

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

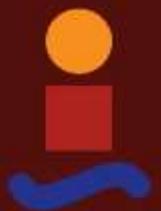
Aplicación web para la creación y modificación de  
ficheros de juegos para la Aplicación de Juegos  
Didácticos en el Aula (AJDA)

Autor: Carlos Ramos León

Tutor: Francisco Javier Muñoz Calle

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

Sevilla, 2018





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

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

Autor:

Carlos Ramos León

Tutor:

Francisco Javier Muñoz Calle

Profesor titular

Departamento de Ingeniería Telemática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Grado: Aplicación web para la creación y modificación de ficheros de juegos para la  
Aplicación de Juegos Didácticos en el Aula (AJDA)

Autor: Carlos Ramos León

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:

Sevilla, 2018

El Secretario del Tribunal

*A mi familia*

*A mis maestros*

*A mis amigos*



## Agradecimientos

---

Durante esta etapa de mi vida, muchas son las personas que han pasado por ella, tantos en los buenos como en los malos momentos y por eso me gustaría agradecerles todo el apoyo recibido para alcanzar mis metas, porque sin ellos, esto no hubiera sido posible.

En primer lugar, agradecer a mis padres Francisco y María Montserrat por todo el sacrificio y trabajo realizado para que tanto mi hermano como yo tuviésemos la oportunidad de formarnos y poder cumplir nuestros objetivos en la vida, sin ellos, nada de esto hubiese sido posible.

En segundo lugar, agradecer al resto de mi familia todo el apoyo recibido y por todas las experiencias vividas y compartidas, las cuáles me han llevado a ser como soy hoy en día.

Agradecer profundamente a mi pareja, Rosa, la cual me ha acompañado en esta etapa desde el principio y que, sin su apoyo, esto hubiese sido muy complicado. Gracias por acompañarme en los buenos y los malos momentos vividos y sobretodo, gracias por estar ahí en los malos momentos, que sin duda son los que más me han ayudado a salir adelante. Me siento muy orgulloso y afortunado de estar a tu lado y que cada día que pasa me haces sentir muy especial. Eres una persona indispensable en mi vida.

Agradecer a mis compañeros todo el esfuerzo realizado, esas batallas que a veces se ganaban y a veces se perdían, sin ellos esto hubiese sido aún más duro. Javi, Rafa, Lolo, Ana, Juanmi, Antonio, Salva. Me llevo de ellos en esta etapa a grandes personas, grandes amigos y únicas personas las cuales me siento afortunado por haberlas conocido.

Gracias también a mi tutor Francisco Javier, por darme la oportunidad de realizar este proyecto junto a él. Gracias por su implicación, sus conocimientos y su profesionalidad, he aprendido mucho de usted como profesor y como persona.

Por último y no por eso menos importante, quiero agradecer desde lo más profundo de mi corazón y dedicar este trabajo de fin de grado y título de Grado en Ingeniería de las Tecnologías de la Telecomunicación a mi madre, María Montserrat, porque ella ha sido principal responsable de que pudiera lograr este objetivo. Sin su fuerza y ánimos no hubiese sido capaz de lograrlo.

*Te quiero mamá.*

*Carlos Ramos León*

*Sevilla, 2018*



## Resumen

---

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

En este proyecto se ha desarrollado una aplicación web que es capaz de crear ficheros de juegos desde cero, leer y modificar ficheros de juegos procedentes de la aplicación de AJDA, mezclar tipos de juegos compatibles y exportar la información de los juegos a los formatos de txt, xml y pdf.

La principal finalidad de esta aplicación es liberar al usuario de tener que modificar a mano los ficheros en formato txt con los que trabaja la aplicación de AJDA y hacer que esto sea mucho más intuitivo a través del formulario de creación y modificación de preguntas, el cual te permite mezclar otros ficheros de preguntas compatibles, con la finalidad de agrupar todas las preguntas en un mismo tipo y que, una vez hecho esto, el usuario pueda tener estas preguntas en un formato pdf para por ejemplo sacar las preguntas de un examen a partir de un juego de la aplicación.



# Índice

<b>Agradecimientos</b>	<b>9</b>
<b>Resumen</b>	<b>11</b>
<b>Índice</b>	<b>13</b>
<b>Índice de Tablas</b>	<b>15</b>
<b>Índice de Figuras</b>	<b>17</b>
<b>1 Introducción</b>	<b>20</b>
1.1 <i>Motivación y objetivos</i>	20
1.2 <i>Descripción del problema</i>	21
1.3 <i>Antecedentes</i>	21
1.4 <i>Descripción de la solución</i>	23
1.4.1 <i>Introducción</i>	23
1.4.2 <i>Objetivos</i>	23
1.4.3 <i>Arquitectura</i>	24
1.5 <i>Composición de la memoria</i>	25
<b>2 Tecnologías utilizadas</b>	<b>27</b>
2.1 <i>JavaServer Faces (JSF)</i>	27
2.1.1 <i>¿Qué es JSF?</i>	27
2.1.2 <i>Ventajas que JSF nos ofrece para nuestro proyecto</i>	27
2.2 <i>PrimeFaces</i>	28
2.2.1 <i>Extensiones JSF</i>	28
2.2.2 <i>¿Por qué usamos PrimeFaces?</i>	28
2.2.3 <i>Uso en el proyecto</i>	29
2.3 <i>Managed Beans</i>	30
2.3.1 <i>Introducción</i>	30
2.3.2 <i>Uso en el Proyecto</i>	30
2.4 <i>AJAX (Asynchronous JavaScript And XML)</i>	31
2.4.1 <i>Definición</i>	31
2.4.2 <i>JSF Ajax (Ejecución y Render)</i>	33
2.5 <i>XML (Extensible Markup Language)</i>	35
2.5.1 <i>Definición</i>	35
2.5.2 <i>Uso en el Proyecto</i>	35
2.6 <i>IText</i>	35
2.6.1 <i>Definición</i>	35
2.6.2 <i>Uso en el Proyecto</i>	36
<b>3 Herramientas utilizadas</b>	<b>39</b>
3.1 <i>NetBeans</i>	39
3.2 <i>Apache Tomcat</i>	40
3.3 <i>Google Chrome</i>	41

3.4	<i>GitHub</i>	42
3.4.1	Git	42
3.4.2	Uso en el Proyecto	43
<b>4</b>	<b>Estructura y desarrollo de la aplicación web</b>	<b>44</b>
4.1	<i>Estructura de directorios</i>	44
4.1.1	Generador.xhtml	45
4.1.2	Tipos de preguntas	45
4.1.3	Paquetes de fuentes	45
4.2	<i>Página Principal</i>	47
4.3	<i>Acciones que el usuario puede realizar</i>	48
4.3.1	Creación de un nuevo fichero de juego	48
4.3.2	Importar preguntas desde un fichero existente (XML o TXT)	48
4.3.3	Exportar ficheros a formato PDF o TXT	50
<b>5</b>	<b>Plan de pruebas</b>	<b>52</b>
5.1	<i>Introducción</i>	52
5.2	<i>Pruebas</i>	52
5.2.1	Pruebas unitarias	52
5.2.2	Pruebas de integración	56
5.2.3	Pruebas de validación	57
<b>6</b>	<b>Conclusiones</b>	<b>59</b>
<b>7</b>	<b>Líneas de mejora</b>	<b>61</b>
	<b>Referencias y Bibliografía</b>	<b>63</b>

# ÍNDICE DE TABLAS

---

Tabla 1: PU-01	53
Tabla 2: PU-02	53
Tabla 3: PU-03	54
Tabla 4: PU-04	54
Tabla 5: PU-05	55
Tabla 6: PU-06	55
Tabla 7: PI-01	56
Tabla 8: PI-02	57
Tabla 9: PV-01	57



