
4.1 <i>VirtualBox</i>	13
4.2 <i>Eclipse</i>	14
4.3 <i>Apache Tomcat</i>	14
4.4 <i>Navegadores</i>	15
4.5 <i>Git</i>	15
4.6 <i>Atom</i>	16
5 Estructura y Desarrollo de la Aplicación Web	17
5.1 <i>Instalación y configuración inicial</i>	17

5.2	<i>Estructura de los directorios</i>	20
5.3	<i>Generador</i>	21
5.3.1	Funcionamiento del generador	21
5.3.2	Exportación de ficheros	23
5.3.3	Importación de ficheros	26
5.4	<i>Buscador</i>	28
5.4.1	Funcionamiento del buscador	28
5.4.2	La base de datos	29
5.4.3	Guardar en la base de datos	31
5.4.4	Obtener resultados de la base de datos	32
5.4.5	Funciones con los resultados obtenidos	33
6	Ficheros Soportados	36
6.1	<i>Paneles sí o no</i>	37
6.2	<i>Columnas para relacionar</i>	38
6.3	<i>Respuestas con campo de texto</i>	38
6.4	<i>Respuestas con opciones</i>	39
6.5	<i>Contar letras</i>	40
6.6	<i>Paneles de letras</i>	40
6.7	<i>Respuesta abierta</i>	41
6.8	<i>Respuestas con cifras</i>	41
7	Plan de Pruebas	42
7.1	<i>Introducción</i>	42
7.2	<i>Pruebas</i>	43
7.2.1	Pruebas unitarias	43
7.2.2	Pruebas de integración	51
8	Conclusiones	53
9	Líneas de Mejora	55
10	Referencias	57
Anexo A: Documentación		58
A. <i>Conversor</i>		58
A.1 CreateXML		58
A.1.2 Exportador		58
A.1.3 XMLtoCSV		59
A.1.4 XMLtoDOC		59
A.1.5 XMLtoGIFT		59
A.1.6 XMLtoPDF		59
A.2 <i>ConversorGIFT</i>		60
A.2.1 FichTipoAbiertaGift		60
A.2.2 FichTipoCampoTextoGift		60
A.2.3 FichTipoAbiertaGift		60
A.2.4 FichTipoCifrasGift		60
A.2.5 FichTipoOpcionesGift		61
A.2.6 FichTipoRelacionarGift		61
A.2.7 FichTipoVFGift		61
A.3 <i>Lectura</i>		61
A.3.1 Importador		61
A.3.2 ReadCSV		62
A.3.3 ReadGIFT		62
A.3.4 ReadTXT		63
A.3.5 ReadXML		63
A.4 <i>AdministrarPreguntas</i>		64
A.4.1 AdminPreguntas		64

<i>A.5 Búsqueda</i>	65
A.5.1 FiltroBuscador	65
A.5.2 ResultTableItem	66
<i>A.6 Comunes</i>	66
A.6.1 CommonFunc	66
A.6.2 DBConnection	66

ÍNDICE DE TABLAS

<i>Tabla 2.1: Gantt parte 1</i>	9
<i>Tabla 2.2: Gantt parte 2</i>	9
<i>Tabla 5.1: Equivalencia AJDA-GIFT</i>	25
<i>Tabla 5.2: Marcas GIFT</i>	26
<i>Tabla 5.3: Base de datos</i>	30
<i>Tabla 5.4: Correspondencia buscador-base de datos</i>	32
<i>Tabla 6.1: Tabla de tipos</i>	37
<i>Tabla 6.2: Tipos de respuestas con opciones</i>	39
<i>Tabla 6.3: Tipos de paneles de letras</i>	40
<i>Tabla 7.1: PU-07</i>	43
<i>Tabla 7.2: PU-08</i>	43
<i>Tabla 7.3: PU-09</i>	44
<i>Tabla 7.4: PU-10</i>	44
<i>Tabla 7.5: PU-11</i>	45
<i>Tabla 7.6: PU-12</i>	45
<i>Tabla 7.7: PU-13</i>	46
<i>Tabla 7.8: PU-14</i>	46
<i>Tabla 7.9: PU-15</i>	47
<i>Tabla 7.10: PU-15</i>	47
<i>Tabla 7.11: PU-16</i>	47
<i>Tabla 7.12: PU-17</i>	48
<i>Tabla 7.13: PU-18</i>	48
<i>Tabla 7.14: PU-19</i>	48
<i>Tabla 7.15: PU-20</i>	49
<i>Tabla 7.16: PU-21</i>	49
<i>Tabla 7.17: PU-22</i>	50
<i>Tabla 7.18: PU-23</i>	50
<i>Tabla 7.19: P1-03</i>	51
<i>Tabla 7.20: P1-04</i>	51
<i>Tabla 7.21: P1-05</i>	52

ÍNDICE DE FIGURAS

<i>Figura 1.1: Página principal AJDA</i>	3
<i>Figura 1.2: Ejemplo de conversor</i>	3
<i>Figura 1.3: Ejemplo de búsqueda</i>	4
<i>Figura 1.4: Página principal TFG de Ramos León</i>	5
<i>Figura 1.5: MVC en JSF</i>	7
<i>Figura 3.1: Ejemplo de consulta SQL</i>	12
<i>Figura 3.2: Ejemplo de consulta SQL con XPath</i>	12
<i>Figura 4.1: Logo VirtualBox</i>	13
<i>Figura 4.2: Logo Eclipse</i>	14
<i>Figura 4.3: Logo Apache Tomcat</i>	14
<i>Figura 4.4: Logo Google Chrome</i>	15
<i>Figura 4.5: Logo Mozilla Firefox</i>	15
<i>Figura 4.6: Logo Microsoft Edge</i>	15
<i>Figura 4.7: Logo git</i>	16
<i>Figura 4.8: Logo Atom</i>	16
<i>Figura 5.1: Cambio de rama en Eclipse</i>	18
<i>Figura 5.2: Iniciar el servidor</i>	19
<i>Figura 5.3: Estructura de los ficheros</i>	20
<i>Figura 5.4: Diagrama conversores</i>	21
<i>Figura 5.5: Generador visto en navegador</i>	22
<i>Figura 5.6: Pasos para generar un fichero</i>	23
<i>Figura 5.7: Exportación</i>	23
<i>Figura 5.8: Exportación</i>	24
<i>Figura 5.9: Importación</i>	26
<i>Figura 5.10: Importación</i>	27
<i>Figura 5.11: Buscador visto en el navegador</i>	28
<i>Figura 5.12: Búsqueda</i>	29
<i>Figura 5.13: Modelo E-R</i>	29
<i>Figura 5.14: JAR añadido</i>	31
<i>Figura 5.15: Ejemplo de entrada en la base de datos</i>	31
<i>Figura 5.16: Ejemplo de consulta</i>	33
<i>Figura 5.17: Ejemplo de uso erróneo</i>	34
<i>Figura 5.18: Ejemplo de uso</i>	34
<i>Figura 6.1: Paneles Sí o No</i>	38
<i>Figura 6.2: Columnas para relacionar</i>	38
<i>Figura 6.3: Respuestas con campo de texto</i>	39
<i>Figura 6.4: Respuestas con opciones</i>	39
<i>Figura 6.5: Contar letras</i>	40
<i>Figura 6.6: Paneles de letras</i>	40
<i>Figura 6.7: Respuesta abierta</i>	41
<i>Figura 6.8: Respuestas con cifras</i>	41

1 INTRODUCCIÓN

«Cualquier idiota puede usar un ordenador. De hecho, muchos lo hacen».

Ted Nelson

Es innegable que la tecnología es una parte fundamental de la docencia hoy en día. Permite no solo dinamizar y mejorar los métodos clásicos de enseñanza, sino también formas alternativas de hacer que los alumnos aprendan de una manera más cómoda y con mayor interés.

Este trabajo es parte un proyecto global que busca añadir juegos dinámicos a las aulas, consiguiendo que los alumnos trabajen diferentes competencias clave, que usen varios tipos de aprendizaje y, al volverse sujetos activos, adquieren un aprendizaje más significativo.

No obstante, el estudiante no es el único perfil que se beneficia de estos avances. El docente también puede encontrar beneficios en estos juegos, tanto recibiendo una mejor realimentación por parte del alumnado como disminuyendo su carga de trabajo. Este trabajo se centra en esto último: busca facilitar la interacción del profesor con dichos juegos.

1.1 Motivación y objetivos

La motivación principal para hacer este trabajo la posibilidad de aprender a usar nuevas herramientas como, por ejemplo, PrimeFaces, así como adquirir destreza en el uso de muchas de las distintas tecnologías aprendidas en diversas asignaturas del grado.

Otro motivo por el cual es interesante este proyecto es porque se trata de un proyecto real, que será usado más adelante, en lugar de ser un trabajo que sirva únicamente para demostrar mis capacidades. La posibilidad de ofrecer nuevas herramientas para mejorar la docencia es algo que siempre he considerado muy interesante, en parte por lo limitados que se ven cada día más los métodos tradicionales de enseñanza.

El objetivo de hacer específicamente este trabajo es doble. En lo personal, demostrar que todo lo aprendido a lo largo de estos años son herramientas que se utilizan conjuntamente, dado que muchos de los aspectos aquí tratados han sido vistos anteriormente en asignaturas que, en un principio, carecían de relación entre sí.

En cuanto al objetivo más técnico del trabajo, ha sido principalmente desarrollar funciones que son necesarias para el proyecto global pero que aún no existían, siendo estas las conversiones entre ficheros y la persistencia de datos. Estas nuevas funciones intentan proporcionar a los docentes una mayor flexibilidad y la posibilidad de trabajar en varias sesiones desde varios equipos de forma transparente.

1.2 Descripción del problema

En la aplicación de AJDA hay dos problemas que se han intentado solucionar en este proyecto: la flexibilidad de los formatos de archivos y la posibilidad de trabajar con varios de ellos simultáneamente para poder crear nuevos archivos.

El problema de la flexibilidad de formatos de archivo viene dado por las limitaciones que tenía la plataforma, la cual solo permitía trabajar con tres formatos: TXT, XML y PDF (este último solo para exportar).

El formato TXT es propenso a fallos a la hora de convertirlo a otro sistema más estructurado, que en nuestro caso es un sistema de preguntas y respuestas, por lo que se debe evitar todo lo posible.

El formato XML es la solución que se propuso en el proyecto anterior a este (dicho proyecto se detallará más en el siguiente apartado), pero tiene una limitación esencial: muchos de los docentes no están habituados a trabajar con este formato de archivo, por lo que les puede suponer un inconveniente.

En cuanto a la manipulación y creación de distintos archivos de preguntas, el problema reside en que no hay forma de almacenar estos ficheros en el servidor y deben estar en el cliente en todo momento. Esto supone un engorro para el profesor, que debe mantener en su equipo local todos los ficheros que necesite. Tampoco puede combinar las preguntas de dichos ficheros, ya que actualmente no existe dicha funcionalidad en la aplicación.

1.3 Antecedentes

El proyecto tiene dos antecedentes completamente diferentes: el proyecto de fin de grado de Carlos Ramos León, titulado *Aplicación web para la creación y modificación de ficheros de juegos para la Aplicación de Juegos Didácticos en el Aula (AJDA)* [1] y la aplicación AJDA [2].

1.3.1 Aplicación de Juegos Didácticos en el Aula

Aplicación de Juegos Didácticos en el Aula (AJDA de ahora en adelante) [2] es una plataforma cuyo objetivo es poner al servicio de la comunidad educativa una serie de recursos didácticos que faciliten la enseñanza. En dicha plataforma encontramos una serie de juegos educativos que pretende gamificar algunos de los aspectos de la enseñanza (figura 1.1).

Los diversos juegos que podemos encontrar permiten generar su contenido de diversas maneras, siendo una de ellas a partir de ficheros de texto. Para facilitar esta tarea, existe la opción de «Ficheros de preguntas», visible en el menú principal. Dicha opción contiene un submenú con varias opciones, que son las que trataremos en este proyecto.

La opción del generador de preguntas ya se trató en el proyecto de Ramos León, por lo que en el presente proyecto solo haremos referencia a ella en ocasiones puntuales.

Las opciones que van a mejorarse en este trabajo son la del conversor (figura 1.2), buscador y lista de ficheros.

Actualmente el conversor de ficheros permite solo dos opciones: convertir de TXT a GIFT/Aiken y viceversa. Para ello, simplemente se elige un fichero que tengamos del tipo TXT y, si se lee correctamente, se convertirá a GIFT. De la misma forma, podemos hacer la conversión en sentido opuesto, de GIFT a TXT. La limitación de esta página está en que solo existen dos formatos con los que trabajar y son propensos a error, por lo que se han buscado alternativas más fiables.

El buscador de ficheros, como se puede ver en el ejemplo (figura 1.3), consiste en una búsqueda en función de varios parámetros de nuestro interés. Los resultados se muestran debajo, en un listado con todos los ficheros que cumplen los requisitos de dicha búsqueda. Las limitaciones del buscador son que no permite buscar por el contenido de las preguntas, seleccionarlas ni mezclar diversos ficheros.

Selección un juego

Lista -- Selecciona Juego --

Buscador de juegos

Inicio Buscadores Juegos Ficheros de preguntas Documentación Participación

Está aquí: Inicio

EL GRAN MINUTO UNOS BATALLA NAVAL NUMERO OCULTO

Visto: 3480112

Guía rápida. Uso de juegos

+ Mostrar/Ocultar

Figura 1.1: Página principal AJDA

Abrir en una ventana nueva

AJDA Convertor de ficheros de preguntas

Juegos a Gift o Aiken GIFT a Juegos DESMARCAR

T1 a Gift T1 a Aiken Tipo 2 a Gift Tipo 3 a Gift Tipo 4 a Gift T5 a Gift T5 a Aiken T7 a Gift T7 a Aiken Tipo 9 a Gift T12 a Gift T12 a Aiken T13 a Gift T13 a Aiken T14 a Gift T15 a Gift T16 a Gift V-F a Gift

Paso a GIFT de archivos de tipo 1

Seleccionar archivo fichero_tipo1.bt

Entrada

```
'AUTOR/A: '
'TEMA: '
'Pregunta 1'
'Pregunta 1'
"
"
'Respuesta 1.1'
'Respuesta 1.2'
'Respuesta 1.3'
'Respuesta 1.4'
'A'
'Pregunta 2'
'Pregunta 2'
'Línea adicional 1'
'Línea adicional 2'
```

Resultado

```
Respuesta 1.1 {
~Respuesta 1.2
~Respuesta 1.3
~Respuesta 1.4
~A
}
Línea adicional 1 Línea adicional 2 Respuesta 2.1 {
~Respuesta 2.2
~Respuesta 2.3
~Respuesta 2.4
~B
}
```

Nombre del nuevo archivo

Nombre del archivo de salida (sin extensión)

Guardar GIFT

Figura 1.2: Ejemplo de conversor

Está aquí: Inicio > Ficheros > Buscador de ficheros

Buscador de ficheros

Etapa	Curso	Asignatura
-- Selecciona Etapa --	2°, 3°, 4°	Conocimiento medio
Tema del fichero	Nombre fichero	Autor del fichero
<input type="text"/>	<input type="text"/>	-- Selecciona Autor del fichero --
Idioma	Tipo ficheros de preguntas	J. fich. individuales
Español	Tipo 01 (297/ilim/3/4/A-D)	-- Selecciona J. fich. individuales --
Forma respuesta	Intervalo preg.	Num. preg. fich.
-- Selecciona Forma respuesta --	-- Selecciona Intervalo preg. --	Desde <input type="text"/> - a <input type="text"/>

Resultados encontrados (7)

Figura 1.3: Ejemplo de búsqueda

1.3.2 Aplicación web para la creación y modificación de ficheros de juegos para la Aplicación de Juegos Didácticos en el Aula (AJDA)

El trabajo de fin de grado que da título a esta sección (de ahora en adelante, el proyecto de Ramos León) es la base de la que parte el nuestro, por lo que contiene muchas similitudes en cuanto a metodología, arquitectura y tecnologías. En apartados posteriores se resumirán conceptos y explicaciones que ya estén detalladas en el proyecto de Ramos León, haciéndose referencia a él.