---

# ÍNDICE DE FIGURAS

---

Ilustración 1-1: Menú de selección de tipos AJDA	22
Ilustración 1-2: Generador de ficheros de preguntas AJDA	22
Ilustración 1-3: MVC JSF	24
Ilustración 2-1: Logo JSF	27
Ilustración 2-2: Ventajas PrimeFaces	29
Ilustración 2-3: Logo PrimeFaces	29
Ilustración 2-4: Ejemplo de Managed Bean usando anotaciones	30
Ilustración 2-5: Ejemplo de Managed Bean usando la config XML	30
Ilustración 2-6: Uso de la etiqueta @ViewScoped	31
Ilustración 2-7: Logo AJAX	31
Ilustración 2-8: Funcionamiento básico AJAX	32
Ilustración 2-9: Ejecución de componentes JSF	33
Ilustración 2-10: Renderizado	33
Ilustración 2-11: Ejemplo con h:inputText	34
Ilustración 2-12: Ejemplo con h:outputText	34
Ilustración 2-13: Logo XML	35
Ilustración 2-14: Función XML	35
Ilustración 2-15: Logo iText	36
Ilustración 2-16: Esquema iTEXT	36
Ilustración 3-1: Logo NetBeans	39
Ilustración 3-2: Interfaz gráfica de NetBeans.	40
Ilustración 3-3: Logo Apache	41
Ilustración 3-4: Logo Google Chrome	41
Ilustración 3-5: Depurador de Chrome	42
Ilustración 3-6: Logo GitHub	42
Ilustración 3-7: Logo Git	43
Ilustración 3-8: Uso de GitHub en el proyecto.	43
Ilustración 4-1: Estructura de directorios	44
Ilustración 4-2: Página principal generador de ficheros de Juegos	45
Ilustración 4-3: Tipos de preguntas	45
Ilustración 4-4: Menú principal aplicación	47
Ilustración 4-5: Creación de nuevo fichero de preguntas	48
Ilustración 4-6: Importar fichero de preguntas	49
Ilustración 4-7: Error al importar por formato de fichero	49

Ilustración 4-8: Error al importar por tipo de juego incompatible	50
Ilustración 4-9: Conversión a formato PDF o TXT	50



# 1 INTRODUCCIÓN

---

*"Los programas deben ser escritos para que los lean las personas, y sólo incidentalmente, para que lo ejecuten las máquinas".*

*Abelson and Sussman*

Cada vez es más común que en cualquier tipo de entidad (empresa, asociación, negocio, autónomo, institución...) necesite una solución informática ajustada a sus necesidades. Y cada vez es más habitual que se nombre una "Aplicación Web" como la respuesta a estas necesidades.

Entre los beneficios que las aplicaciones desarrolladas para la web tienen respecto a las aplicaciones de escritorio se encuentran:

- El trabajo a distancia se realiza con mayor facilidad.
- Para trabajar en la aplicación Web solo se necesita un equipo con un buen navegador Web y conexión a internet.
- Las aplicaciones Web no necesitan conocimientos previos de informática. Con una aplicación Web tendrá total disponibilidad en cuanto a hora y lugar, podrá trabajar en ella en cualquier momento y en cualquier lugar del mundo siempre que tenga conexión a internet.
- Las aplicaciones Web le permiten centralizar todas las áreas de trabajo.

## 1.1 Motivación y objetivos

Este proyecto pretende desarrollar una aplicación web para la manipulación de los ficheros de juegos que utiliza la aplicación del proyecyo AJDA. Con la finalidad de que el usuario de esta aplicación pueda crear, modificar y/o exportar cómodamente ficheros de juegos sin tener conocimiento de cómo la aplicación de AJDA los utiliza para la creación de los juegos didácticos en el aula.

Esta aplicación estará formada principalmente por una página web la cual se implementará con el lenguaje de programación JSF, basándonos en Primefaces, la biblioteca de componentes para Java Server Faces (JSF) que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web.

Por último, quiero destacar que, entre todas las razones que me han llevado a decidirme por este proyecto de programación web, la cual ha sido durante toda mi etapa en el grado la que más curiosidad me ha despertado, ha sido la posibilidad de poder aprender un lenguaje nuevo, como es JSF y adentrarme en él a través de su biblioteca de componentes de Primefaces. Ya que aquí, ví una oportunidad de mejora de mi carrera profesional y tenía que aprovecharla.

Una vez descritos las motivaciones y objetivos que me han llevado a relizar esto, se detallará a continuación el problema a resolver y la solución que se propone.

## 1.2 Descripción del problema

El principal escollo que tiene el uso de la aplicación de AJDA en el aula, es la necesidad, de los usuarios de dicha aplicación, de realizar un curso formativo, que principalmente está enfocado para aquellos usuarios inexpertos en el mundo de la informática, con la finalidad de poder utilizar en clase los ficheros de preguntas de juegos que se utilizan en esta aplicación.

Otro problema que se propone en este proyecto es que los ficheros de juegos que se corren en la aplicación están definidos en formato txt, y el propio usuario de la aplicación debe aprender cómo ésta interpreta cada línea de dicho fichero para la creación del juego sin que haya ningún error. Esto hace que la aplicación sea un poco deficiente en ese aspecto ya que este problema debería ser transparente para el usuario. El usuario no debería poder manipular esos ficheros txt, si no que se deberían manipular a través de la aplicación.

Atacando el problema mencionado anteriormente, nuestra aplicación busca que el usuario a través de la página pueda crear y manipular estos ficheros de juegos de manera más eficiente a como se hace actualmente en la aplicación de AJDA y, además, añadir la funcionalidad de poder mezclar diferentes tipos de juegos compatibles, que en la aplicación actual si se quería hacer, se tendrían que hacer cambiando a mano los ficheros txt.

En resumen, la idea de esta página es que, lo mencionado anteriormente, sea transparente para el usuario y no tenga que modificar manualmente ningún fichero txt, con la finalidad de que la aplicación de AJDA sea más intuitiva y con el objetivo final de no tener que realizar ningún curso formativo.

## 1.3 Antecedentes

Como principal antecedente a nuestro trabajo, tenemos la aplicación de AJDA:

### **Aplicación de Juegos Didácticos en el Aula**

Aplicación de juegos didácticos en el aula es un proyecto que pone al servicio de la comunidad educativa una amplia colección de juegos y materiales relacionados con ellos, con el objetivo de que sirvan como recursos didácticos de aplicación en el aula.

La parte que nos ocupa de esta aplicación es el generador de ficheros, el cuál es una página web para la creación y modificación de ficheros de juegos que utiliza la aplicación para generar el juego didáctico.

Contiene una gran variedad de juegos didácticos para cualquier nivel de enseñanza, se generan a través de dicha web, y se guardan en ficheros de texto.

Según la manera en la que introducimos la información, se cuentan con diferentes modalidades o tipos de juegos: preguntas con respuestas con opciones, preguntas con respuestas de campo de texto, preguntas con respuestas de cifras, etc..

En función del tipo de juego escogido, cuando se genere el fichero de preguntas, lo hará, como se ha dicho previamente, en formato txt, el cual no posee una estructura común entre diferentes tipos de juegos existentes, sino que todo lo contrario. Cada fichero de cada tipo de juego tiene una estructura diferente, la cual, se trata de forma distinta en la aplicación.



Ilustración 1-1: Menú de selección de tipos AJDA

The screenshot shows the AJDA 'Generador de ficheros de preguntas' interface in the question editor mode. It includes fields for 'Título' and 'Autor/a:', a 'Pregunta' counter set to 0, and a 'Formulario tipo 1' dropdown. The question text is entered in three lines: 'Enunciado línea 1', 'Enunciado línea 2', and 'Enunciado línea 3'. Below this are four multiple-choice options: 'A)', 'B)', 'C)', and 'D)', each with an input field. A 'Solución' dropdown is also present. At the bottom, there are several buttons: 'Rescindir (borrar todo)', 'Ir al panel de contenidos', 'Limpiar', 'Cargar de un fichero', and 'Cargar preguntas al final'. The 'Cargar de un fichero' and 'Cargar preguntas al final' buttons are highlighted in yellow and pink respectively.

Ilustración 1-2: Generador de ficheros de preguntas AJDA

## 1.4 Descripción de la solución

En este apartado explicaremos el funcionamiento, características principales y arquitectura que sigue este Proyecto.

### 1.4.1 Introducción

El objetivo principal de este proyecto es facilitar al usuario de la aplicación AJDA la creación y modificación de ficheros de preguntas para su uso en juegos de la aplicación en clase o para la elaboración de preguntas de exámenes a partir de las preguntas del juego.

A través de la aplicación, el usuario podrá crear un nuevo fichero de juego, importar tanto en formato xml o txt ficheros ya creados, reutilizar ficheros de distinto tipo al que se quiere crear que sean compatibles, así como exportar a formato pdf las preguntas de un fichero de juego de la aplicación AJDA.

### 1.4.2 Objetivos

Cuando iniciamos el navegador y accedemos a la aplicación web, nos encontramos una página web con las siguientes alternativas:

- En primer lugar, un selector de tipo de juego con el que se va a trabajar.
- Una vez elegido el tipo de juego podremos:
  - ✓ Crear desde cero un nuevo fichero de preguntas.
  - ✓ Importar un fichero de preguntas ya generado.
  - ✓ Exportar un fichero de preguntas ya generado en formato xml para su conversión a pdf o nuevamente a txt para usarlo en la aplicación AJDA.
- Cuando tenemos las preguntas del fichero de juego importadas a la página web, podremos realizar lo siguiente:
  - ✓ Modificar el contenido de la pregunta.
  - ✓ Añadir/Eliminar respuestas.
  - ✓ Añadir/Eliminar preguntas.
  - ✓ Minimizar o expandir preguntas.

### 1.4.3 Arquitectura

JSF utiliza el framework MVC (Modelo-Vista-Controlador) para gestionar las aplicaciones web, así que hemos seguido una arquitectura basada en el modelo vista controlador (MVC), que es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

La correspondencia entre cada parte M, V, y C y cada uno de los elementos de la aplicación es la siguiente:

- **Modelo:** Se entiende a éste como todo el conjunto de datos y toda la lógica de negocio que procesa esos datos. En JSF el Modelo se implementa en las clases managed bean, también conocidos como backing beans.
- **Vista:** Constituye también la vista de la aplicación. Aquí se sitúan los elementos para la entrada y salida de datos de usuario implementados con la tecnología Facelets de JSF y con el lenguaje EL de expresiones.
- **Controlador:** Lo proporciona JSF a través de la clase FacesServlet. Gestiona las acciones del usuario sobre la Vista y muestra la información adecuada en la misma.



Ilustración 1-3: MVC JSF

Esta aplicación web se construye como una pantalla con la que va interactuando el usuario. Esta pantalla contiene textos, botones, imágenes, tablas y elementos de selección que el usuario modifica.

Todos estos elementos están agrupados en formularios HTML, que es la forma que tienen las páginas web de mandar los datos introducidos por el usuario.

El controlador JSF asocia a la pantalla clases java que recoge y trata la información del usuario, así como responder a las acciones del usuario.

JSF nos resuelve de forma automática tareas como:

- Mostrar datos al usuario en cajas de texto y tablas.
- Recoger los datos introducidos por el usuario en los campos del formulario.
- Realizar validaciones de los datos introducidos por el usuario.
- Rellenar campos, listas, combos y otros elementos a medida que el usuario va interactuando con la pantalla.
- Controlar los eventos que ocurren en los controles (pulsaciones de teclas, botones y movimientos el ratón).

Esta aplicación se apoya en un *HashMap* para el almacenamiento de los datos acerca de los ficheros de juegos, tipos, características, etc. Con la intención de emular una base de datos, que aún no está desarrollada, para que en el futuro, cuando se acoplen todos los proyectos involucrados en la mejora de la aplicación AJDA, esta información se extraiga de dicha base de datos.

## 1.5 Composición de la memoria

En este apartado se resume los puntos tratados en esta memoria, con la intención de tener una visión global de la organización del documento:

1. Introducción: En este punto abordaremos temas como la motivación y objetivos, así como la descripción del problema y de la solución.
2. Tecnologías utilizadas: En esta sección hablaremos de las tecnologías utilizadas para la realización de este Proyecto y también se explicará por qué se usan.
3. Herramientas utilizadas: Este punto se centra en la descripción de las herramientas utilizadas para implementar este Proyecto y el motivo de su utilización.
4. Estructura y desarrollo de la aplicación web: Aquí entraremos más en detalle de la descripción de la solución de la cual hablamos en la introducción. Explicaremos la estructura de directorios que tenemos, detallando la funcionalidad de cada fichero y explicando cómo sería la interfaz que tendría el usuario y cómo usarla.
5. Plan de pruebas: Consta de las pruebas que se le ha hecho a la aplicación con la finalidad de comprobar que se cumplen los objetivos del Proyecto.
6. Conclusiones: Se incluye una conclusión final tras la realización del proyecto y una valoración del mismo a nivel general y personal.
7. Líneas de mejora: Se mencionarán algunas tecnologías para mejorar este servicio y un razonamiento de las mismas.
8. Referencias y Bibliografía



# 2 TECNOLOGÍAS UTILIZADAS

---

*“El gran motor del cambio – la tecnología.”*

*-Alvin Toffler*

En este apartado se mostrarán las tecnologías utilizadas para la realización de esta aplicación web. Se realizará una explicación detallada y razonada de cada tecnología utilizada y su finalidad. En este apartado sólo se hará una descripción detallada sobre las tecnologías utilizadas, será en el próximo capítulo donde se entrará en detalle sobre la forma en la que se han implementado dichas tecnologías en nuestra aplicación web.

## 2.1 JavaServer Faces (JSF)

### 2.1.1 ¿Qué es JSF?

JSF es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como lenguaje basado en XML para la interfaz de usuario.



Ilustración 2-1: Logo JSF

### 2.1.2 Ventajas que JSF nos ofrece para nuestro proyecto

JSF nos ofrece una serie de ventajas:

- ✓ El código JSF con el que creamos las vistas (etiquetas jsp) es muy parecido al HTML estándar. Lo pueden utilizar fácilmente desarrolladores y diseñadores web.
- ✓ JSF se integra dentro de la página JSP y se encarga de la recogida y generación de los valores de los elementos de la página.
- ✓ JSF resuelve validaciones, conversiones, mensajes de error e internacionalización (i18n).

- ✓ JSF permite introducir javascript en la página, para acelerar la respuesta de la interfaz en el cliente (navegador del usuario).
- ✓ JSF es extensible, por lo que se pueden desarrollar nuevos componentes a medida, También se puede modificar el comportamiento del framework mediante APIs que controlan su funcionamiento.

Desde el punto de vista técnico podemos destacar lo siguiente:

- JSF forma parte del estándar J2EE, mientras que otras tecnologías para creación de vistas de las aplicaciones no lo forman, como por ejemplo Struts.
- JSF dispone de varias implementaciones diferentes, incluyendo un conjunto de etiquetas y APIs estándar que forman el núcleo del framework.
- El desarrollo de JSF está realmente empezando. Las nuevas versiones del framework recogen la funcionalidad de versiones anteriores siendo su compatibilidad muy alta, de manera que el mantenimiento de aplicaciones no se ve penalizado por el cambio de versiones.

## 2.2 PrimeFaces

### 2.2.1 Extensiones JSF

JSF nos ofrece una gran variedad de extensiones:

- RichFaces: es un marco de componentes de interfaz de usuario avanzado para integrar fácilmente las capacidades de Ajax en las aplicaciones empresariales utilizando JSF.
- ICEfaces: ICEfaces es un marco de desarrollo de aplicaciones de Internet enriquecidas (RIA) de código abierto para Java EE. ICEfaces funciona en plataformas que van desde computadoras de escritorio a teléfonos inteligentes y desde Apple a Android. Mejora la eficiencia del desarrollador al tiempo que reduce el tiempo de comercialización y los costos operativos.
- jQuery4jsf: es una biblioteca de código abierto para Java Server Faces. El propósito de esta biblioteca es poner a disposición del desarrollador un conjunto de componentes que contengan tecnología RIA, y permitir que el desarrollador realice rápidamente sus aplicaciones.
- PrimeFaces: es una biblioteca de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web.