El funcionamiento básico del proyecto consiste en un generador de ficheros con distintas opciones, mostrándose en la imagen posterior un ejemplo del «Tipo 1», el cual forma parte del grupo «respuestas con 4 opciones». A modo de introducción, el tipo de fichero es, en esencia, un formato de preguntas. En la sección 6 del documento se pueden consultar los distintos tipos de ficheros que existen, así como su estructura y diferencias.

Podemos observar que la página solo permite exportar a PDF y TXT, como ya hemos mencionado anteriormente. Además, aunque no es visible, el botón de «importar» solo permite la importación ficheros TXT o XML.

Por otro lado, no existe la opción de almacenar de ninguna manera los ficheros generados. La única opción disponible es exportarlos a XML y que sea el profesor el que se encargue de guardarlos, teniendo que importarlos luego sin poder combinarlos de ninguna manera.

Figura 1.4: Página principal TFG de Ramos León

1.4 Descripción de la solución

En los distintos apartados que aparecen a continuación se muestra la solución propuesta, detallándose el proceso de desarrollo que se ha llevado a cabo desde el inicio de cada idea hasta la implementación final.

1.4.1 Introducción

Como ya se ha mencionado en la introducción, lo que se intenta conseguir con este proyecto es proporcionar a la ADJA nuevas capacidades que aún no posee, utilizando de base el proyecto de Ramos León. Para ello se han mejorado algunos de los aspectos que quedaban por pulir en el trabajo anterior y se han añadido otros completamente nuevos.

El trabajo se puede dividir en dos grandes bloques: el generador y el buscador. El generador es la parte que ya existía y está formada por una única página, mientras que el buscador ha sido creado como un nuevo componente.

1.4.2 Objetivos

Dado que este proyecto es extensión de uno anterior, todos los objetivos de dicho proyecto deben seguir siendo válidos y no haberse visto comprometidos, los cuales se pueden consultar en [1]. Adicionalmente, los nuevos objetivos son los siguientes:

- Dentro del generador de ficheros se deben añadir las siguientes funcionalidades:
 - Exportar el grupo de preguntas en uso a una base de datos alojada en el servidor.
 - Convertir y exportar cualquier fichero generado a los siguientes formatos:
 - XML
 - DOC
 - CSV
 - TXT
 - GIFT
 - PDF

- Importar cualquier fichero anteriormente generado que esté en cualquiera de los siguientes formatos:
 - XML
 - DOC
 - CSV
 - TXT
- Posibilidad de navegar al buscador de ficheros fácilmente.
- El buscador de ficheros debe ser una página independiente y contar con las siguientes funcionalidades:
 - Buscar un fichero incluido en la base de datos utilizando cualquier combinación posible de los siguientes parámetros:
 - Nombre del fichero
 - Tema del fichero
 - Autor del fichero
 - Etapa educativa
 - Asignatura a la que pertenece
 - Idioma
 - Tipo de fichero (ya sea un tipo genérico o uno individual)
 - Palabras claves que estén en alguna de las preguntas o respuestas
 - Número de preguntas
 - Posibilidad de seleccionar los resultados uno a uno para ver su contenido.
 - Posibilidad de seleccionar varios resultados simultáneamente, siempre que sean del mismo tipo de pregunta, para poder generar combinaciones de ellos.
 - Posibilidad de eliminar ficheros de la base de datos.
 - Posibilidad de navegar al generador de ficheros fácilmente.

1.4.3 Arquitectura

El proyecto está basado en un *framework* llamado JavaServer Faces (JSF) [3], que utiliza el patrón de diseño Modelo-Vista-Controlador (MVC) [4]. Los tres elementos que componen la arquitectura del proyecto son, por tanto, los siguientes:

- Modelo: contiene los datos y lógica de la aplicación. El modelo se define principalmente en las clases llamadas Managed Bean [5], que no es más que una clase de Java con una serie de convenciones. Estas convenciones son:
 - Debe tener un constructor sin argumentos
 - Debe tener propiedades privadas y accesibles mediante métodos *get/set*
 - Deber ser serializable

La base de datos que utilizemos también será parte del modelo, sin embargo, no forma parte de JSF.

- Vista: Es la interfaz de usuario. Se muestra a través de ficheros XHTML y será donde el usuario pueda introducir y visualizar los datos del modelo.
- Controlador: es el principal intermediario entre la vista y el modelo. Se encarga de gestionar las acciones del usuario en la vista y se asegura de que el modelo las reciba adecuadamente. En caso de JSF se encarga la interfaz Faces Servlet.

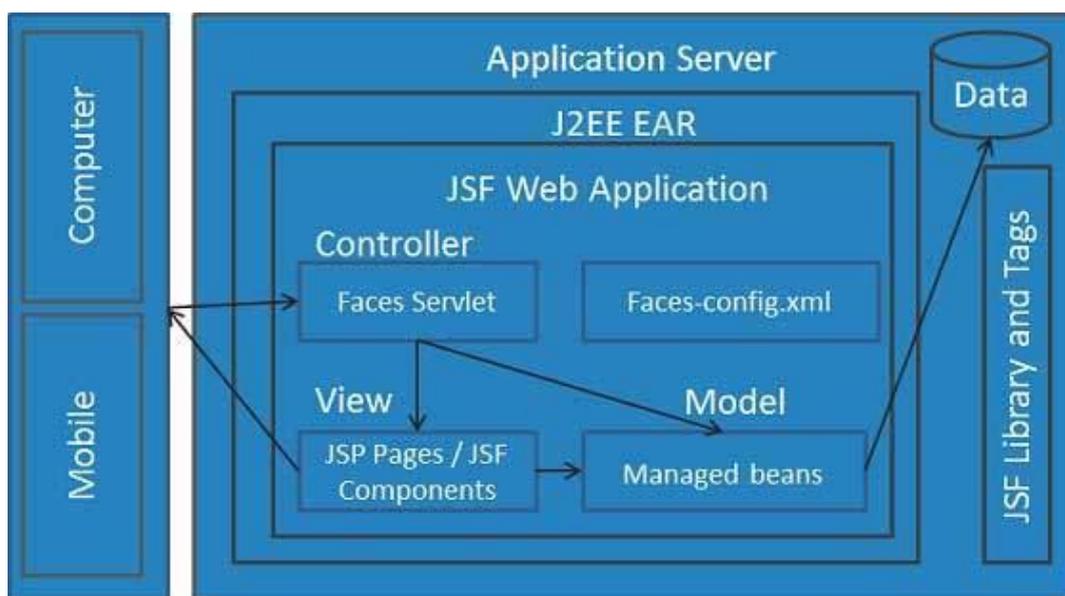


Figura 1.5: MVC en JSF

De manera muy simplificada, el flujo de la aplicación es el siguiente:

1. El usuario entra en la aplicación, se le muestra la vista.
2. El usuario introduce algunos datos, los cuales recibe el controlador.
3. El controlador se encarga simultáneamente de mostrar al usuario los datos introducidos a través de la vista y de enviárselos al modelo para que puedan ser tratados.
4. De los datos que recibe el modelo, los que sean necesarios se guardarán, modificarán o borrarán en la base de datos para ser utilizados en un futuro.

1.5 Composición de la memoria

El objetivo de este apartado es ofrecer una síntesis de todo este trabajo para que sea posible tener una visión global del mismo. Las secciones del presente documento son las siguientes:

1. Introducción: ofrece una idea general de qué trata el proyecto, así como de su estructura.
2. Metodología: presenta un resumen de las horas invertidas en cada aspecto del proyecto.
3. Tecnologías utilizadas: aquí se pueden ver las distintas tecnologías que se han utilizado para hacer posible el proyecto. Dado que un gran número de estas tecnologías son idénticas a las del proyecto de Ramos León [1], se mostrarán de forma más breve los elementos que sean considerados menos influyentes.
4. Herramientas utilizadas: este apartado posee una estructura muy similar al apartado 2, centrándose esta vez en las herramientas.
5. Estructura y Desarrollo de la Aplicación Web: se trata del apartado más técnico del documento. Aquí se puede encontrar la solución implementada para lograr cada objetivo, así como los ficheros, estructuras y procedimientos utilizados para hacerlo posible.
6. Plan de Pruebas: en esta parte del documento se pueden ver las distintas pruebas realizadas para comprobar que el funcionamiento del proyecto es el adecuado y que satisface las necesidades impuestas por los distintos objetivos.
7. Conclusiones: se muestran las conclusiones obtenidas tras realizar el proyecto.
8. Líneas de Mejora: se muestran las limitaciones que aún existen en el proyecto y se proponen soluciones para mitigarlas.
9. Referencias: ofrece información sobre los recursos a los que se ha accedido para realizar el proyecto.

2 METODOLOGÍA

«Primero resuelve el problema. Entonces, escribe el código».

John Johnson

En este apartado podemos ver un resumen de las horas que han sido necesarias para la realización del proyecto, así como en qué labores se ha empleado cada una de ellas. El objetivo de esta sección no es ofrecer un análisis exhaustivo del tiempo requerido para hacer cada tarea, sino más bien mostrar una idea generalizada de la ocupación que supone un proyecto de esta magnitud.

Tarea	S1	S2	S3	S4	S5	S6	S7	S8	S9
Familiarización con las bases del proyecto									
Documentación previa al entorno									
Familiarización con el entorno de trabajo									
Implementación de la exportación CSV									
Implementación de la exportación DOC									
Documentación de XML en la BD									
Implementación de la BD									

Tabla 2.1: Gantt parte 1

Tarea	S10	S11	S12	S13	S14	S15	S16	S17	S18
Implementación de la BD									
Implementación del buscador									
Implementación de la importación DOC y CSV									
Revisión de la exportación CSV									
Revisión de la exportación DOC									
Elaboración de la memoria									
Elaboración de la presentación									
Reestructuración de la memoria									

Tabla 2.2: Gantt parte 2

Ambos diagramas muestran una estimación de las horas necesarias para la realización del proyecto, separadas por semanas. Cada casilla corresponde a una semana, que son aproximadamente 25h de trabajo.

Los colores de las casillas son meramente informativos: el verde indica que es un trabajo de documentación y, por tanto, no hay un producto visible. Las casillas amarillas indican la creación de un elemento del proyecto y, por último, las rojas indican la modificación de algún componente ya creado.

3 TECNOLOGÍAS UTILIZADAS

«Cualquier tecnología suficientemente avanzada es equivalente a la magia.»

Arthur C. Clarke

En este apartado se detallan todas las tecnologías que son necesarias para hacer funcionar el proyecto completo a día de hoy. Entendemos por *tecnología* el conjunto de *frameworks*, técnicas y lenguajes tecnológicos que forman parte del proyecto.

3.1 Tecnologías ya explicadas

Dado que este proyecto es ampliación del de Ramos León, se puede consultar la definición de las tecnologías ya explicadas en su proyecto [1] y no se explicarán de nuevo aquí. Las tecnologías allí descritas son las siguientes:

- JavaServer Faces (JSF)
- Primefaces
- Managed Beans
- AJAX (Asynchronous JavaScript and XML)
- IText
- XML

3.1.1 Uso de XML

Si bien es cierto que la definición de XML ya ha sido expuesta por Ramos León, se han añadido nuevas funciones que utilizan XML de maneras distintas.

En concreto, cualquier fichero que se guarde en la base de datos lo hará en formato XML, independientemente del fichero de entrada. A la hora de convertir entre distintos formatos, el formato que hará de intermediario siempre será el XML, ya que es el que permite la flexibilidad necesaria para hacer el proceso sencillo.

3.2 Nuevas tecnologías

Además de las tecnologías ya utilizadas anteriormente, existen algunas nuevas que son necesarias para poder hacer uso de las nuevas funcionalidades, las cuales se explican en este apartado.

3.2.1 SQL (Structured Query Language)

SQL es el lenguaje más comúnmente utilizado para acceder y manipular bases de datos relacionales. Es un estándar sobre el que se implementan varias bases de datos, como la que se utiliza en el proyecto: PostgreSQL.

Dado que se trata de un modelo relacional, la información se almacena en relaciones o tablas, las cuales están compuestas por tuplas o filas y columnas o campos. Para acceder a dichos elementos se hace uso de consultas (*queries*).

```
SELECT * FROM preguntas;
```

Figura 3.1: Ejemplo de consulta SQL

3.2.2 PostgreSQL

PostgreSQL (o Postgres) [6] es una base de datos que implementa todas las funcionalidades de SQL, además de algunas propias. Se caracteriza por su extensibilidad, rendimiento, integridad y por ofrecer integración con tipos de datos como XML, que será de especial interés.

3.2.2.1 Uso en el proyecto

En el proyecto se utiliza Postgres porque es una base de datos que permite trabajar con XML, aunque posea algunas limitaciones. En este caso, su uso es para almacenar ficheros de preguntas, junto con metadatos que son relevantes a la hora de hacer las búsquedas.

Durante las fases iniciales del desarrollo se consideró también hacer uso de MongoDB, dado que las capacidades de Postgres respecto a XML no eran conocidas en un principio. Sin embargo, esta idea se abandonó una vez comprobado que Postgres es suficientemente flexible para el uso que hay que darle a la base de datos.

3.3 XPath

XPath es un lenguaje de consulta que se utiliza para obtener información sobre nodos en un documento XML [8]. Es uno de los elementos principales del estándar XSLT, el cual se utiliza para la transformación de elementos XML, aunque en el proyecto no será necesario hacer ninguna transformación [9].

3.3.1.1 Uso en el proyecto

En este proyecto se utiliza XPath dentro de consultas PostgreSQL, ya que incluye la función `xpath(xpath, xml [, nsarray])`, la cual permite obtener resultados de campos que sean de tipo xml. Su uso será para devolver el número de nodos de cierto tipo y para obtener el texto de un nodo en campos xml.

El ejemplo a continuación muestra cómo se obtiene el texto asociado al campo «autor» de una columna XML llamada «datos» en una relación llamada «preguntas»:

```
SELECT (xpath('autor/text()', datos))[1] FROM preguntas;
```

Figura 3.2: Ejemplo de consulta SQL con XPath

4 HERRAMIENTAS UTILIZADAS

*«El software es como el sexo: mejor si es libre y gratis»
Linus Torvalds*

Las distintas herramientas que se han utilizado se pueden ver a lo largo de esta sección. A pesar de no ser visibles en el producto final, son un recurso indispensable de cara al desarrollo, ya que sin ellas sería imposible escribir, implementar, probar y depurar los distintos elementos que componen el proyecto. Por este motivo, saber hacer un uso adecuado de ellas resulta tan importante.

4.1 VirtualBox

VirtualBox es un software de virtualización *open source* multiplataforma, que se utiliza para virtualización x86 capaz de virtualizar diversos sistemas operativos [9].

En este proyecto se ha utilizado para emular un sistema Ubuntu 18.04 que contenía preinstaladas todas las herramientas necesarias para empezar a trabajar. Esto ha permitido ahorrar mucho, ya que no se necesita instalar cada una de las herramientas por separado ni mantener un control de versiones sobre todas ellas.



Figura 4.1: Logo VirtualBox

4.2 Eclipse

Eclipse es un entorno de desarrollo integrado (IDE) [10] cuyo público objetivo son desarrolladores de aplicaciones Java, no obstante, ofrece funcionalidad para trabajar con otros lenguajes de programación, como pueden ser C, C++, JavaScript o Python.