### 2.2.2 ¿Por qué usamos PrimeFaces?

PrimeFaces es una librería de componentes de código abierto para JSF 2.0 con más de 100 componentes. PrimeFaces es mucho mejor que muchas otras bibliotecas de componentes JSF, debido a varias razones:

- ❖ Un amplio conjunto de componentes de Interfaz de usuario (DataTable, AutoComplete, HtmlEditor, Charts etc).
- ❖ No se requiere extra configuración, ni son necesarias dependencias.
- ❖ Ajax incorporado.
- ❖ Conjunto de más de 25 temas integrados
- ❖ Excelente documentación con ejemplos de código.
- ❖ Es compatible con otras librerías de componentes, como JBoss RichFaces.

- ❖ Uso de javascript no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque <script>).
- ❖ Es un proyecto open source, activo y bastante estable entre versiones.



Ilustración 2-2: Ventajas PrimeFaces



Ilustración 2-3: Logo PrimeFaces

### 2.2.3 Uso en el proyecto

En este proyecto hemos hecho uso de esta tecnología para la elaboración de la página principal del Generador de ficheros ya que cumple con la gran mayoría de requisitos que necesitamos, es estable, gran calidad de componentes ofrece skins, soporte en JSF 2.0, además de contar con una excelente documentación.

Implementar un componente de PrimeFaces resulta sencillo cuando disponemos de una gran variedad de demos como es el caso de PrimeFaces, donde por cada componente, existen diversos ejemplos que incluyen su código xhtml y su Bean de Java.

## 2.3 Managed Beans

### 2.3.1 Introducción

Managed Bean es una clase regular de Java Bean registrada con JSF, es decir, es un bean Java administrado por el marco JSF. El bean administrado contiene los métodos getter y setter, la lógica empresarial o incluso un bean de respaldo (un bean contiene todo el valor del formulario HTML).

Los beans administrados funcionan como modelo para el componente UI. Managed Bean se puede acceder desde la página JSF. Este enfoque mantiene los beans y su registro en un lugar, por lo que es más fácil de administrar.

```
<managed-bean>
  <managed-bean-name>helloWorld</managed-bean-name>
  <managed-bean-class>com.tutorialspoint.test.HelloWorld</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>

<managed-bean>
  <managed-bean-name>message</managed-bean-name>
  <managed-bean-class>com.tutorialspoint.test.Message</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
```

Ilustración 2-5: Ejemplo de Managed Bean usando la config XML

```
@ManagedBean(name = "helloWorld", eager = true)
@RequestScoped
public class HelloWorld {
    @ManagedProperty(value = "#{message}")
    private Message message;
    ...
}
```

Ilustración 2-4: Ejemplo de Managed Bean usando anotaciones

### 2.3.2 Uso en el Proyecto

En este proyecto se ha hecho uso de la versión de JSF 2.0, es decir, hemos utilizado las anotaciones. Las etiquetas más utilizadas en este proyecto son:

- `@ManagedBean`: marca el bean para ser un managed bean con el nombre específico en el atributo nombre. En el caso de no especificar el nombre en la anotación, el nombre del managed bean será por defecto el nombre de la clase.
- `@RequestScoped`: establece el ámbito en el que se sitúa el bean. Si no se establece, por defecto el ámbito será un requestscope. Los posibles ámbitos son `@NoneScoped`, `@RequestScoped`, `@ViewScoped`, `@SessionScoped`, `@ApplicationScoped`, y `@CustomScope`.

Como de momento no se dispone de base de datos, se ha utilizado mucho la etiqueta `ViewScoped`, que permite almacenar los objetos mientras nos mantengamos en la misma vista. De esta forma podremos, de una forma relativamente sencilla, eliminar muchos de los datos que antiguamente almacenábamos en la sesión y mejorar la escalabilidad de nuestra aplicación.

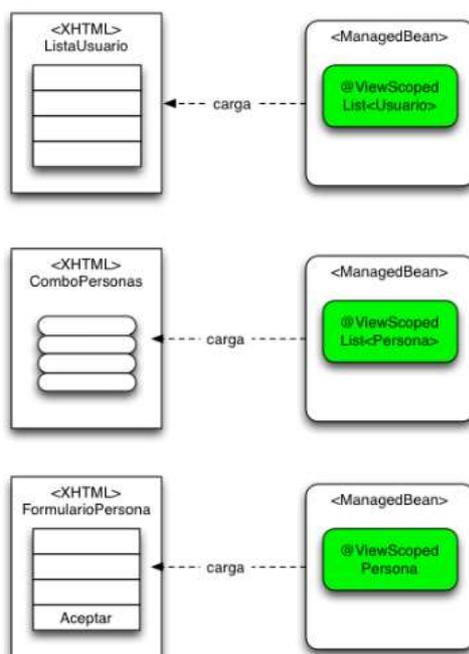


Ilustración 2-6: Uso de la etiqueta `@ViewScoped`

## 2.4 AJAX (Asynchronous JavaScript And XML)

### 2.4.1 Definición

AJAX es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.



Ilustración 2-7: Logo AJAX

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. No es una tecnología en sí mismo si no que es una combinación de cuatro tecnologías ya existentes:

- ✓ XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- ✓ Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- ✓ El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios. PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico también utilizado en el método Ajax.
- ✓ XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

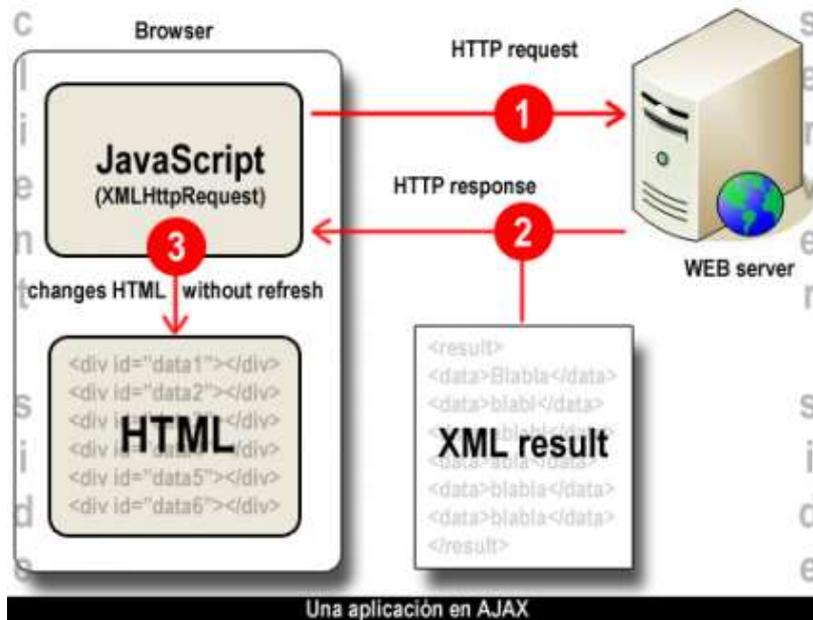


Ilustración 2-8: Funcionamiento básico AJAX

## 2.4.2 JSF Ajax (Ejecución y Render)

Para entender como funciona AJAX en JSF hay que entender que aunque el ciclo de vida de JSF es muy complejo se puede dividir en dos grupos fundamentales. A estos grupos se les denomina Ejecución y Renderizado.