Las funcionalidades que ofrece Eclipse son muy diversas, de las cuales las más utilizadas para el proyecto han sido las siguientes:

- Gestor de ficheros
- Gestor de paquetes de Java
- Posibilidad de gestionar el servidor de la aplicación
- Depurador
- Integración con Git
- Soporte para extensiones, de las cuales son destacables:
 - Extensiones de personalización de interfaz
 - Extensiones con funcionalidades de búsqueda en el propio proyecto
- Soporte multiplataforma



Figura 4.2: Logo Eclipse

4.3 Apache Tomcat

Apache Tomcat [11] es el servidor web utilizado para el proyecto. Es una implementación *open source* de la plataforma Jakarta EE, creada por la Fundación Apache.

Se utiliza como contenedor de *servlets* y ha sido elegido por ser uno de los servidores más extendidos dadas a su robustez y eficiencia. Sin embargo, es posible que en un futuro este proyecto se migre a un servidor WildFly.



Figura 4.3: Logo Apache Tomcat

4.4 Navegadores

En este proyecto se han utilizado diversos navegadores para, en esencia, probar las funcionalidades que se han ido implementando. Dado que el usuario final puede acceder al proyecto desde cualquier navegador, se ha optado por hacer uso de varios de los que existen hoy en día en el mercado, para asegurar que la funcionalidad es la esperada en cualquier circunstancia.

Para las herramientas de depuración se ha utilizado principalmente el navegador Google Chrome y, en casos puntuales, Mozilla Firefox por familiaridad, aunque el resto de navegadores ofrecen unas funcionalidades de depuración muy similares, siendo todos ellos opciones igualmente válidas.



Figura 4.4: Logo Google Chrome



Figura 4.5: Logo Mozilla Firefox



Figura 4.6: Logo Microsoft Edge

4.5 Git

Git es el software de control de versiones más extendido hoy en día. Su objetivo es hacer un seguimiento de cualquier tipo de proyecto haciendo uso de versiones y ramas. Sirve tanto para mantener lo que se conoce comúnmente como copia de seguridad y facilita que distintos miembros de un grupo trabajen en un mismo proyecto simultáneamente.

Para trabajar con Git se han utilizado las herramientas que proporciona Eclipse. En ellas se han preparado los ficheros modificados, se ha hecho *commit* y se han subido al repositorio remoto que contiene el resto de componentes del proyecto. Todo esto se ha hecho en una rama designada a este TFG, para no alterar las modificaciones que otros usuarios puedan hacer en el mismo intervalo temporal.



Figura 4.7: Logo git

4.6 Atom

Atom es un editor de texto orientado a código, *open source*, multiplataforma y muy sencillo que permite trabajar fácilmente con cualquier tipo de fichero que sea interpretable como texto.

Se ha utilizado como editor adicional a Eclipse para poder analizar en detalle el comportamiento de los distintos exportadores desarrollados, ya que es capaz de leer y mostrar de forma cómoda todos los ficheros con los que se trabajan en el proyecto.



Figura 4.8: Logo Atom

5 ESTRUCTURA Y DESARROLLO DE LA APLICACIÓN WEB

«Cualquier código tuyo que no hayas mirado en los últimos seis meses o más es como si lo hubiese escrito otro»

Ley de Eagleson

En este apartado se describirán los distintos ficheros que componen las distintas funcionalidades del proyecto. Dado que parte de la estructura de ficheros proviene del proyecto anterior, aquellos ficheros que no hayan sido alterados ni modificados de ninguna manera no serán mencionados. También se explicará cómo realizar una configuración inicial del proyecto para ponerlo en funcionamiento por primera vez.

5.1 Instalación y configuración inicial

Para acceder al proyecto por primera vez es necesario, en primer lugar, tener disponible la máquina virtual del entorno de desarrollo. En caso de no tenerla, es posible configurarla siguiendo la guía disponible en [12].

Una vez dentro de dicha máquina, se abre la terminal y se accede al directorio «gamificación» con el comando `cd gamificación`.

En dicho directorio cambiaremos a la rama de este proyecto, es decir, `tfg-guillermo_mejias_climent`. Para ello utilizaremos el comando `git checkout tfg-guillermo_mejias_climent`. Si esta rama no existía en la máquina local, se obtendrá la última versión de ella. Si ya existía será necesario ejecutar `git pull` para actualizarla.

Una vez tengamos el proyecto en la rama adecuada accederemos a Eclipse. En él abriremos el proyecto con *File* → *Open Projects from File System...* → *Directory...* y seleccionando el directorio «gamificación».

Si no hemos realizado el primer paso y queremos cambiar a la rama adecuada dentro de Eclipse, también es posible. Para ello primero es necesario cambiar a la perspectiva de Git con *Window* → *Perspective* → *Open Perspective* → *Other* → *Git*. Aparecerá a la izquierda una pestaña con el repositorio, en la cual podemos cambiar de rama como se muestra a continuación:

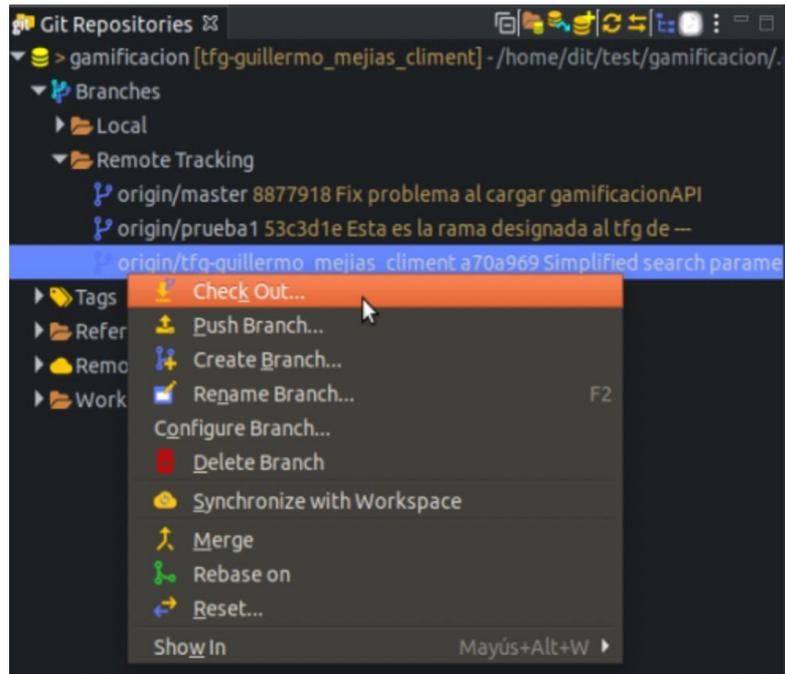


Figura 5.1: Cambio de rama en Eclipse

El último paso que hará falta será encender el servidor. Para ello ejecutaremos los pasos que se muestran a continuación:

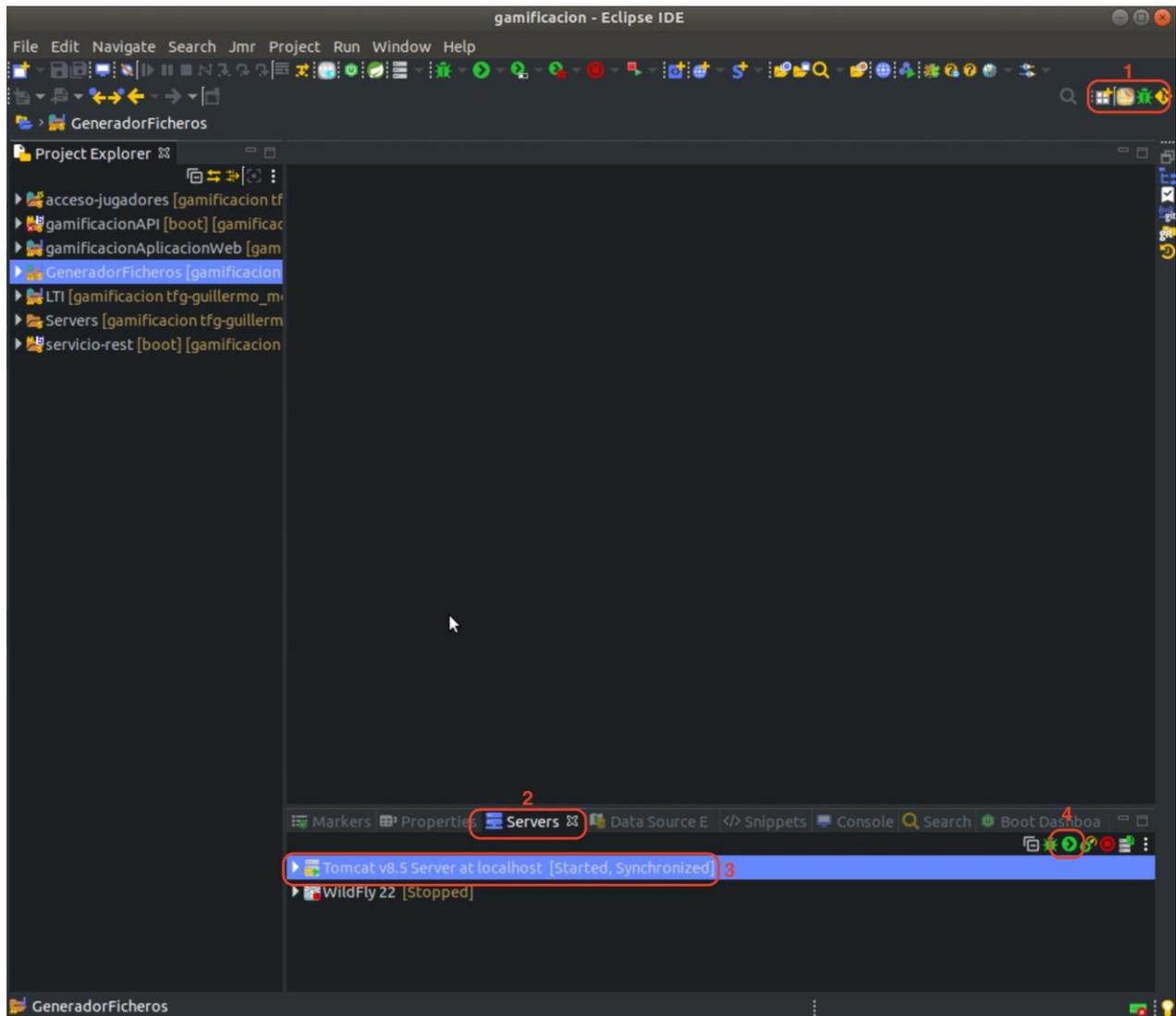


Figura 5.2: Iniciar el servidor

- 1) Cambiamos de vista a la de desarrollo (se encuentra resaltado en la imagen).
- 2) Seleccionamos la pestaña de servidores.
- 3) Seleccionamos el servidor Tomcat, que es el servidor en el que se aloja actualmente en el proyecto.
- 4) Pulsamos el botón de iniciar.

Una vez todo esté listo podemos acceder al proyecto desde cualquier navegador. Si utilizamos el navegador de la máquina virtual, con la URL `localhost:8081/GeneradorFicheros` podemos visualizar el generador, el cual se explicará más adelante.

En caso de querer acceder desde fuera de la máquina virtual, podemos hacerlo accediendo a dirección en la que dicha máquina virtual se encuentre. Un ejemplo para hacer esto funcionar en la red local sería el siguiente:

- 1) En la configuración de la máquina virtual, cambiamos el adaptador de red al tipo «Adaptador puente».
- 2) Seleccionamos el nombre del adaptador con el que el equipo host esté conectado a la red.
- 3) Dentro de la máquina virtual, ejecutamos el comando `ip address` o equivalente para ver la dirección del equipo.
- 4) Accedemos desde cualquier máquina que tenga acceso a esa dirección con la url `x.x.x.x:8081/GeneradorFicheros`

Por último, si en algún momento se desea alterar la URL de acceso se puede hacer modificando la siguiente línea, disponible en el fichero `server.xml` del servidor:

```
<Context docBase="GeneradorFicheros path=/GeneradorFicheros reloadable="true"
source="org.eclipse.jst.Jee.server:GeneradorFicheros"/>
```

5.2 Estructura de los directorios

La estructura de los paquetes java necesarios es la siguiente:

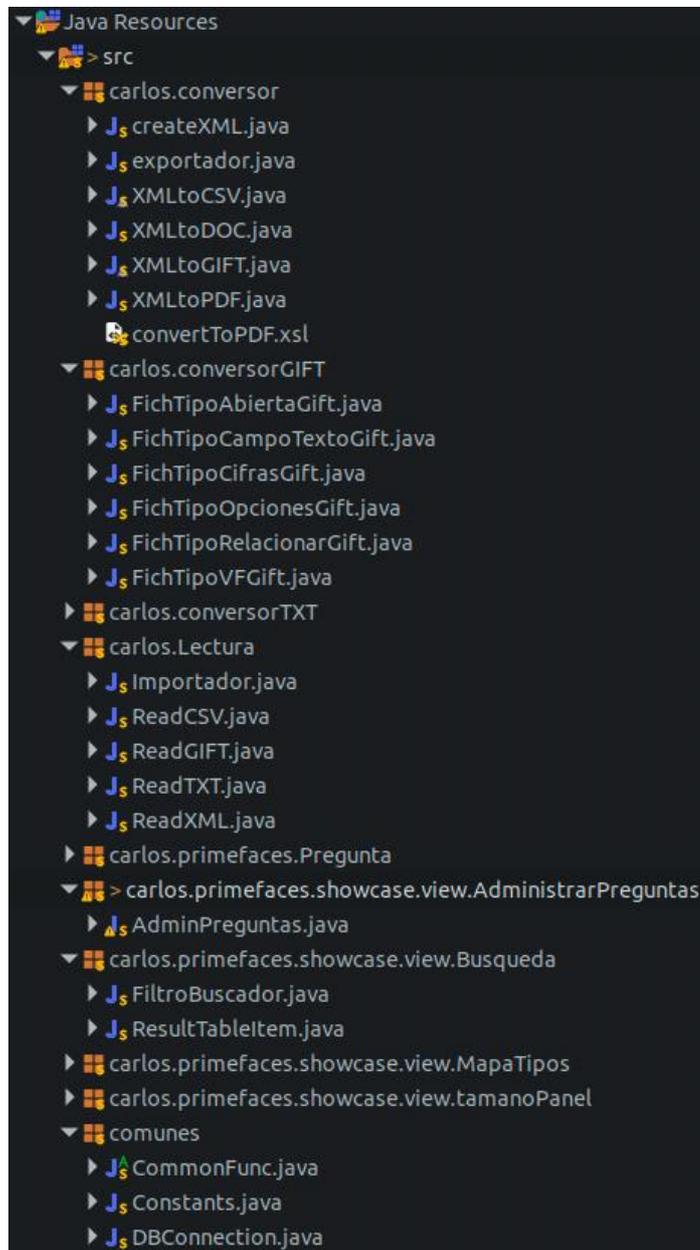


Figura 5.3: Estructura de los ficheros

Estos paquetes, junto con sus clases, contienen toda funcionalidad necesaria para hacer funcionar la aplicación. A lo largo de esta sección se explicará su uso, no obstante, se puede obtener una información más detallada de cada función en el anexo.

Además de estos ficheros, se utilizan varios ficheros XHTML, entre los que podemos destacar dos: `generador.xhtml` y `buscador.xhtml`. El resto de ficheros XHTML son módulos que se utilizan para visualizar las preguntas, que se explicarán en su propia sección más adelante.

También hay diversos ficheros de configuración y librerías que se utilizan y serán mencionados cuando sea necesario.