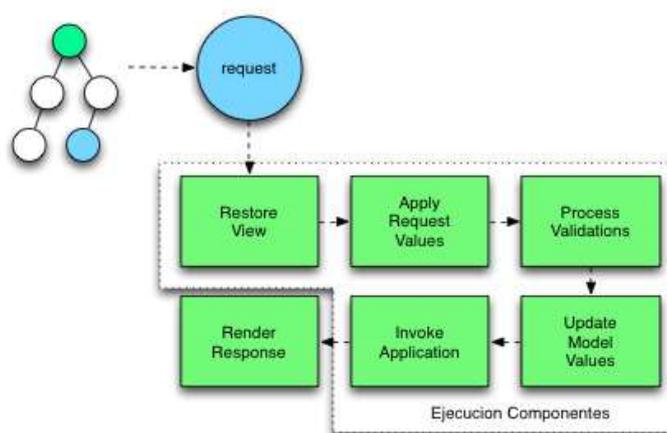


Ilustración 2-9: Ejecución de componentes JSF

La parte de ejecución hace referencia, como su nombre indica, a todo el ciclo de vida que está ligado a la propia ejecución de los componentes, dejando de lado la última fase, la fase de renderizado. En cambio, la segunda fase es opuesta a la primera solo se encarga del renderizado.

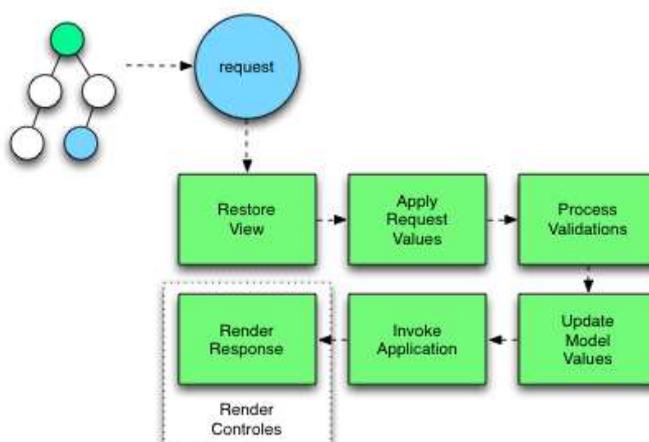
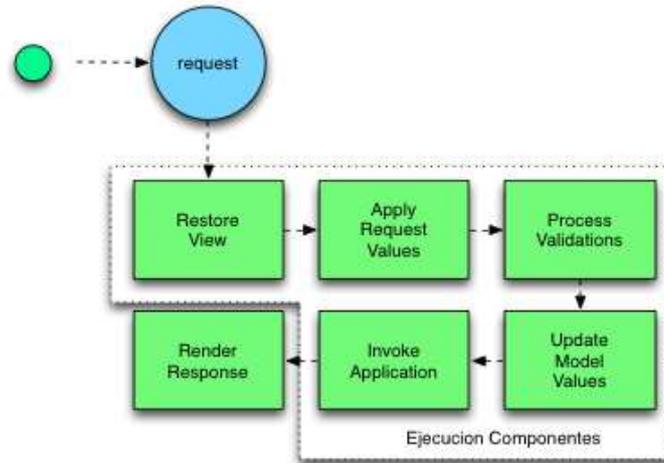
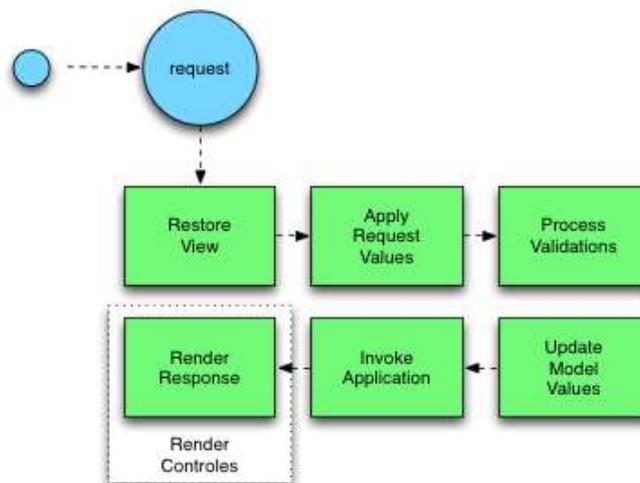


Ilustración 2-10: Renderizado

JSF y AJAX nos permiten aplicar estas fases a controles diferentes y obtener un resultado. Por ejemplo, podemos hacer que el control de `h:inputText` pase por todas las fases de ejecución.

Ilustración 2-11: Ejemplo con `h:inputText`

Pero que no le haga falta pasar por la fase de render ya que esta renderizado correctamente. Ahora podemos hacer que el control `h:outputText` funcione de la forma contraria y no pase por ninguna fase salvo la ultima.

Ilustración 2-12: Ejemplo con `h:outputText`

## 2.5 XML (Extensible Markup Language)

### 2.5.1 Definición

Es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.



Ilustración 2-13: Logo XML

### 2.5.2 Uso en el Proyecto

El uso de este lenguaje a lo largo de este Proyecto es muy importante, ya que a partir de este se va a poder sacar los diferentes formatos de los ficheros de juegos de la aplicación. Actualmente, la aplicación crea archivos de juego en formato XML o los convierte desde txt a XML (Si se importan en este formato). Una vez tenemos el fichero en formato XML, se podrá subir a la aplicación con el fin de convertirlo a otros formatos como PDF o de nuevo a txt.



Ilustración 2-14: Función XML

## 2.6 iText

### 2.6.1 Definición

iText es una biblioteca Open Source para crear y manipular archivos PDF, RTF, y HTML en Java. El mismo documento puede ser exportado en múltiples formatos, o múltiples instancias del mismo formato. Los datos pueden ser escritos a un fichero o, por ejemplo, desde un servlet a un navegador web.

El soporte de PDF de iText es bastante extensivo. Este soporta firmas basadas en PKI de PDF, cifrado de 40-bit y 128-bit, corrección de colores, PDF/X, gestión de colores por perfiles ICC, y es anfitriona de otras características.



Ilustración 2-15: Logo iText

La ventaja que ofrece iText sobre otras librerías de generación de documentos PDF que existen para la plataforma java, es que la clase que genera los PDF ya esta compilada, no hay que hacer una transformación xslt cada vez que se genera el documento, con lo que la generación es mucho más rápida y se consumen muchos menos recursos del sistema.

## 2.6.2 Uso en el Proyecto

En este Proyecto se hará uso de esta tecnología para exportar los ficheros de juegos generados por la aplicación en formato XML. La aplicación recogerá todas las preguntas que tiene dicho fichero, con sus respectivas respuestas y soluciones, y lo exporta a un formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware (PDF). Con la finalidad de que el usuario de la aplicación disponga de las preeguntas y respuestas del juego en un formato más legible para las presonas y de esta forma poder utilizar dicho contenido para la elaboración de un examen en papel por ejemplo.

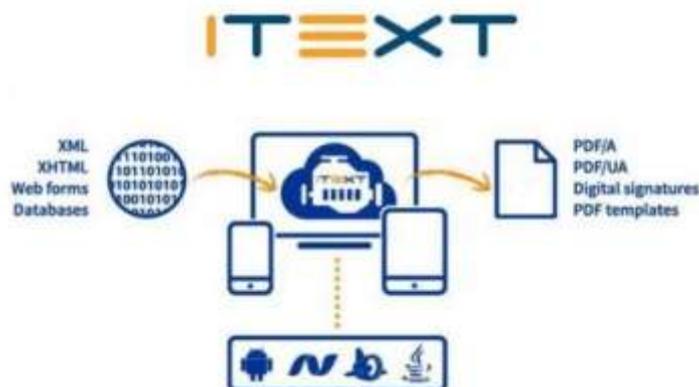


Ilustración 2-16: Esquema iTEXT





# 3 HERRAMIENTAS UTILIZADAS

---

*“Una de las mayores y más simples herramientas para aprender más es hacer cada día más.”*

*- Washington Irving-*

A lo largo de esta sección describiremos las herramientas utilizadas para la elaboración del proyecto. Es importante saber que las herramientas para el desarrollo de software desempeñan un importante papel en el desarrollo de aplicaciones, con lo cual es muy importante elegir bien las herramientas que se van a utilizar para conseguir los hitos que se proponen. Pero más importante es aún tener un conocimiento previo sobre ellas y su funcionalidad, si no, de nada sirve el uso de las mismas.