5.3 Generador

El generador es el componente utilizado para crear y leer ficheros de cada tipo. En él se encuentran las opciones para generar ficheros XML, convertirlos a otras extensiones, leer ficheros de cualquier tipo y guardar las preguntas en la base de datos.

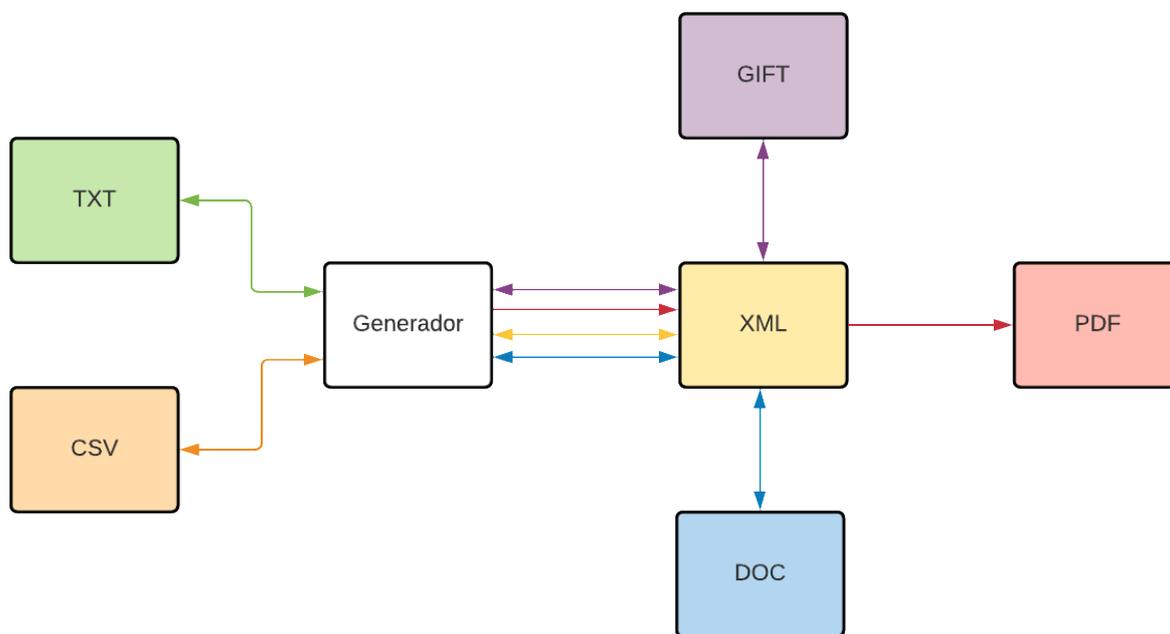


Figura 5.4: Diagrama conversores

La figura superior muestra un esquema del funcionamiento de los distintos conversores que existen en la plataforma, los cuales se detallarán más adelante. Cada tipo de fichero tiene un código de color, que permite ver los pasos que sigue el proyecto para hacer la conversión.

En el caso de TXT, CSV y XML (marcados en verde, naranja y amarillo, respectivamente) vemos que, para importar y exportar, la generación se hace directamente desde el contenido del generador a un archivo de su respectivo formato.

En el caso de GIFT, PDF y DOC (marcados en morado, rojo y azul, respectivamente) lo que se hace internamente (a través de clases que se explicarán en sus respectivos apartados), es un paso intermedio: convertir a XML. Para exportar, se obtienen los datos del generador, se convierten a XML y es ese propio fichero XML el que se transforma en el formato adecuado. Para importar se realiza el proceso contrario: se lee el fichero en cuestión, se convierte a XML y después se extraen los datos de dicho XML para mostrarlos en el generador.

5.3.1 Funcionamiento del generador

El generador ha sido actualizado para incluir las nuevas funcionalidades:

Generador de ficheros de Preguntas

Opciones

Nuevo fichero

Respuestas con opciones

Cargar

Seleccionar archivo | Ningú...ionado

Importar

Exportar

Seleccionar archivo | Ningú...ionado

PDF | TXT | DOC | CSV | GIFT

Enlaces

Buscador de juegos

Tipo 1 Al cambiar el tipo de juego, se borrarán las preguntas de otro tipo.

Idioma: Etapa: Asignatura:

Elegir Elegir

Curso: Nivel: Observaciones:

Borrar Preguntas

Autor:

Título:

Aplicación de juegos didácticos en el aula

Figura 5.5: Generador visto en navegador

En primer lugar, la pestaña de exportar ahora posee nuevos botones para exportar a nuevos formatos. Además, se ha añadido una nueva sección que contiene enlaces a otra parte del proyecto, en este caso, al buscador.

En segundo lugar, se han añadido nuevos campos opcionales que se pueden rellenar independientemente del tipo de pregunta. Estos campos se guardarán en la base de datos (siempre que sean rellenados) para poder utilizarlos en las búsquedas.

Por último, se ha añadido un nuevo campo en la parte inferior del formulario de preguntas que permite guardar preguntas en la base de datos. Este botón está acompañado de un campo de texto, que será el nombre del fichero que se guardará.

A continuación, se muestra una captura con los pasos a seguir para generar un nuevo fichero de preguntas, con los pasos obligatorios en rojo y los opcionales en azul:

Figura 5.6: Pasos para generar un fichero

- 1) Se selecciona el formato de juego.
- 2) Se selecciona el tipo de juego. Este menú contendrá unas opciones distintas en función del formato seleccionado.
- 3) En caso de querer guardar el fichero en la base de datos, se pueden rellenar estos campos adicionales para poder buscarlo posteriormente.
- 4) Se rellena el autor y título del juego.
- 5) Se rellenan las preguntas. El formato de pregunta variará en función del tipo de juego seleccionado, como se ha visto en la sección 5.
- 6) Si el tipo de juego lo permite, se pueden añadir nuevas preguntas al fichero o duplicar las ya existentes.
- 7) En caso de querer guardar el fichero en el equipo local, se pulsará el botón «Crear fichero XML», el cual iniciará una descarga. Si se desea guardar en la base de datos, se rellenará el campo con un nombre de fichero y se pulsará el botón «Guardar preguntas».

5.3.2 Exportación de ficheros

En el generador, para exportar un fichero hay que seguir los siguientes pasos:

Figura 5.7: Exportación

- 1) En primer lugar, se selecciona un archivo XML que esté en el equipo local. Esto puede hacerse pulsando el botón o arrastrando el archivo encima de dicho botón.
- 2) En segundo lugar, se selecciona el formato de salida y comenzará la descarga.

A continuación, se muestran los ficheros principales que componen la exportación de ficheros:



Figura 5.8: Exportación

Aquí se ha añadido al fichero exportador.java, el cual ya existía, las funcionalidades asociadas a los botones de exportar en los nuevos formatos que se vieron en el generador: exportar a DOC, GIFT y CSV. También se ha añadido un nuevo paquete con las exportaciones GIFT.

Cuando se pulsa uno de los botones del exportador se llama a la clase exportador, que crea una nueva instancia de la clase correspondiente (el exportador del tipo) y es esta clase la que se encarga de generar el fichero. A continuación, se indica qué hace cada tipo de exportación.

5.3.2.1 Exportar a CSV

Las siglas CSV significan *comma-separated values* [13], o valores separados por comas en español. Esto nos da una idea de cómo funciona este formato. Simplemente se trata igual que un fichero de texto plano, pero cada coma implica una separación de celda.

Lo que se ha hecho es una equivalencia. Cada línea en el formato TXT es una celda en el formato CSV, eliminando las comillas y manteniendo el resto del formato. Esto hace que la estructura sea muy similar a un TXT, pero con una mayor legibilidad.

5.3.2.2 Exportar a GIFT

El formato GIFT o *General Import Format Template* [14] es un lenguaje de marcas utilizado por plataformas educativas para generar cuestionarios.

Lo que se ha hecho es buscar la equivalencia entre el formato GIFT y las preguntas del proyecto AJDA, las cuales se pueden ver en la siguiente tabla:

Proyecto AJDA	Formato GIFT	Clase encargada
Respuestas con opciones	Opción múltiple	FichTipoOpcionesGift
Respuestas con cifras	Preguntas numéricas	FichTipoCifrasGift
Respuestas campo de texto	Respuesta corta	FichTipoCampoTextoGift
Paneles sí o no	Falso-verdadero	FichTipoVFGift
Relacionar	Relacionar columnas (emparejamiento)	FichTipoRelacionarGift
Respuesta abierta	Ensayo	FichTipoAbiertaGift

Tabla 5.1: Equivalencia AJDA-GIFT

Con estas equivalencias hechas lo que hace la clase XMLtoGIFT es identificar qué tipo de archivo se quiere exportar y enviárselo a la clase encargada del paquete de conversorGIFT (ver figura 5.1), la cual hace la conversión con las marcas necesarias. Dentro de estas clases no se distingue por tipo, dado que las variaciones que caracterizan a los ficheros dentro de un formato son de cantidad de elementos y no de su tipo.

Debido a que cada tipo de pregunta es distinto, las marcas necesarias también lo son. Sin embargo, hay un elemento común a todas las preguntas: el autor, título y tipo de pregunta, que deben formar parte del fichero para mantener compatibilidad con el resto de tipos, aunque no forman parte de las preguntas. Se han añadido, por tanto, como comentario; el autor y título al principio fichero, mientras que el tipo, al final.

Existen dos formatos que carecen de conversión a GIFT: paneles de letras y contar letras. Paneles de letras no tiene porque funciona a base de pistas, y el concepto de pista no existe en el formato GIFT. De igual manera, contar letras no tiene conversor porque no existen las marcas ni la estructura necesaria para hacer un cuestionario de adivinar letras.

A continuación, se explican brevemente las marcas que se han utilizado con un ejemplo de pregunta para cada formato, aunque una explicación más detallada se puede encontrar en [14].

Marca	Uso	Formatos utilizados
//	Comentario. Lo que venga a continuación no lo leerá el procesador de texto.	Todos
::texto::	Título de la pregunta. En caso de no existir se utilizará como título el texto sin marcas.	<ul style="list-style-type: none"> • Paneles sí o no • Respuesta abierta
{	El contenido entre corchetes serán las respuestas a la pregunta.	Todos
=	Añadido como prefijo a una respuesta, indica respuesta correcta.	<ul style="list-style-type: none"> • Respuestas con opciones • Respuestas campo de texto • Relacionar
~	Añadido como prefijo a una respuesta, indica respuesta incorrecta.	<ul style="list-style-type: none"> • Respuestas con opciones
T	Dentro de unos corchetes, indican que la respuesta a esa pregunta es «sí».	<ul style="list-style-type: none"> • Paneles sí o no

Marca	Uso	Formatos utilizados
FALSE	Dentro de unos corchetes, indican que la respuesta a esa pregunta es «no».	Paneles sí o no
#	Como prefijo a un número dentro de unos corchetes, indica que es la respuesta numérica correcta.	Respuestas con cifras
->	Indican, como prefijo, el primer campo de una pareja de relacionar (debe estar seguido por una respuesta con =).	Relacionar

Tabla 5.2: Marcas GIFT

Como aclaración final, en la mayoría de los casos la conversión es directa, pero en las respuestas con campo de texto se ha tenido que adaptar el campo «pista», ya que GIFT, como ya se ha indicado, no tiene marca para definirlos. Lo que se ha hecho ha sido añadirlo como parte del enunciado. El enunciado para este tipo de ficheros es, por tanto: «texto_enunciado» + Pista: + «texto_pista».

5.3.2.3 Exportar a DOC

El formato DOC es un formato de texto que tiene una diferencia fundamental respecto a los mencionados anteriormente: no posee ningún tipo de marca. Esto es una limitación, ya que, para saber qué es cada elemento del fichero, necesitamos algún tipo de identificador de los elementos.

Como solución, se ha propuesto utilizar el mismo lenguaje de marcas que GIFT, ya que son marcas sencillas de ver y utilizar, no intrusivas. La clase XMLtoDOC lo que hace, por tanto, es generar un fichero de tipo DOC y utilizar las mismas clases que fueron utilizadas en GIFT para generar la estructura del fichero.

5.3.3 Importación de ficheros

Para importar un fichero hay que seguir una serie de pasos, que se muestran a continuación:

Figura 5.9: Importación

- 1) En primer lugar, seleccionamos el formato de fichero que vayamos a importar.
- 2) En segundo lugar, seleccionamos el tipo.
- 3) A continuación, seleccionamos el archivo en local. Esto puede hacerse pulsando el botón o arrastrando el archivo encima de dicho botón.
- 4) Por último, pulsamos el botón de importar.

Si el tipo de fichero importado se corresponde con el seleccionado, las preguntas aparecerán en la página, en caso contrario aparecerá un mensaje de error.

Los ficheros que hacen posible la importación son los siguientes:

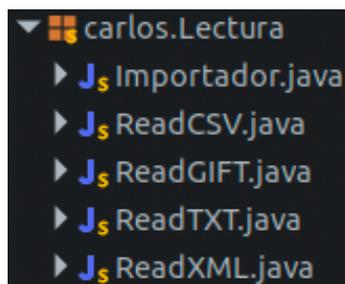


Figura 5.10: Importación

En este caso, vemos que hay un importador base, cuya clase es llamada al pulsar el botón de importar. Esta clase leerá el tipo de fichero y llamará a la clase correspondiente al formato del que sea el fichero. Si se intenta importar un fichero de un tipo incompatible se mostrará un error.

5.3.3.1 Importar a CSV

Como ya se ha visto en la exportación, el fichero CSV es muy similar al TXT. Esto implica que la clase ReadCSV es muy similar a ReadTXT. La diferencia reside en que se asume que cada línea de texto es una celda en lugar del contenido entre comillas simples, ya que no están presentes. Esta clase también utiliza las clases MenuMapTipos y MenuMapTiposOpciones, que se pueden encontrar definidas y explicadas en el proyecto de Ramos León.

5.3.3.2 Importar a GIFT

La importación a GIFT solo es compatible con las preguntas de tipo «respuestas con opciones», por el momento, debido a restricciones de tiempo.

Lo que hace la clase ReadGIFT es lo siguiente:

- Crea un XML vacío y copia el contenido del fichero GIFT en una cadena de texto.
- Lee la última línea del fichero, la cual será un comentario con el tipo de preguntas que contiene.
- Si el tipo es compatible, leerá también el título y autor. Posteriormente pondrá estos tres parámetros en el XML y los borrará de la cadena de texto
- Leerá cada pregunta y pondrá el enunciado en el XML. Borrará cada pregunta (enunciado y respuesta) de la cadena de texto.
- Leerá cada respuesta y las añadirá al XML, y guardará cuál de ellas es la correcta.
- Una vez que no quede contenido en la cadena de texto, se entiende que el fichero ha sido leído por completo y se enviará el XML creado a una instancia de la clase ReadXML, la cual se encargará de mostrar dicho XML al usuario.

5.3.3.3 Importar a DOC

Si observamos bien la figura 5.5, veremos que no existe una clase llamada ReadDOC. Esto se debe a que, como ya vimos a la hora de exportar, las marcas utilizadas para DOC son las mismas que para GIFT.

Lo que hace ReadGIFT es leer el contenido del fichero, independientemente de su extensión, e interpretarlo siguiendo las marcas de opción múltiple de GIFT. De esta forma, si el fichero es DOC se leerá exactamente igual que si es GIFT.

5.4 Buscador

Figura 5.11: Buscador visto en el navegador

El buscador es un elemento nuevo y se accede a él mediante el enlace en el generador.

La vista se divide en tres elementos principales: un menú de opciones, que permite volver al generador, un menú de búsqueda y uno de resultados.

El menú de búsqueda contiene los elementos necesarios para realizar las búsquedas, con campos desplegable en las opciones que permiten un rango de resultados y un campo de texto en las que permiten buscar cualquier contenido. También hay un campo de número de preguntas que permite filtrar en un rango.

El menú de resultados muestra lo obtenido al realizar una búsqueda. Se obtiene una lista de resultados junto a su tipo para que sea más sencillo componer ficheros de preguntas.

El botón de borrar elimina los resultados que hayan sido marcados de la búsqueda. El de usar ficheros lleva al usuario a la página del generador con el contenido de los ficheros marcados, pero solo si son del mismo tipo. El botón de eliminar de la base de datos elimina los ficheros tanto de la base de datos como de la búsqueda.

5.4.1 Funcionamiento del buscador

En este apartado se muestra un ejemplo de uso del buscador, paso a paso:

Nueva búsqueda

Etapa: Asignatura: Título del fichero: 1

Nombre del fichero: Autor del fichero: Idioma: Idioma

Tipo ficheros: Ficheros individuales: Curso: 0

Nivel: Observaciones: Palabra Clave:

Entre 0 y 100 preguntas

3 6 Debe cumplir todos los criterios 2

Ficheros encontrados

Nombre del fichero	Tipo
<input type="checkbox"/> jurado 4	TipoJurado
<input type="checkbox"/> Fichero1	Tipo3

5

Figura 5.12: Búsqueda

- 1) En primer lugar, se rellenan todos los campos de los que se desee hacer la búsqueda.
- 2) Opcionalmente, se puede elegir que deban cumplirse todos los campos.
- 3) En segundo lugar, se pulsa el botón «buscar».
- 4) De forma opcional, se puede seleccionar uno de los resultados para visualizar su contenido.
- 5) También es posible realizar otras funciones con los ficheros encontrados, que se explicarán en apartados posteriores.
- 6) Por último, si se desea hacer una nueva búsqueda desde cero, se pueden limpiar todos los campos con el botón «Limpiar todo».

Las búsquedas se pueden hacer de dos formas: cumpliendo todos los campos o solo alguno de ellos. Si se desea que para que aparezca un resultado sea necesario cumplir únicamente uno de los campos, la casilla «Debe cumplir todos los criterios» debe mantenerse sin marcar (este es el caso del ejemplo). Si, por el contrario, se desea que cada resultado satisfaga todos los criterios, dicha casilla deberá marcarse.

5.4.2 La base de datos

El núcleo del buscador de ficheros es una base de datos PostgreSQL llamada «genfich». Esta base de datos contiene una única tabla llamada preguntas. El modelo entidad-relación de dicha base de datos es el siguiente:

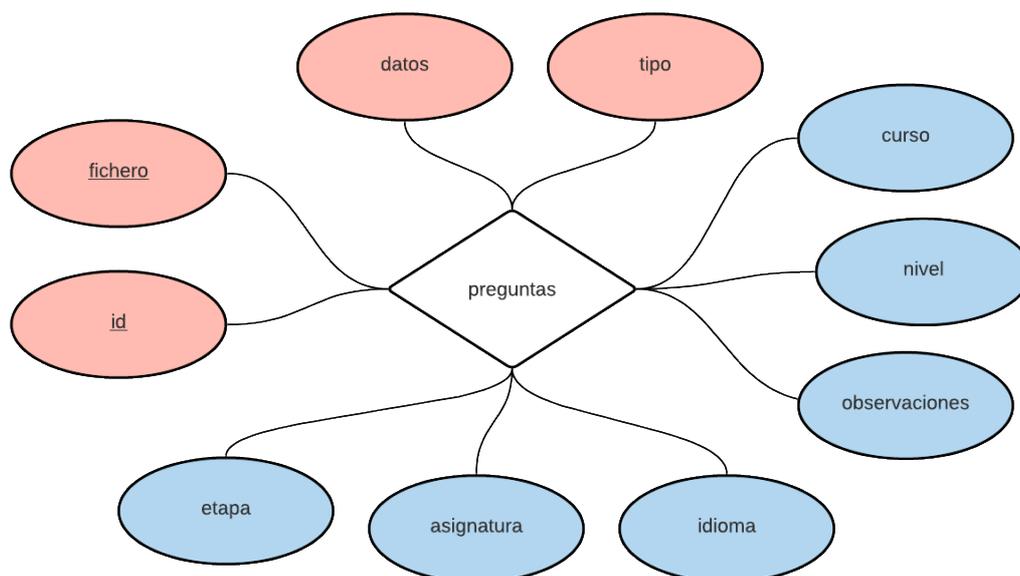


Figura 5.13: Modelo E-R

Por claridad se han diferenciado los atributos por colores: los rojos siempre contendrán algún dato, mientras que los azules son opcionales.

Se puede apreciar que se trata de un modelo muy simple, el cual se traduce a una única tabla con los siguientes campos:

Columna	Tipo de dato
id	integer
fichero	text
datos	xml
etapa	text
asignatura	text
idioma	text
tipo	text
curso	integer
nivel	text
observaciones	text

Tabla 5.3: Base de datos

Los atributos representan lo siguiente:

- **Id:** identificador y parte de la clave primaria. Es un número que aumenta con cada nueva entrada en la base de datos y no tiene ninguna limitación.
- **Fichero:** contiene el nombre del fichero. Es el otro elemento de la clave primaria, dado que cada fichero se puede identificar por su nombre.
- **Datos:** contiene el XML de dentro del fichero con todas sus etiquetas.
- **Etapa:** un campo opcional con la etapa educativa a la que corresponden las preguntas.
- **Asignatura:** otro campo opcional que contiene el nombre de la asignatura a la que corresponden las preguntas.
- **Idioma:** un tercer campo opcional que contiene el idioma en el que estén las preguntas.
- **Curso:** un identificador numérico del curso. También es opcional.
- **Nivel:** el nivel de la prueba. Opcional.
- **Observaciones:** un campo de texto para complementar alguna información que pueda ser de interés para localizar el fichero más tarde. Este último campo también es opcional.
- **Tipo:** el tipo de juego. Este campo es necesario para identificar de qué tipo es el juego y poder utilizarlo en el generador.

Para poder hacer uso de una base de datos PostgreSQL en el proyecto, ha sido necesario añadir la librería correspondiente. Los pasos que se han seguido son los siguientes:

- 1) Descargar el fichero jar más reciente de [15] (en este caso la versión 42.2.19).
- 2) Moverlo a la siguiente ruta: /home/dit/gamificacion/GeneradorFicheros/WebContent/WEB-INF/lib

- 3) En eclipse, acceder al Java Build Path mediante: *Project* → *Properties* → *Java Build Path*.
- 4) Se pulsa el botón «Add JARs...» y se selecciona el archivo añadido.
- 5) Se pulsa «Apply and Close».

Si todos los pasos se han seguido adecuadamente, antes de cerrar la ventana debería verse algo similar a la siguiente imagen:

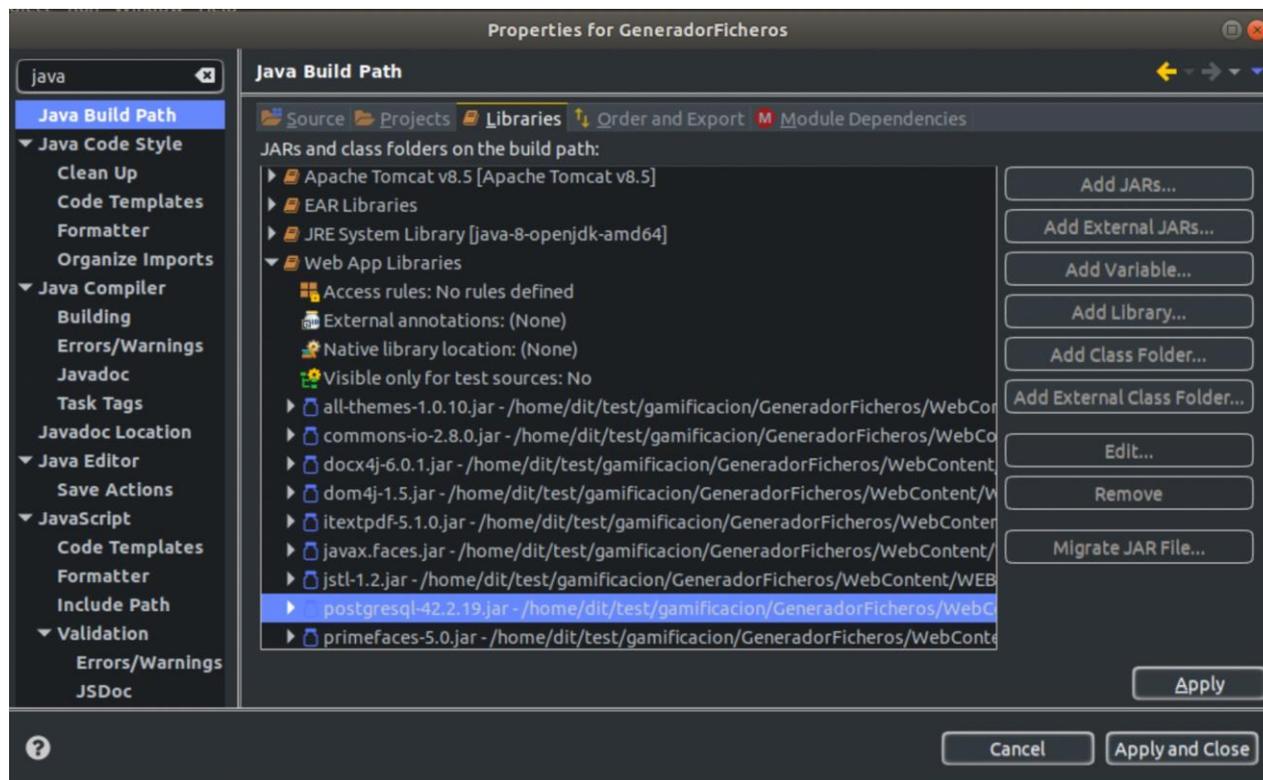


Figura 5.14: JAR añadido

5.4.3 Guardar en la base de datos

La primera función relacionada con las búsquedas es guardar un fichero generado en la base de datos. Esto se hace con el método `saveQuestion(AdminPreguntas b)`, el cual genera el contenido XML de las preguntas actuales, junto con el autor y título. También guarda los campos añadidos en la sección 5.2.1, como los campos con el mismo nombre de la base de datos.

Un detalle que debe tenerse en cuenta es que no se genera un fichero XML. En la base de datos se guarda el contenido y el nombre, pero el fichero como tal no existe en el servidor.

A continuación, se muestra un ejemplo de una entrada en la base de datos:

id	fichero	datos	etapa	asignatura	idioma	tipo	
53	Fichero1	<?xml version="1.0" standalone="no"?> <Gamificacion Id="1"> <titulo>Titulo3</titulo> <autor>Autor3</autor> <preguntas tipo="Tipo3"> <pregunta EmpiezaPor="" comodin50="" id="preg_0" sol="32"> <tema/> <enunciado numLineas="5">Pregunta</enunciado> </pregunta> </preguntas> </Gamificacion>	+	eso	Latín	cat	Tipo3

Figura 5.15: Ejemplo de entrada en la base de datos

En este ejemplo podemos ver como se guarda cada elemento en la base de datos: se guarda el nombre y el contenido, pero no hay ningún fichero.

5.4.4 Obtener resultados de la base de datos

Para buscar los elementos que hay en la base de datos se utiliza el buscador.xhtml. El *bean* que respalda la mayoría de las funciones es FiltroBuscador, el cual llena las listas desplegables al inicio, se encarga de llamar a las clases de la base de datos y gestiona la navegación al generador.

Por simplicidad, se muestra a continuación una tabla con cada campo que se puede buscar y su equivalencia en la tabla.

Campo del buscador	Elemento en la base de datos
Etapa	etapa
Asignatura	asignatura
Título	datos
Nombre	fichero
Autor	datos
Idioma	idioma
Tipo ficheros	tipo
Ficheros individuales	tipo
Palabra clave	datos
Número de preguntas	datos
Curso	curso
Nivel	nivel
Observaciones	observaciones

Tabla 5.4: Correspondencia buscador-base de datos

Cuando se presiona el botón de buscar, se leen los campos del buscador y se envían a la clase DBConnection, la cual hace una serie de consultas. Las que son un campo de texto son consultas básicas, pero las que utilizan el campo «datos» son un poco más elaboradas, ya que, como se observa en la tabla 5.2, es un campo de tipo XML.

Para obtener datos del XML se utiliza XPath. En el caso del autor y título se obtiene el texto del nodo correspondiente, mientras que para el número de preguntas se cuenta la cantidad de nodos del tipo pregunta. Por último, para el campo «palabra clave», se extrae todo el texto de todos los nodos, se convierte a *string* y ahí se busca la correspondencia con el texto solicitado.

Una vez obtenidas las entradas de la tabla necesarias, se rellena una lista de elementos del tipo ResultTableItem, el cual contiene el nombre fichero y tipo de resultado de la búsqueda. Esta lista se muestra en el buscador para que el usuario pueda ver el resultado de su búsqueda.

5.4.5 Funciones con los resultados obtenidos

Una vez hecha una búsqueda, el usuario puede realizar cuatro acciones nuevas: consultar el contenido de los ficheros, borrarlos de la búsqueda, utilizarlos para generar su propio fichero y eliminarlos de la base de datos.

5.4.5.1 Consultar

Si el usuario presiona el nombre de cualquier fichero, se creará en una pestaña nueva una vista del generador con el contenido del fichero, mostrándose además el nombre del fichero y los datos asociados a él.

Esto se hace pasando como parámetro en la URL el nombre del fichero seleccionado. Cuando llegue al generador, el *bean* AdminPreguntas lo leerá como un metadato y obtendrá el contenido consultando a la base de datos, la cual devolverá el tipo y contenido.

A continuación, se muestra un ejemplo visual del funcionamiento de la consulta:

The screenshot displays a web application interface. At the top, a dark header reads "Ficheros encontrados". Below it is a table with two columns: "Nombre del fichero" and "Tipo". The table contains one row with the value "Fichero1" in the first column and "Tipo3" in the second. The "Fichero1" text is circled in red, with a red arrow pointing from the circle to a dropdown menu labeled "Tipo 3" in the section below. Below the table are three buttons: "Borrar de la lista", "Usar Ficheros (Max 5)", and "Eliminar de la base de datos".

The main content area shows a dropdown menu set to "Tipo 3" with the text "Al cambiar el tipo de juego, se borrarán las preguntas de otro tipo." Below this, the "Título del fichero" is "Fichero1". There are three input fields: "Idioma:" (set to "Inglés"), "Etapa:" (set to "Bachillerato"), and "Asignatura:" (set to "TFG"). Below these are "Curso:" (set to "2") and "Nivel:" (set to "Avanzado"), and "Observaciones:" (set to "Ob"). A "Borrar Preguntas" button is present.

Below the form are two input fields: "Autor:" (set to "Autor1") and "Título:" (set to "Titulo1").

At the bottom, a dark header reads "Pregunta 1" with buttons "Eliminar pregunta" and "Duplicar pregunta". Below this is a form titled "Pregunta" with an "Enunciado:" field (set to "Pregunta") and a "Solución numérica:" field (set to "32").

At the very bottom are three buttons: "Nueva Pregunta", "Crear fichero XML", and "Guardar en BD" (with "Fichero1" in a text box next to it).

Figura 5.16: Ejemplo de consulta

5.4.5.2 Utilizar Ficheros

En caso de que el usuario quiera crear un nuevo cuestionario, puede combinar varios ficheros que hayan aparecido en su búsqueda. Para ello, debe marcar los ficheros que desee y pulsar el botón de «usar ficheros».

Para que esto funcione, deben cumplirse dos requisitos. El primero es que todos los ficheros deben ser del mismo formato, de lo contrario los ficheros no serán compatibles. El segundo es que se pueden seleccionar un máximo de 5 ficheros. En caso de que algún requisito no se cumpla, se alertará al usuario con un mensaje de error, el cual tendrá este aspecto:

Ficheros encontrados

Error de compatibilidad Todos los ficheros deben ser del mismo formato.