## 3.1 NetBeans

NetBeans es un Entorno de Desarrollo Integrado (IDE) para la creación de aplicaciones en Java para escritorio, móvil o web. También permite desarrollarlas en HTML5 (HTML, JavaScript y CSS) y dispone de herramientas para programadores en PHP, C y C++. NetBeans es un IDE desarrollado en Java y de código abierto.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.



Ilustración 3-1: Logo NetBeans

La plataforma ofrece servicios reusables comunes para las aplicaciones de escritorio, permitiendo a los desarrolladores centrarse en la lógica de sus aplicaciones. Algunas de las características de la aplicación son:

- Gestión de la interfaz de usuario (menús y barras de herramientas)
- Gestión de configuración de usuario

- Gestión de almacenamiento (guardar o cargar algún tipo de dato)
- Gestión de ventana
- Marco Asistente (soporta diálogos paso a paso)
- Librería visual de Netbeans
- Herramientas de desarrollo integrado

En este proyecto se ha elegido este entorno de desarrollo para la realización completa del proyecto, ya que NetBeans ofrece un fuerte apoyo al desarrollo de aplicaciones basadas en MVC en Java. Servlet desarrollo / JSP, y es bastante simple en comparación con Eclipse, por ejemplo, especialmente en el campo de la implementación y depuración.

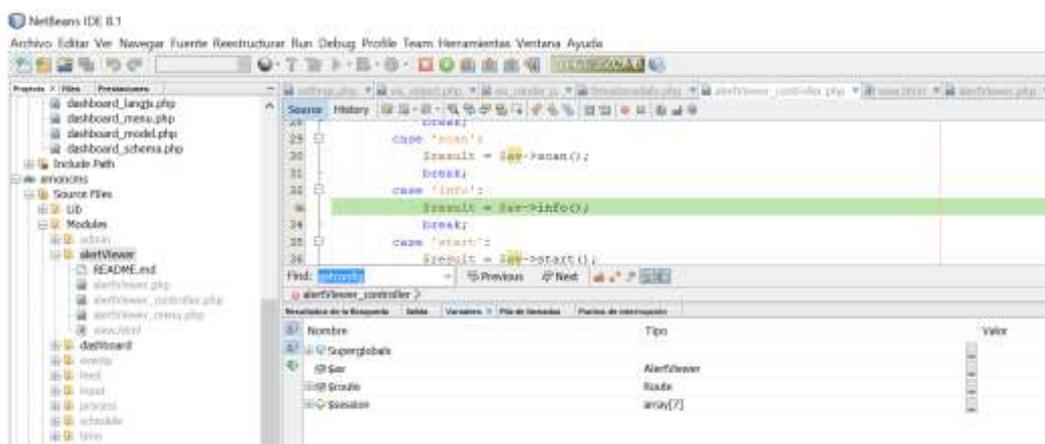


Ilustración 3-2: Interfaz gráfica de NetBeans.

## 3.2 Apache Tomcat

Apache Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation.

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

Tomcat posee más tiempo en el campo y es un contenedor web con soporte de servlets y JSP, es más ligero, tiene menos complejidad y uso de recursos así como de modularidad. Es por estos motivos, por los que he elegido utilizar este servidor en lugar de otros como GlassFish.

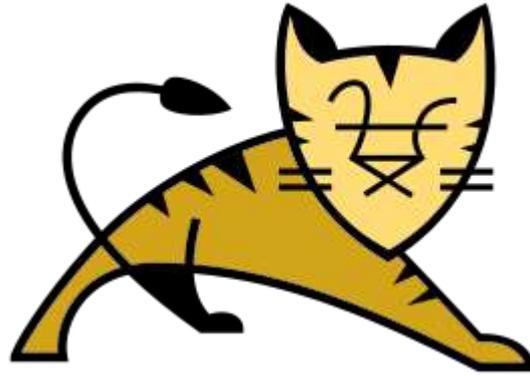


Ilustración 3-3: Logo Apache

### 3.3 Google Chrome

Google Chrome es un navegador web de código abierto desarrollado por Google. Este navegador está constituido por componentes de otros navegadores de código abierto como WebKit y Mozilla. El objetivo de Google Chrome es lograr un navegador con una interfaz sencilla, y una muy buena estabilidad, velocidad y seguridad.



Ilustración 3-4: Logo Google Chrome

Se ha utilizado esta herramienta para la interacción del usuario con la aplicación web junto con las funciones que se ofrece. Otro de los motivos por los cuales se ha elegido este navegador es para el uso de la herramienta de depuración de código del cliente, ya que JSF y todos los frameworks de componentes que complementan JSF integran en mayor o menor medida javascript mediante acciones o eventos. JSF y Javascript se complementan perfectamente y nos permiten entre otras cosas, aumentar la capacidad AJAX para nuestra aplicación y realizar llamadas asincronas a servicios mediante REST, acción que mejorará notablemente el rendimiento de nuestra aplicación.

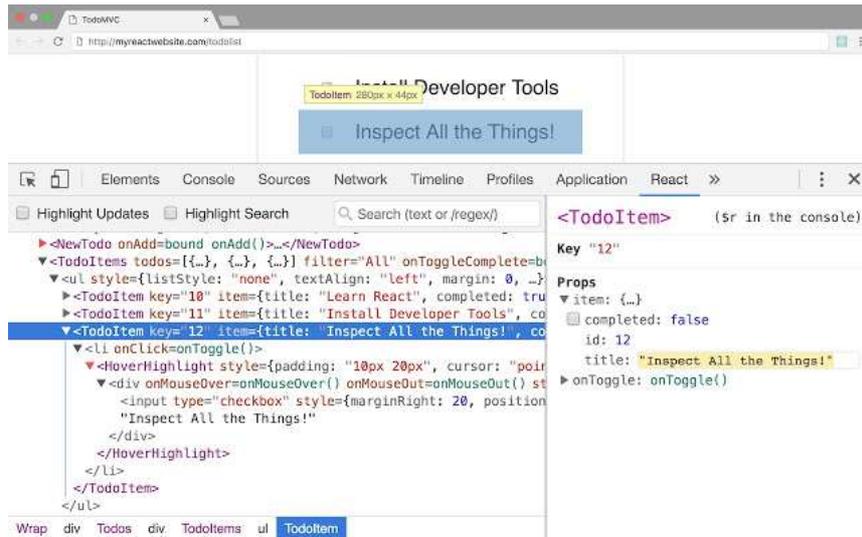


Ilustración 3-5: Depurador de Chrome

## 3.4 GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código Fuente. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.



Ilustración 3-6: Logo GitHub

### 3.4.1 Git

Git, es un software de control de versiones diseñado por Linus Torvalds. El control de versiones es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo, es decir, la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración.

Git fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones

cuando éstas tienen un gran número de archivos de código fuente, es decir Git nos proporciona las herramientas para desarrollar un trabajo en equipo (software o página que implique código el cual necesitemos hacerlo con un grupo de personas) de manera inteligente y rápida.



Ilustración 3-7: Logo Git

### 3.4.2 Uso en el Proyecto

Todos sabemos que las copias de seguridad de nuestros datos son importantes, tanto si son fotografías, ficheros o código de programación. Las alternativas siempre son las mismas: la nube usando Dropbox o Drive, en local usando discos duros aparte de los que empleamos para nuestro uso diario, pero en el caso del código tenemos una alternativa mucho mejor: los repositorios Git. Esta clase de repositorios son una copia local del código generado con una característica muy importante, y es que podemos hacer varias versiones para poder recular si nos hemos equivocado y nuestra aplicación ya no funciona, o para trabajar en funcionalidades nuevas sin necesidad de modificar la versión funcional y así no romper el proyecto.