<input checked="" type="checkbox"/>	Nombre del fichero	Tipo
<input checked="" type="checkbox"/>	jurado	TipoJurado
<input checked="" type="checkbox"/>	Fichero1	Tipo3

Borrar de la lista Usar Ficheros (Max 5)

Eliminar de la base de datos

Figura 5.17: Ejemplo de uso erróneo

Una vez pulsado el botón y los requisitos se cumplan, se navegará al generador y se seguirá el mismo proceso que se utilizó en la consulta para mostrar el contenido de los ficheros, con la diferencia de que ahora puede haber más de uno.

A continuación, se muestra un ejemplo de un uso con éxito:

Ficheros encontrados

<input checked="" type="checkbox"/>	Nombre del fichero	Tipo
<input checked="" type="checkbox"/>	test4	Tipo1
<input checked="" type="checkbox"/>	test1	Tipo1

Borrar de la lista **Usar Ficheros (Max 5)** Eliminar de la base de datos

Pregunta 1

Eliminar pregunta Duplicar pregunta

Enunciado:

Opción 1:

Opción 2:

Opción 3:

Opción 4:

Añadir Opción

Opción correcta:
 Opción 1 Opción 2 Opción 3 Opción 4

Pregunta 2

Eliminar pregunta Duplicar pregunta

Enunciado:

Opción 1:

Opción 2:

Opción 3:

Opción 4:

Añadir Opción

Figura 5.18: Ejemplo de uso

5.4.5.3 Borrar de la lista y Eliminar de la base de datos

Si se desea eliminar un fichero de la base de datos, es tan sencillo como marcar los que se deseen borrar y pulsar el botón «eliminar de la base de datos». En este caso no hay ninguna restricción, pero sí que aparece un mensaje de confirmación. Una vez confirmado se llamará a DBConnection y se borrarán los datos. Por último, se actualizará la tabla para que no se muestren los datos borrados.

En caso de solo querer borrar de la búsqueda, pero no de la base de datos, se puede pulsar el botón «borrar». Esto actualizará dicha búsqueda excluyendo los ficheros marcados, pero no los borrará de la base de datos.

6 FICHEROS SOPORTADOS

«Hacer preguntas es prueba de que se piensa».

Rabindranath Tagore

La aplicación AJDA contiene multitud de juegos, de los cuales una gran mayoría permiten utilizar ficheros los ficheros de preguntas a los que se da soporte. A continuación, se muestra una tabla con todos los tipos que el proyecto soporta actualmente, junto con el número de juegos que existen de cada tipo:

Formato de juego	Tipo de juego	Número de juegos
Paneles sí o no	2	3
	18	9
Columnas para relacionar	16	4
	Parejas Ocultas	1
	De par en par	1
	Del tirón	1
Respuestas Con campo de texto	4	2
	Palabra Imposible	1
	La escalera	1
	Pasapalabra	1
	Empieza por	1
	Identity	1
Respuestas con opciones	1	300
	5	2
	7	3
	12	4

Formato de juego	Tipo de juego	Número de juegos
Respuestas con opciones	13	11
	14	2
	15	3
	17	2
	19	2
	20	8
Contar letras	Letritas que faltan	1
	Cuántas Letras	1
Paneles de letras	6	2
	8	2
	10	4
Respuesta abierta	Jurado	11
	Un, Dos, Tres	1
Respuestas con cifras	3	8
	9	2
	La patata caliente	1

Tabla 6.1: Tabla de tipos

Cada formato de juego contiene varios tipos de fichero, que pueden ser de dos tipos: comunes o individuales. Los comunes son, como su propio nombre indica, comunes a varios tipos de juegos. Se identifican por un número (a excepción del «Un, Dos, Tres», que es individual). Por otro lado, los individuales corresponden a un único tipo de juego y se identifican por tener el nombre de dicho juego en el tipo.

Los apartados a continuación detallan en qué consiste cada tipo de juego. Es importante tener en cuenta que, salvo que se indique lo contrario, todos los campos son obligatorios.

6.1 Paneles sí o no

Estos tipos de preguntas contienen un panel con un enunciado y varias respuestas, con el aspecto que se muestra a continuación:

Panel		
Enunciado:	<input type="text" value="Enunciado"/>	SI/NO
Respuesta 1:	<input type="text" value="Respuesta 1"/>	Si <input type="button" value="v"/>
Respuesta 2:	<input type="text" value="Respuesta 2"/>	No <input type="button" value="v"/>
Respuesta 3:	<input type="text" value="Respuesta 3"/>	No <input type="button" value="v"/>
Respuesta 4:	<input type="text" value="Respuesta 4"/>	Si <input type="button" value="v"/>
Respuesta 5:	<input type="text" value="Respuesta 5"/>	Si <input type="button" value="v"/>
Respuesta 6:	<input type="text" value="Respuesta 6"/>	Si <input type="button" value="v"/>
Respuesta 7:	<input type="text" value="Respuesta 7"/>	Si <input type="button" value="v"/>
Respuesta 8:	<input type="text" value="Respuesta 8"/>	Si <input type="button" value="v"/>

Figura 6.1: Paneles Sí o No

Los tipos de juego 2 y 18 son muy similares, con la única diferencia siendo que en el tipo 2 el panel contiene 8 elementos, mientras que en el tipo 18 contiene 16.

6.2 Columnas para relacionar

Las respuestas de columnas para relacionar contienen dos columnas distintas, las cuales relacionan cada respuesta con una solución. A continuación, se puede ver el aspecto:

Relacionar columnas	
<input type="text" value="Respuesta columna 1"/>	<input type="text" value="Solucion columna 1"/>
<input type="text" value="Respuesta columna 2"/>	<input type="text" value="Solucion columna 2"/>
<input type="text" value="Respuesta columna 3"/>	<input type="text" value="Solucion columna 3"/>
<input type="text" value="Respuesta columna 4"/>	<input type="text" value="Solucion columna 4"/>
<input type="text" value="Respuesta columna 5"/>	<input type="text" value="Solucion columna 5"/>
<input type="text" value="Respuesta columna 6"/>	<input type="text" value="Solucion columna 6"/>
<input type="text" value="Respuesta columna 7"/>	<input type="text" value="Solucion columna 7"/>
<input type="text" value="Respuesta columna 8"/>	<input type="text" value="Solucion columna 8"/>
<input type="text" value="Respuesta columna 9"/>	<input type="text" value="Solucion columna 9"/>
<input type="text" value="Respuesta columna 10"/>	<input type="text" value="Solucion columna 10"/>

Figura 6.2: Columnas para relacionar

Pese a contar con diversos tipos de fichero, todos se tratan de la misma forma en el generador. Su diferencia está en la implementación dentro del juego.

6.3 Respuestas con campo de texto

Como su propio nombre indica, este tipo preguntas se responden con un campo de texto. Tienen el siguiente aspecto:

Figura 6.3: Respuestas con campo de texto

Aquí existen diferencias entre los formatos. Todos los tipos tienen una pista, pero los tipos Identity y Escalera contienen un único campo de solución, mientras que el resto tres.

6.4 Respuestas con opciones

Este formato de juego es el de mayor importancia, dado que 343 juegos utilizan este formato, siendo 300 de tipo 1. Se componen de ficheros con un enunciado y una respuesta correcta, de entre un número variable de opciones.

El aspecto básico es este:

Figura 6.4: Respuestas con opciones

La principal diferencia entre los tipos de este formato es el número de respuestas, el cual se puede ver en la siguiente tabla:

Tipo de juego	Nº de respuestas
1	4
5	3
7	4
12	2 a 4
13	6
14	2
15	2 a 4
17	6
19	4 o 5
20	2

Tabla 6.2: Tipos de respuestas con opciones

Además del número de respuestas, algunos de estos tipos de juegos son temáticos, por lo que tienen un campo extra llamado «Tema». Estos juegos son los tipos 5, 12, 15 y 17. En el caso del 17 el tema se aplica a todo el fichero, mientras que en el resto es un campo de cada pregunta.

6.5 Contar letras

Los ficheros de contar letras consisten en una palabra y una lista de letras, cada una con un número de repeticiones distinto:

Figura 6.5: Contar letras

Ambos tipos de juego permiten añadir hasta 18 letras, pero no es necesario rellenar cada uno de los campos. Por último, están limitados a una única pregunta, desactivándose el botón «Nueva Pregunta» al generar la primera.

6.6 Paneles de letras

Estos juegos contienen preguntas con un enunciado y una cantidad variable de pistas:

Figura 6.6: Paneles de letras

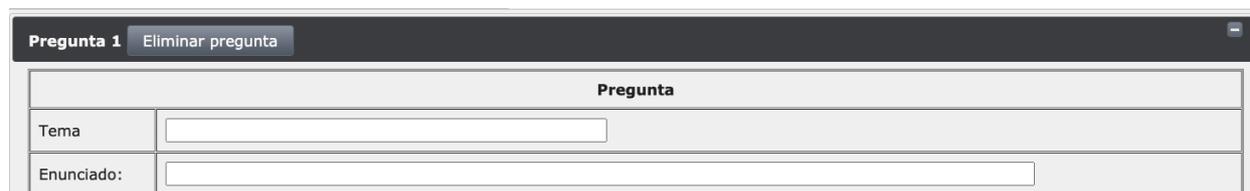
Cada tipo de juego contiene una cantidad de pistas distintas, que se pueden ver en la tabla a continuación:

Tipo de juego	Número de pistas
6	8
8	3
10	9

Tabla 6.3: Tipos de paneles de letras

6.7 Respuesta abierta

Ambos juegos de este formato son muy sencillos, pues contienen únicamente un campo con el tema y otro con el texto:



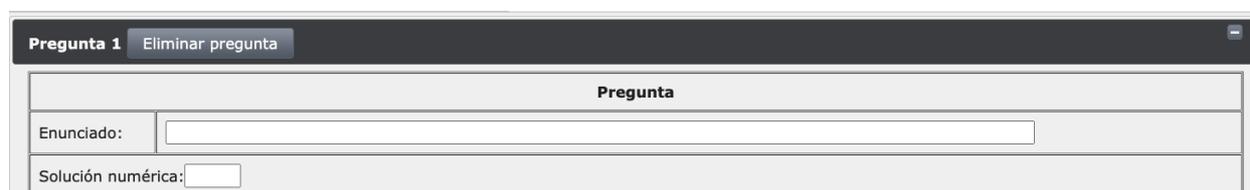
The screenshot shows a window titled "Pregunta 1" with a close button and a button labeled "Eliminar pregunta". Below the title bar is a header "Pregunta". The main area contains two input fields: "Tema" and "Enunciado".

Pregunta	
Tema	<input type="text"/>
Enunciado:	<input type="text"/>

Figura 6.7: Respuesta abierta

6.8 Respuestas con cifras

Por último, los tipos de ficheros de respuestas con cifra son también todos idénticos. Contienen el enunciado y un campo con la solución numérica:



The screenshot shows a window titled "Pregunta 1" with a close button and a button labeled "Eliminar pregunta". Below the title bar is a header "Pregunta". The main area contains two input fields: "Enunciado:" and "Solución numérica:".

Pregunta	
Enunciado:	<input type="text"/>
Solución numérica:	<input type="text"/>

Figura 6.8: Respuestas con cifras

7 PLAN DE PRUEBAS

«No importa lo estupendamente que haya ido la demo en los ensayos, cuando lo haces frente a tu audiencia la probabilidad de que sea una presentación exitosa es inversamente proporcional al número de personas mirando»

Mark Gibbs

7.1 Introducción

Para asegurar el correcto funcionamiento del proyecto, se ha elaborado un nuevo plan de pruebas cuyo objetivo es garantizar el funcionamiento de los distintos nuevos componentes que se han integrado, así como su integración con el resto de módulos ya existentes.

Es importante mencionar que para asegurar que el conjunto del proyecto funciona adecuadamente, las pruebas anteriores deben repetirse y han de completarse con éxito, salvo que alguna funcionalidad se haya visto modificada o eliminada.

Las pruebas se han separado en tres grandes grupos para poder distinguir, en caso de fallo, la causa de los errores. Los grupos son los siguientes:

- Pruebas unitarias: se comprueba el funcionamiento de cada función de forma aislada. El éxito de cada una de estas pruebas debe ser completamente independiente del resto y no deben utilizar elementos comunes. La suma de todas estas pruebas debe también cubrir la totalidad del código añadido.
- Pruebas de integración: estas pruebas deben ejecutarse únicamente una vez que todas las pruebas unitarias se completen con éxito. Deben asegurar que las nuevas funcionalidades funcionan bien entre sí y son compatibles con las ya existentes.
- Pruebas de validación: se asegura que lo que haga el programa es lo solicitado. El usuario final debe hacer uso de las distintas funciones y quedar satisfecho con el resultado. Para este proyecto no se han añadido nuevas pruebas de validación.

7.2 Pruebas

7.2.1 Pruebas unitarias

Nombre	PU-07
Objetivo	Exportación de fichero con formato X a formato DOC
Prerrequisitos de la prueba	Tener en el equipo un fichero XML de tipo N Tener instalado un procesador de texto (Microsoft Word o similar)
Procedimiento	<ol style="list-style-type: none"> 1) Se subirá el fichero en la opción de exportar 2) Se presionará el botón «DOC» 3) Se abrirá el archivo descargado 4) Se elegirá la codificación por defecto del programa
Superación de la prueba	El fichero abierto contendrá texto legible, incluyendo al final la marca «//TipoN»
Comentarios	X representa cada formato posible de prueba (si o no, campo de texto, etc.), por lo que esta prueba deberá repetirse tantas veces como pruebas existan. N representa el mínimo de todos los posibles tipos de fichero dentro de cada opción. Por ejemplo, en las respuestas con opciones la prueba se hará con el Tipo 1; en las respuestas con cifras, el Tipo 3, etc.

Tabla 7.1: PU-07

Nombre	PU-08
Objetivo	Exportación de fichero con formato X a formato CSV
Prerrequisitos de la prueba	Tener en el equipo un fichero XML de tipo N Tener instalado un programa de hojas de cálculo (Microsoft Excel o similar)
Procedimiento	<ol style="list-style-type: none"> 1) Se subirá el fichero en la opción de exportar 2) Se presionará el botón «CSV» 3) Se abrirá el archivo descargado 4) Se elegirá la codificación por defecto del programa y una columna por cada línea de texto
Superación de la prueba	El fichero abierto contendrá celdas legibles, incluyendo al final la marca «Compatibilidad N»
Comentarios	X representa cada formato posible de prueba (si o no, campo de texto, etc.), por lo que esta prueba deberá repetirse tantas veces como pruebas existan. N representa el mínimo de todos los posibles tipos de fichero dentro de cada opción. Por ejemplo, en las respuestas con opciones la prueba se hará con el Tipo 1; en las respuestas con cifras, el Tipo 3, etc.

Tabla 7.2: PU-08

Nombre	PU-09
Objetivo	Exportación de fichero con formato X a formato GIFT
Prerrequisitos de la prueba	Tener en el equipo un fichero XML de tipo N Tener instalado un editor de texto
Procedimiento	<ol style="list-style-type: none"> 1) Se subirá el fichero en la opción de exportar 2) Se presionará el botón «GIFT» 3) Se abrirá el archivo descargado con cualquier editor de texto
Superación de la prueba	El fichero abierto contendrá texto legible, incluyendo al final la marca «//TipoN»
Comentarios	X representa cada formato posible de prueba (si o no, campo de texto, etc.), por lo que esta prueba deberá repetirse tantas veces como pruebas existan. N representa el mínimo de todos los posibles tipos de fichero dentro de cada opción. Por ejemplo, en las respuestas con opciones la prueba se hará con el Tipo 1; en las respuestas con cifras, el Tipo 3, etc.