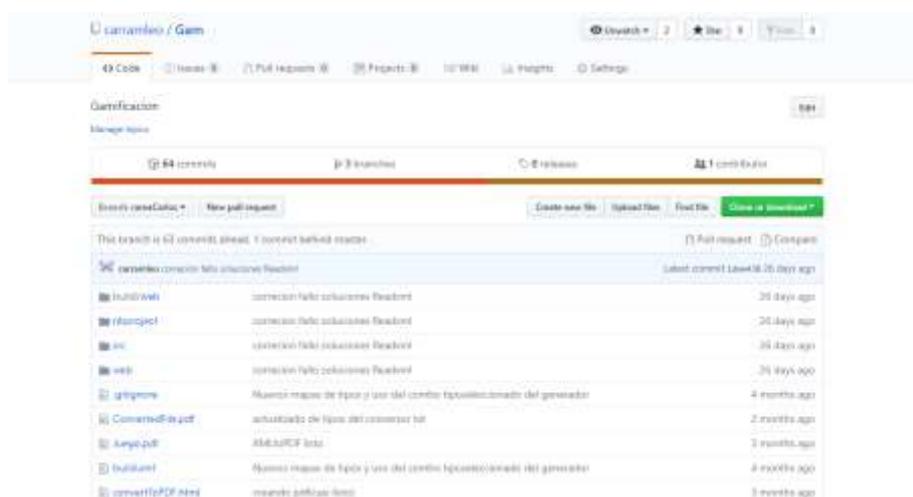


Ilustración 3-8: Uso de GitHub en el proyecto.

# 4 ESTRUCTURA Y DESARROLLO DE LA APLICACIÓN WEB

*“El éxito no se logra sólo con cualidades especiales.  
Es sobre todo un trabajo de constancia, de método y de organización.”*  
J.P. Sergent

**E**n este apartado se mostrará una descripción detallada de la estructura de la aplicación web y el propósito de cada uno de los elementos que la conforman.

## 4.1 Estructura de directorios

La estructura de directorios es la siguiente:

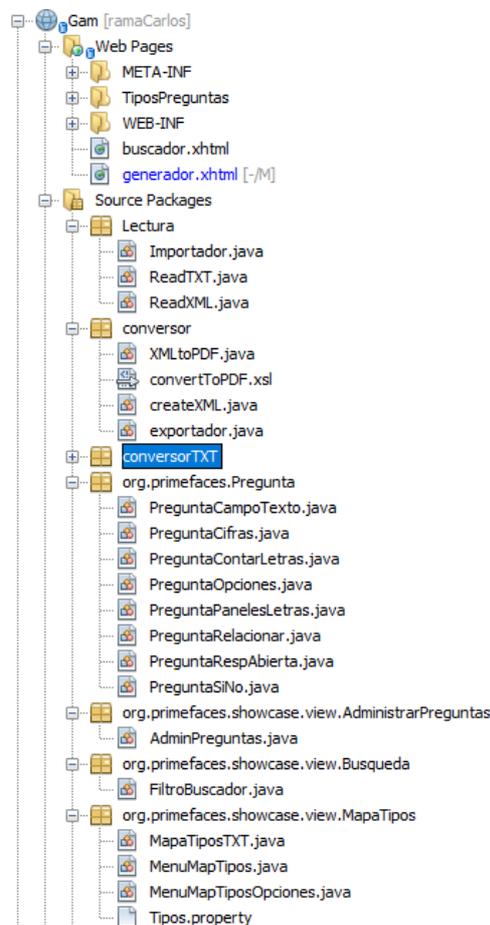


Ilustración 4-1: Estructura de directorios

### 4.1.1 Generador.xhtml

Es la página web principal de este Proyecto, la encargada de mostrar al usuario todas las funciones disponibles para él en el generador de ficheros de juegos. En ella se podrán crear ficheros de preguntas nuevos, importar ficheros de preguntas ya creados o convertir ficheros de preguntas creados a otros formatos.



Ilustración 4-2: Página principal generador de ficheros de Juegos

### 4.1.2 Tipos de preguntas

Este directorio contiene los ficheros XHTML con los diferentes tipos de pregunta que se pueden crear. En cada fichero se tiene un panel de pregunta distinto en función del tipo elegido para la creación del fichero de juego ya que cada tipo de juego requiere de una recopilación de datos distinta.

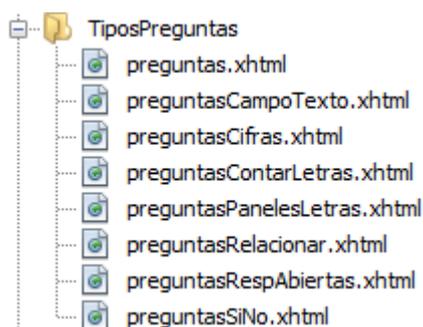


Ilustración 4-3: Tipos de preguntas

### 4.1.3 Paquetes de fuentes

#### 4.1.3.1 Lectura

En este paquete se tendrán todos los ficheros correspondientes a la importación de ficheros de preguntas ya generados, con la finalidad de modificarlos en la aplicación o convertirlos a formato XML, el cual es el formato general con el que trabaja la aplicación.

Los formatos de entrada disponibles para los ficheros a importar son XML o TXT.

- `Importador.java`: Esta clase se encargará de leer el fichero importado a la aplicación e identificar si se

trata de un fichero en formato compatible (XML o TXT) y en función de ello llamar a la clase de lectura de formato XML o TXT.

- `ReadTXT.java`: Esta clase es la encargada de leer los ficheros en formato TXT que usa la aplicación de AJDA, los cuáles hay que leerlos línea a línea, y los importa a la aplicación web, si el tipo a importar es un tipo compatible con el que se ha seleccionado en la aplicación, creando los respectivos paneles de preguntas del tipo seleccionado previamente, con la correspondiente información de cada uno.
- `ReadXML.java`: Esta clase es la encargada de leer los ficheros en formato XML que usa la aplicación de AJDA, los cuáles hay que leerlos línea a línea, y los importa a la aplicación web, si el tipo a importar es un tipo compatible con el que se ha seleccionado en la aplicación, creando los respectivos paneles de preguntas del tipo seleccionado previamente, con la correspondiente información de cada uno.

#### 4.1.3.2 Conversor

En este paquete se tendrán los ficheros encargados de convertir a formato XML las preguntas creadas y/o modificadas de la aplicación web, así como también los ficheros encargados de convertir los ficheros XML, previamente importados, a formato PDF.

- `Exportador.java`: Es la clase encargada de leer el formato de entrada del fichero a convertir, para en base a esta información llamar a la clase que lee en formato XML o en formato TXT.
- `XMLtoPDF.java`: Este fichero es el encargado de leer un fichero XML de entrada y haciendo uso de `iText` convertir a formato PDF. Previamente se convierta a HTML donde se le da formato al fichero a través de un fichero XSL.
- `createXML`: Esta clase es la encargada de generar un script en formato XML a partir de los paneles de preguntas de la aplicación.

#### 4.1.3.3 ConversorTXT

Este paquete contiene los fuentes encargados de leer los ficheros de juegos en formato TXT y a partir de ellos, generar los paneles de preguntas en la aplicación web.

- `XMLtoTXT.java`: Esta clase es la encargada de leer el fichero importado en formato XML para realizar la conversión a formato TXT que usa la aplicación de AJDA.
- `FichTipo1.java`: Es la clase encargada de dar formato en TXT a los ficheros importados XML de tipo test.

#### 4.1.3.4 `Org.primefaces.Pregunta`

Paquete que contiene las clases java que definen los tipos de preguntas existentes en el juego.