Tabla 7.3: PU-09

Nombre	PU-10
Objetivo	Importación de un fichero con formato DOC a la plataforma
Prerrequisitos de la prueba	PU-07
Procedimiento	<ol style="list-style-type: none"> 1) Se elegirá el formato y tipo del archivo que vaya a ser subido 2) Se subirá el archivo seleccionado a la opción de cargar 3) Se pulsará el botón de importar
Superación de la prueba	Deberán mostrarse las preguntas del fichero en la página
Comentarios	

Tabla 7.4: PU-10

Nombre	PU-11
Objetivo	Importación de un fichero con formato CSV a la plataforma
Prerrequisitos de la prueba	PU-08
Procedimiento	<ol style="list-style-type: none"> 1) Se elegirá el formato y tipo del archivo que vaya a ser subido 2) Se subirá el archivo seleccionado a la opción de cargar 3) Se pulsará el botón de importar
Superación de la prueba	Deberán mostrarse las preguntas del fichero en la página
Comentarios	

Tabla 7.5: PU-11

Nombre	PU-12
Objetivo	Importación de un fichero con formato GIFT a la plataforma
Prerrequisitos de la prueba	PU-09 con un fichero de formato «Respuestas con opciones»
Procedimiento	<ol style="list-style-type: none"> 1) Se elegirá el formato «Respuestas con opciones» 2) Se elegirá el tipo del archivo que vaya a ser subido 3) Se subirá el archivo seleccionado a la opción de cargar 4) Se pulsará el botón de importar
Superación de la prueba	Deberán mostrarse las preguntas del fichero en la página
Comentarios	

Tabla 7.6: PU-12

Nombre	PU-13
Objetivo	Guardar fichero en la base de datos
Prerrequisitos de la prueba	Tener al menos una pregunta rellena en cualquier tipo de fichero
Procedimiento	<ol style="list-style-type: none"> 1) Se elige idioma inglés 2) Se elige etapa infantil 3) Se rellena el campo asignatura con «TFG» 4) Se rellena el campo autor con «Autor1» 5) Se rellena el campo título con «Título1» 6) Se rellena el campo guardar preguntas con «ejemplo» 7) Se pulsa el botón «Guardar Preguntas»
Superación de la prueba	Aparece una notificación que indica que el fichero se ha guardado con éxito en la base de datos
Comentarios	

Tabla 7.7: PU-13

Nombre	PU-14
Objetivo	Búsqueda por etapa
Prerrequisitos de la prueba	PU-13
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se elige etapa infantil 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado con el nombre «ejemplo»
Comentarios	

Tabla 7.8: PU-14

Nombre	PU-15
Objetivo	Búsqueda por asignatura
Prerrequisitos de la prueba	PU-13
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se rellena el campo asignatura con «TFG» 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado con el nombre «ejemplo»
Comentarios	

Tabla 7.9: PU-15

Nombre	PU-16
Objetivo	Búsqueda por título
Prerrequisitos de la prueba	PU-13
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se rellena el campo título del fichero con «Título1» 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado con el nombre «ejemplo»
Comentarios	

Tabla 7.10: PU-15

Nombre	PU-17
Objetivo	Búsqueda por nombre
Prerrequisitos de la prueba	PU-13
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se rellena el campo título del fichero con «ejemplo» 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado con el nombre «ejemplo»
Comentarios	

Tabla 7.11: PU-16

Nombre	PU-17
Objetivo	Búsqueda por autor
Prerrequisitos de la prueba	PU-13
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se rellena el campo título del fichero con «Autor1» 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado con el nombre «ejemplo»
Comentarios	

Tabla 7.12: PU-17

Nombre	PU-18
Objetivo	Búsqueda por idioma
Prerrequisitos de la prueba	PU-13
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se selecciona el idioma inglés 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado con el nombre «ejemplo»
Comentarios	

Tabla 7.13: PU-18

Nombre	PU-19
Objetivo	Búsqueda por idioma
Prerrequisitos de la prueba	PU-13
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se selecciona el idioma inglés 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado con el nombre «ejemplo»
Comentarios	

Tabla 7.14: PU-19

Nombre	PU-20
Objetivo	Búsqueda por tipo de fichero
Prerrequisitos de la prueba	Se debe tener en la base de datos algún fichero guardado
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se selecciona el tipo del fichero guardado con la opción de tipo de ficheros o ficheros individuales 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero
Comentarios	

Tabla 7.15: PU-20

Nombre	PU-21
Objetivo	Búsqueda por palabra clave
Prerrequisitos de la prueba	Se debe tener en la base de datos algún fichero guardado
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se busca el enunciado de alguna pregunta del fichero guardado en la base de datos 3) Se pulsa el botón «Buscar»
Superación de la prueba	Se muestra al menos un fichero encontrado
Comentarios	

Tabla 7.16: PU-21

Nombre	PU-22
Objetivo	Filtrar por número de preguntas
Prerrequisitos de la prueba	Se debe tener en la base de datos un fichero guardado con al menos 5 preguntas, cuyo autor sea «Autor1»
Procedimiento	<ol style="list-style-type: none"> 1) Si hay algún fichero visible, se marca y se pulsa el botón «Borrar» 2) En la página del buscador, se rellena el campo título del fichero con «Autor1» 3) En la opción de número de preguntas, se desliza el botón hasta que se muestre en pantalla «Entre 0 y 4 preguntas» 4) Se pulsa el botón «Buscar»
Superación de la prueba	No se muestra el fichero guardado cuyo autor es «Autor1» y tiene al menos 5 preguntas
Comentarios	

Tabla 7.17: PU-22

Nombre	PU-23
Objetivo	Vista previa de un fichero
Prerrequisitos de la prueba	PU-17
Procedimiento	<ol style="list-style-type: none"> 1) Se presiona el nombre del fichero
Superación de la prueba	Se muestra una nueva pestaña con el contenido del fichero en el generador
Comentarios	

Tabla 7.18: PU-23

7.2.2 Pruebas de integración

Nombre	PI-03
Objetivo	Composición de preguntas con varios ficheros
Prerrequisitos de la prueba	Se debe tener la base de datos de prueba, disponible junto al proyecto
Procedimiento	<ol style="list-style-type: none"> 1) Se realiza una búsqueda de Tipo 1 2) Se seleccionan los ficheros test1, test2, test3, test4 y test5 3) Se pulsa el botón «Usar ficheros»
Superación de la prueba	Se navega al generador con todas las preguntas
Comentarios	

Tabla 7.19: P1-03

Nombre	PI-04
Objetivo	Eliminación de preguntas de la base de datos
Prerrequisitos de la prueba	Se debe tener la base de datos de prueba, disponible junto al proyecto
Procedimiento	<ol style="list-style-type: none"> 1) Se realiza una búsqueda de Tipo 1 2) Se seleccionan los ficheros test1, test2, test3, test4 y test5 3) Se pulsa el botón «Eliminar de la base de datos» 4) Se confirma el mensaje de alerta 5) Se repite el primer paso
Superación de la prueba	Los ficheros desaparecen de la búsqueda y no vuelven a aparecer al repetir la búsqueda
Comentarios	

Tabla 7.20: P1-04

Nombre	PI-05
Objetivo	Creación de un nuevo fichero de preguntas en función de otros
Prerrequisitos de la prueba	PI-03
Procedimiento	<ol style="list-style-type: none"> 1) Se elimina la cuarta pregunta 2) Se edita el enunciado y respuestas de la última pregunta con cualquier texto alternativo 3) Se pulsa el botón «Crear fichero XML» 4) Se descarga el fichero y se abre
Superación de la prueba	El fichero XML debe contener todas las preguntas de los 5 ficheros con sus modificaciones
Comentarios	

Tabla 7.21: P1-05

8 CONCLUSIONES

«Nunca confíes en un ordenador que no puedas lanzar por una ventana»

Steve Wozniak

A lo largo de este proyecto se han visto una serie de funcionalidades tanto ya existentes como añadidas, con la esperanza de que sean de utilidad a personal docente de cualquier ámbito. Se espera que los añadidos de formato faciliten aún más el trabajo de los profesores y que la base de datos permita agilizar su trabajo desde cualquier sitio.

A título personal, considero de especial interés mi evolución con el proyecto. En estos meses he aprendido a familiarizarme con nuevas tecnologías de forma completamente autodidacta, lo cual será imprescindible en un futuro. También, gracias al proyecto, tengo una idea de qué puede suponer el desarrollo web en el mundo laboral: requisitos de un cliente, fechas límite, priorización de funcionalidades o mantener un código limpio y legible son algunos de los aspectos a los que me he tenido que habituar.

Por último, me gustaría destacar uno de los obstáculos más interesantes a los que me he tenido que enfrentar: trabajar con el código de otra persona. El proyecto de Ramos León es, sin duda, un gran proyecto. Sin embargo, la falta de una documentación formal y de unas aclaraciones por parte del autor del código (más allá de los comentarios) han hecho que haya tenido una dificultad añadida.

9 LÍNEAS DE MEJORA

«La tecnología es una palabra que describe algo que no funciona todavía».

Douglas Adams

Debido a la gran extensión del proyecto, es lógico pensar que tiene margen de mejora. Por ello, en este apartado se proponen unas mejoras de funciones incompletas, así como nuevas funciones que puedan mejorar la experiencia de usuario.

- **Importación de GIFT a otros formatos.** De momento, el formato GIFT solo puede importarse a la plataforma si es del tipo con opciones. Sería interesante poder importar otros tipos de fichero para así completar la compatibilidad con todos los tipos.
- **Refactorización del código.** El código del proyecto está, actualmente, en un estado que podría limitar su mantenimiento a largo plazo. Esto es esperable, ya que en él hemos trabajado alumnos que no tenemos experiencia con proyectos a gran escala y hemos utilizado algunas de las tecnologías del proyecto por primera vez.
- **Mejora del diseño.** El aspecto visual es un apartado que a menudo se descuida. Esto ha hecho que el aspecto visual de la página no sea el más atractivo posible. Intentar tener una idea clara de cómo debería verse la página y plasmar ese diseño podría hacer que las mismas herramientas se vean de forma mucho más llamativa.
- **Mejoras en la funcionalidad ya existente.** Hay algunos cambios que se podrían hacer a partes de las funcionalidades que ya existen:
 - El primer cambio, es un botón que permita editar un fichero de la base de datos, de forma que el autor de dicho fichero no tenga que borrarlo e incluirlo de nuevo.
 - El segundo cambio, es añadir caracteres de escape a los formatos de fichero que se importan. Esto permitiría poder generar ficheros con caracteres que, de momento, no son posibles. Un ejemplo de ello es la virgulilla en el caso de GIFT y DOC «~».
 - Por último, también se debería imponer la restricción de que un autor solo pueda eliminar sus propios ficheros, no los de cualquier usuario.

10 REFERENCIAS

- [1] Ramos León, Carlos, *Aplicación web para la creación y modificación de ficheros de juegos para la Aplicación de Juegos Didácticos en el Aula (AJDA)*, 2018 Escuela Técnica Superior de Ingeniería,
- [2] Muñoz Calle, Jesús Manuel, *Aplicación de Juegos Didácticos en el Aula*, ND. [Recurso en línea]. Disponible en: <http://newton.proyectodescartes.org/juegosdidacticos/index.php>
- [3] Oracle, *JavaServer Faces Technology*, 2018. [Recurso en línea]. Disponible en: <https://www.oracle.com/java/technologies/javaserverfaces.html>
- [4] Schalk, Chris, *Introduction to Javasever Faces - What is JSF?*, 2005. [Recurso en línea]. Disponible en: <https://www.oracle.com/topics/technologies/jsf.html>
- [5] Tutorials Point, *JSF – Architecture*, ND. [Recurso en línea]. Disponible en: https://www.tutorialspoint.com/jsf/jsf_architecture.htm
- [6] PostgreSQL, *PostgreSQL*, ND [Recurso en línea]. Disponible en: <https://www.postgresql.org/>
- [7] W3Schools, *XML and XPath*, ND. [Recurso en línea]. Disponible en: https://www.w3schools.com/xml/xml_xpath.asp
- [8] Mozilla, *XSLT: Extensible Stylesheet Language Transformations*, ND. [Recurso en línea]. Disponible en: <https://developer.mozilla.org/en-US/docs/Web/XSLT>
- [9] Oracle, *VirtualBox*, ND. [Recurso en línea]. Disponible en: <https://www.virtualbox.org/>
- [10] Eclipse Foundation, *Eclipse IDE*, ND. [Recurso en línea]. Disponible en: <https://www.eclipse.org/eclipseide/>
- [11] The Apache Software Foundation, *Apache Tomcat*, ND. [Recurso en línea]. Disponible en: <http://tomcat.apache.org/>
- [12] Martínez Delgado, José Manuel *Carga del Entorno de desarrollo (Guía del Desarrollador)*, 2021. [Recurso en línea]. Disponible en: [https://gestionproyectos.us.es/attachments/download/43916/Carga%20del%20Entorno%20de%20desarrollo%20\(Guia%20del%20Desarrollador\).pdf](https://gestionproyectos.us.es/attachments/download/43916/Carga%20del%20Entorno%20de%20desarrollo%20(Guia%20del%20Desarrollador).pdf)
- [13] Kariv, Adam & Pollock, Rufus, *CSV - Comma Separated Values*, ND. [Recurso en línea]. Disponible en: <https://datahub.io/docs/data-packages/csv>
- [14] Moodle, *GIFT Format*, 2021. [Recurso en línea]. Disponible en: https://docs.moodle.org/311/en/GIFT_format
- [15] PostgreSQL, *PostgreSQL JDBC Driver*, ND. [Recurso en línea]. Disponible en: <https://jdbc.postgresql.org/download.html>

ANEXO A: DOCUMENTACIÓN

El objetivo de este anexo es ofrecer una documentación estructurada de las distintas clases y métodos que componen el proyecto, de forma que se pueda consultar fácilmente para extraer información sin necesidad de leer el código.

En primer lugar, es posible crear una documentación automática con Javadoc del estado actual del proyecto. Para ello, en eclipse vamos a *Project* → *Generate Javadoc* y una vez ahí, seleccionamos el proyecto *GeneradorFicheros* y aceptamos. Esto generará una documentación en la ruta */home/dit/gamificacion/GeneradorFicheros/doc*.

El resto del anexo está estructurado de forma que cada punto consta de un paquete, dentro del cual se enumeran las clases de dicho paquete. En el contenido de las clases se mencionan los métodos más importantes, así como cualquier aclaración adicional que permita entender su utilidad.

A. Conversor

A.1 CreateXML

A.1.1.1 `public void createXMLFile(AdminPreguntas b)`

Se utiliza para convertir las preguntas actualmente en el generador en un fichero XML para que el usuario descargue.

- **Llamada por:** botón «Crear fichero XML»
- **Parámetros de entrada:**
 - *b*: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)

A.1.2 Exportador

A.1.2.1 `public void exportadorPDF()`

Llama a la clase `XMLtoPDF` si se cumplen las condiciones necesarias.

- **Llamada por:** botón «PDF»

A.1.2.2 `public void exportadorTXT()`

Llama a la clase `XMLtoTXT` si se cumplen las condiciones necesarias.

- **Llamada por:** botón «TXT»

A.1.2.3 public void exportadorDOC()

Llama a la clase XMLtoDOC si se cumplen las condiciones necesarias.

- **Llamada por:** botón «DOC»

A.1.2.4 public void exportadorCSV()

Llama a la clase XMLtoCSV si se cumplen las condiciones necesarias.

- **Llamada por:** botón «CSV»

A.1.2.5 public void exportadorGIFT()

Llama a la clase XMLtoGIFT si se cumplen las condiciones necesarias.

- **Llamada por:** botón «GIFT»

A.1.3 XMLtoCSV

A.1.3.1 public void convertXMLtoCSV(InputStream XML)

Convierte un XML a CSV y lo descarga en la máquina del usuario.

- **Llamada por:** exportadorCSV
- **Parámetros de entrada:**
 - XML: el fichero XML necesario para la conversión.

A.1.4 XMLtoDOC

A.1.4.1 public void convertXMLtoDOC(InputStream XML)

Lee el tipo de pregunta del fichero XML y le envía el contenido de dicho XML, junto con un *printstream* que será el fichero de salida, a la clase correspondiente del paquete `converterGIFT` para que lo transforme.

- **Llamada por:** exportadorDOC
- **Parámetros de entrada:**
 - XML: el fichero XML necesario para la conversión.

A.1.5 XMLtoGIFT

A.1.5.1 public void convertXMLtoGIFT(InputStream XML)

Lee el tipo de pregunta del fichero XML y le envía el contenido de dicho XML, junto con un *printstream* que será el fichero de salida, a la clase correspondiente del paquete `converterGIFT` para que lo transforme.

- **Llamada por:** exportadorGIFT
- **Parámetros de entrada:**
 - XML: el fichero XML necesario para la conversión.

A.1.6 XMLtoPDF

A.1.6.1 public void convertXMLtoPDF(InputStream XML)

Convierte un XML a PDF y lo descarga en la máquina del usuario

- **Llamada por:** exportadorPDF

- **Parámetros de entrada:**
 - XML: el fichero XML necesario para la conversión.

A.2 ConversorGIFT

A.2.1 FichTipoAbiertaGift

A.2.1.1. public void imprimirTipoAb(Document doc, PrintStream ps)

Convierte un fichero XML al formato GIFT/DOC del tipo respuesta abierta.

- **Llamada por:** XMLtoDOC y XMLtoGIFT
- **Parámetros de entrada:**
 - doc: el documento XML.
 - ps: el stream donde se imprime el contenido.

A.2.2 FichTipoCampoTextoGift

A.2.2.1 public void imprimirTipoCampText(Document doc, PrintStream ps)

Convierte un fichero XML al formato GIFT/DOC del tipo respuesta con campo de texto.

- **Llamada por:** XMLtoDOC y XMLtoGIFT
- **Parámetros de entrada:**
 - doc: el documento XML.
 - ps: el stream donde se imprime el contenido.

A.2.3 FichTipoAbiertaGift

A.2.3.1 public void imprimirTipoAb(Document doc, PrintStream ps)

Convierte un fichero XML al formato GIFT/DOC del tipo respuesta abierta.

- **Llamada por:** XMLtoDOC y XMLtoGIFT
- **Parámetros de entrada:**
 - doc: el documento XML.
 - ps: el stream donde se imprime el contenido.

A.2.4 FichTipoCifrasGift

A.2.4.1 public void imprimirTipoCifras(Document doc, PrintStream ps)

Convierte un fichero XML al formato GIFT/DOC del tipo respuestas con cifras.

- **Llamada por:** XMLtoDOCy XMLtoGIFT
- **Parámetros de entrada:**
 - doc: el documento XML.
 - ps: el stream donde se imprime el contenido.

A.2.5 FichTipoOpcionesGift

A.2.5.1 public void imprimirTipoOp(Document doc, PrintStream ps)

Convierte un fichero XML al formato GIFT/DOC del tipo respuestas con opciones.

- **Llamada por:** XMLtoDOC y XMLtoGIFT
- **Parámetros de entrada:**
 - doc: el documento XML.
 - ps: el stream donde se imprime el contenido.

A.2.6 FichTipoRelacionarGift

A.2.6.1 public void imprimirTipoRel(Document doc, PrintStream ps)

Convierte un fichero XML al formato GIFT/DOC del tipo columnas para relacionar.

- **Llamada por:** XMLtoDOC y XMLtoGIFT
- **Parámetros de entrada:**
 - doc: el documento XML.
 - ps: el stream donde se imprime el contenido.

A.2.7 FichTipoVFGift

A.2.7.1 public void imprimirTipoVF(Document doc, PrintStream ps)

Convierte un fichero XML al formato GIFT/DOC del tipo respuestas sí o no.

- **Llamada por:** XMLtoDOC y XMLtoGIFT
- **Parámetros de entrada:**
 - doc: el documento XML.
 - ps: el stream donde se imprime el contenido.

A.3 Lectura

A.3.1 Importador

A.3.1.1 public void importador(AdminPreguntas b, MenuMapTipos tipos, MenuMapTiposOpciones opcionesTipos)

Método que lee el fichero importado y lo envía a la clase que se encargará de pasarlo a preguntas si corresponde.

- **Llamada por:** botón «Importar»
- **Parámetros de entrada:**
 - b: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)
 - tipos: mapa de tipos para identificar el tipo.
 - opcionesTipos: mapa de tipos para obtener todas las posibles opciones.

A.3.2 ReadCSV

A.3.2.1 public InputSream transformFile(UploadedFile fichero)

Método auxiliar para convertir un `UploadedFile` a `InputStream`

- **Llamada por:** `Importador`

A.3.2.2 public void readFile(UploadedFile fichero, AdminPreguntas b, MenuMapTipos tipos, MenuMapTiposOpciones opcionesTipos)

Método que lee un fichero CSV y rellena los campos de la página con los datos obtenidos.

- **Llamada por:** `Importador`
- **Parámetros de entrada:**
 - `fichero`: el fichero CSV.
 - `b`: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)
 - `tipos`: mapa de tipos para identificar el tipo.
 - `opcionesTipos`: mapa de tipos para obtener todas las posibles opciones.

A.3.3 ReadGIFT

A.3.3.1 public void readFile(UploadedFile fichero, AdminPreguntas b, MenuMapTipos tipos, MenuMapTiposOpciones opcionesTipos)

Método que lee un fichero GIFT o DOC y genera un XML que enviará al lector de `readXML`.

- **Llamada por:** `Importador`
- **Parámetros de entrada:**
 - `fichero`: el fichero GIFT/DOC.
 - `b`: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)
 - `tipos`: mapa de tipos para identificar el tipo.
 - `opcionesTipos`: mapa de tipos para obtener todas las posibles opciones.

A.3.3.2 public void xmlStructure()

Genera la estructura básica del fichero XML para el método `readFile` de la misma clase.

A.3.3.3 public void addTitle(Document doc, String title)

Añade el título al XML.

A.3.3.4 public void addAuthor(Document doc, String autor)

Añade el autor al XML.

A.3.3.5 public void addType(Document doc, String type)

Añade el tipo al XML.

A.3.4 ReadTXT

A.3.4.1 `public void readFile(UploadedFile file, AdminPreguntas b, MenuMapTipos tipos, MenuMapTiposOpciones opcionesTipos)`

Método que lee un fichero TXT y rellena los campos de la página con los datos obtenidos.

- **Llamada por:** `Importador`
- **Parámetros de entrada:**
 - `file`: el fichero TXT.
 - `b`: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)
 - `tipos`: mapa de tipos para identificar el tipo.
 - `opcionesTipos`: mapa de tipos para obtener todas las posibles opciones.

A.3.5 ReadXML

A.3.5.1 `public void readFile(UploadedFile file, AdminPreguntas b, MenuMapTipos tipos, MenuMapTiposOpciones opcionesTipos)`

Método que lee un fichero XML y muestra las preguntas en la web.

- **Llamada por:** `ReadTXT` y `ReadCSV`
- **Parámetros de entrada:**
 - `file`: el fichero XML.
 - `b`: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)
 - `tipos`: mapa de tipos para identificar el tipo.
 - `opcionesTipos`: mapa de tipos para obtener todas las posibles opciones.

A.3.5.2 `public void readFile(Document document, AdminPreguntas b, MenuMapTipos tipos, MenuMapTiposOpciones opcionesTipos)`

Método similar al anterior que utiliza un documento de tipo `Document` en lugar de `UploadedFile`.

- **Llamada por:** `ReadGIFT`, `Importador` y `public void checkFile(...)` en `AdminPreguntas`
- **Parámetros de entrada:**
 - `document`: el fichero XML.
 - `b`: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)
 - `tipos`: mapa de tipos para identificar el tipo.
 - `opcionesTipos`: mapa de tipos para obtener todas las posibles opciones.

A.3.5.3 `private static Preguntaopciones getPreguntaopciones(Node node, String tipo, AdminPreguntas bb)`

Devuelve un elemento del tipo `Preguntaopciones` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.4 private static PreguntaCampoTexto getPreguntaCampoTexto (Node node, String tipo, AdminPreguntas bb)

Devuelve un elemento del tipo `PreguntaCampoTexto` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.5 private static PreguntaCifras getPreguntaCifras (Node node, String tipo, AdminPreguntas bb)

Devuelve un elemento del tipo `PreguntaCifras` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.6 private static PreguntaSiNo getPreguntaSiNo (Node node, String tipo, AdminPreguntas bb)

Devuelve un elemento del tipo `PreguntaSiNo` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.7 private static PreguntaPanelesLetras getPreguntaPanelesLetras (Node node, String tipo, AdminPreguntas bb)

Devuelve un elemento del tipo `PreguntaPanelesLetras` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.8 private static PreguntaRelacionar getPreguntaRelacionar (Node node, String tipo, AdminPreguntas bb)

Devuelve un elemento del tipo `PreguntaRelacionar` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.9 private static PreguntaRespAbierta getPreguntaRespAbierta (Node node, String tipo, AdminPreguntas bb)

Devuelve un elemento del tipo `PreguntaRespAbierta` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.10 private static PreguntaContarLetras getPreguntaContarLetras (Node node, String tipo, AdminPreguntas bb)

Devuelve un elemento del tipo `PreguntaContarLetras` que contendrá las preguntas de dicho tipo, para poder añadirlo al XML en la función `readFile` de la misma clase.

A.3.5.11 private static String getTagValuePregunta(String tag, Element element)

Método privado a para obtener el texto de las etiquetas XML recibidas.

A.4 AdministrarPreguntas

A.4.1 AdminPreguntas

Bean principal del generador. Se utiliza para rellenar las listas desplegables, guardar preguntas

A.4.1.1 public void saveQuestion(AdminPreguntas b)

Función que crea un fichero XML y lo guarda, junto con sus metadatos, en la base de datos.

- **Llamada por:** botón «Guardar Preguntas»

- **Parámetros de entrada:**

- *b*: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)

A.4.1.2 Métodos add y delete

Conjunto de métodos que permite añadir y borrar preguntas y respuestas de la lista de preguntas visible en el generador.

- **Llamadas por:** botones «Nueva Pregunta», «Eliminar Pregunta», «Añadir Respuesta» y «X» (junto a las respuestas en el formato con opciones).
- **Parámetros de entrada:**
 - *id*: El identificador de la pregunta actual.

A.4.1.3 public void checkFile(AdminPreguntas b, MenuMapTipos tipos, MenuMapOpciones opcionesTipos

Método que lee los parámetros de entrada de la URL del generador y llama al método correspondiente de la base de datos para buscar el contenido de los ficheros que se corresponden con dichos parámetros. Posteriormente llama al lector de XML para rellenar el generador con las preguntas de esos ficheros.

- **Llamada por:** `generador.xhtml` (al cargar)
- **Parámetros de entrada:**
 - *b*: El estado del *bean* en el momento actual. De aquí se obtienen los campos rellenos por el usuario (autor, título, etc.)
 - *tipos*: mapa de tipos para identificar el tipo.
 - *opcionesTipos*: mapa de tipos para obtener todas las posibles opciones.

A.5 Búsqueda

A.5.1 FiltroBuscador

Bean que da soporte al buscador.

A.5.1.1 public String navigate()

Método que navega al generador con los nombres de los ficheros marcados en la URL.

- **Llamada por:** botón «Usar Ficheros»
- **Parámetro de salida:** cadena con la URL a la que navegar.

A.5.1.2 public void busqueda()

Función que llama a la clase `DBConnection` para hacer una búsqueda genérica.

- **Llamada por:** botón «Búsqueda»

A.5.1.3 public void borrar()

Función que borra de la lista de resultados los marcados.

- **Llamada por:** botón «Borrar»

A.5.1.4 public void eliminarBBDD()

Función que llama a la clase `DBConnection` para borrar las entradas de la tabla marcadas en el buscador. También borra de la lista dichos elementos.

- **Llamada por:** botón «Eliminar de la base de datos»

A.5.2 ResultTableItem

Esta clase es el elemento que compone la lista de resultados. Tiene dos cadenas de texto: una con el nombre del fichero buscado y otra con el tipo.

- **Utilizada por:** `FiltroBuscador` y `DBConnection`.

A.6 Comunes

A.6.1 CommonFunc

Contiene funciones que se pueden utilizar en varios sitios. La mayoría de ellas simplemente imprimen mensajes de fallo o éxito.

A.6.1.1 public String getFileFormat(String type)

Método que devuelve el formato de preguntas en función del tipo recibido.

A.6.1.2 public static String toString(Document doc)

Método que convierte un documento recibido de tipo XML en una cadena de texto con las marcas incluidas.

- **Parámetros de entrada:**
 - `doc`: el XML a transformar.
- **Parámetro de salida:** cadena con el contenido del XML.

A.6.2 DBConnection

Clase encargada de comunicarse con la base de datos. Cualquier uso de la base de datos que se haga deberá hacerse con una llamada a esta clase.

A.6.2.1 public static void addQuestion(String name, String docContent, String idioma, String etapa, String asignatura, String tipo)

Añade una pregunta a la base de datos

- **Llamada por:** `AdminPreguntas`
- **Parámetros de Entrada:** (cada uno de ellos se corresponde a una columna)
 - `name`: fichero
 - `docContent`: datos
 - `idioma`: idioma
 - `etapa`: etapa
 - `asignatura`: asignatura
 - `tipo`: tipo

A.6.2.2 public static void search(String[] etapa, String asignatura, String tema, String nombreFich, String autor, String[] idioma, String[] tipo, String[] individual, String palabra, List<ResultTableItem> resultList, int min, int max)

Realiza una búsqueda en la base de datos.

- **Llamada por:** `FiltroBuscador`
- **Parámetros de Entrada:**
 - etapa: lista de etapas posibles.
 - asignatura: columna asignatura.
 - tema: título del fichero.
 - nombreFich: nombre del fichero.
 - autor: autor del fichero.
 - idioma: lista de idiomas.
 - tipo: lista de tipos, pero solo los comunes.
 - tipo: lista de tipo, pero solo los individuales.
 - palabra: cadena de texto que esté contenida en el XML.
 - resultList: lista en la que se van a guardar los resultados devueltos por la base de datos.
 - min: mínima cantidad de preguntas.
 - max: máxima cantidad de preguntas.

A.6.2.4 public static String getFile(String nombreFich)

Devuelve el contenido de un fichero que se encuentre en la base de datos

- **Llamada por:** `AdminPreguntas`
- **Parámetros de Entrada:**
 - nombreFich: el nombre del fichero deseado.
- **Parámetros de Salida:** el contenido XML del fichero encontrado pasado a String.

A.6.2.4 public static boolean fileExists(String nombreFich)

Comprueba si el fichero pasado como parámetro existe en la base de datos. Devuelve *true* si existe y *false* en caso contrario.

- **Llamada por:** `AdminPreguntas`

A.6.2.5 public static String getType(String nombreFich)

Devuelve el tipo del fichero pasado como parámetro.

- **Llamada por:** `AdminPreguntas`

A.6.2.6 public static void delete(List<ResultTableItem> selectedResult)

Borra los ficheros pasados como parámetro de la base de datos.

- **Llamada por:** `FiltroBuscador`