- `PreguntaCampoTexto.java`: Clase que define el tipo de pregunta de campo de texto.
- `PreguntaCifras.java`: Clase que define el tipo de pregunta de cifras.
- `PreguntaContarLetras.java`: Clase que define el tipo de pregunta de contar letras.
- `PreguntaOpciones.java`: Clase que define el tipo de pregunta de test.
- `PreguntaPanelesLetras.java`: Clase que define el tipo de pregunta de Paneles de letras.
- `PreguntaRelacionar.java`: Clase que define el tipo de pregunta de relacionar columnas.
- `PreguntaRespAbierta.java`: Clase que define el tipo de pregunta de respuesta abiertas.
- `PreguntaSiNo.java`: Clase que define el tipo de pregunta de respuesta de dos opciones (Si o No).

#### 4.1.3.5 org.primefaces.showcase.view.AdministrarPreguntas

Paquete que contiene la clase con el bean principal de la página web.

- AdminPreguntas.java: Clase que se encargada de la gestión de las preguntas de la aplicación web. Esta clase contendrá los métodos necesarios para añadir, borrar y modificar pregunta y respuestas.

#### 4.1.3.6 org.primefaces.showcase.view.MapaTipos

Este paquete contiene los HashMap con los datos de cada tipo de juego, con la finalidad de emular una base de datos.

- MapaTipos.TXT: esta clase recoge el HashMap con las propiedades de los tipos de fichero tratados en la lectura de ficheros de juegos de tipo TXT por la aplicación.
- MenuMapTipos.java: Esta clase contiene el HashMap agrupando todos los tipos de juegos en categorías, como por ejemplo respuestas con opciones, respuestas de respuesta abierta, etc.
- MenuMapTiposOpciones.java: Esta clase contiene el HashMap en el que se especifican las propiedades que tiene cada tipo de juego existente.

## 4.2 Página Principal

La interfaz de usuario es el medio con el que el usuario podrá interactuar con el generador de ficheros de juegos.

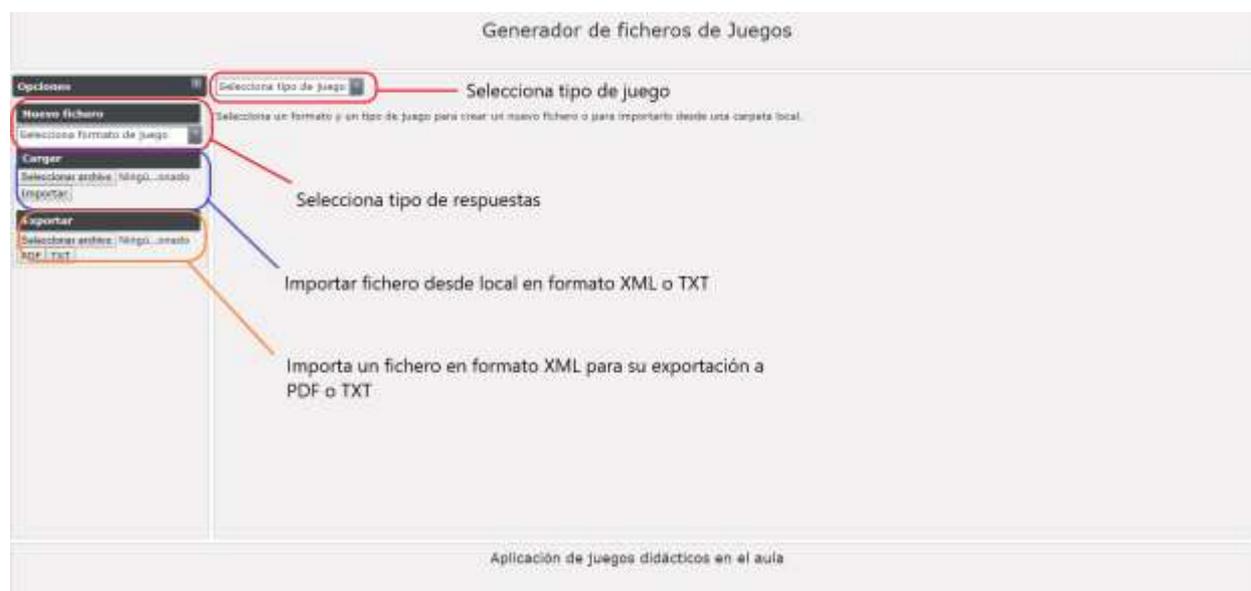


Ilustración 4-4: Menú principal aplicación

## 4.3 Acciones que el usuario puede realizar

### 4.3.1 Creación de un nuevo fichero de juego

Una de las acciones que puede realizar el usuario de la aplicación es la de generar un nuevo fichero de preguntas de juego desde cero. Para ello es necesario seleccionar tipo de formato de respuestas y posteriormente el tipo de juego.

Una vez seleccionado esto, se pueden ir añadiendo tantas preguntas como se quiera. Cuando les tenemos todas, sólo nos falta pulsar el botón de generar fichero y se generará un nuevo fichero de preguntas de juego en formato XML.



Ilustración 4-5: Creación de nuevo fichero de preguntas

### 4.3.2 Importar preguntas desde un fichero existente (XML o TXT)

Con esta acción podremos importar a la aplicación tanto ficheros de formato XML como de formato TXT con la finalidad de modificar su contenido, mezclar diferentes tipos de juegos compatibles, cambiar el número de preguntas, respuestas, etc. Para ello, se realiza un click en el botón de seleccionar archivo de el menú de Cargar, seleccionamos el archivo que queremos subir a la aplicación desde nuestro equipo y clickamos en el botón de importar. La propia aplicación se encargará de comprobar que el fichero que se quiere importar es del tipo permitido (TXT o XML), además de comprobar si el tipo de fichero de preguntas a importar es compatible con el que se ha seleccionado en la aplicación.



Ilustración 4-6: Importar fichero de preguntas



Ilustración 4-7: Error al importar por formato de fichero



Ilustración 4-8: Error al importar por tipo de juego incompatible

### 4.3.3 Exportar ficheros a formato PDF o TXT

Una acción muy útil que puede realizar el usuario de la aplicación es la posibilidad de convertir ficheros de juegos procedentes de la aplicación AJDA (en formato TXT) o ficheros generados por esta aplicación (en formato XML) a formatos como PDF o de nuevo a TXT, con la finalidad de importarlos de nuevo a la aplicación AJDA una vez modificados.



Ilustración 4-9: Conversión a formato PDF o TXT



# 5 PLAN DE PRUEBAS

---

*“Lo mismo que el hierro se oxida por falta de uso  
y el agua estancada se vuelve putrefacta,  
también la inactividad destruye el intelecto.”*  
-Leonardo da Vinci.

**E**n esta sección hablaremos de las pruebas a las que se someterá esta aplicación, con la correspondiente aclaración de cada prueba en concreto.

## 5.1 Introducción

El plan de prueba describe el ámbito del esfuerzo de prueba general y proporciona un registro del proceso de planificación de prueba. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Un plan de pruebas identifica los requisitos, riesgos, casos de prueba y entornos de prueba que hay que probar, los objetivos de negocio y calidad, las planificaciones de prueba y otros elementos.

- Pruebas unitarias: Estas pruebas se basan en ejecutar cada módulo, lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores. Son independientes unas de otras y deben cubrir la totalidad del código.
- Pruebas de integración: Una vez superado las pruebas unitarias, estas pruebas se realizan en el ámbito del desarrollo software, las cuales prueban que todos los elementos unitarios funcionan juntos correctamente.
- Pruebas de validación: Se centran en las acciones y reacciones visibles para el usuario.

## 5.2 Pruebas

### 5.2.1 Pruebas unitarias

<b>Nombre</b>	<b>PU-01</b>
<b>Objetivo</b>	Creación de panels de preguntas
<b>Prerrequisitos de la prueba</b>	Creación de un nuevo panel de pregunta, el cual posee las opciones de borrar pregunta y añadir/borrar respuestas, del tipo previamente seleccionado que se

	añadirá debajo de la última pregunta creada o importada en la aplicación
<b>Procedimiento</b>	Una vez se ha seleccionado el tipo de fichero a generar, se pueden crear manualmente preguntas nuevas que añadir al fichero que queremos obtener, para ello pulsamos el botón de añadir pregunta.
<b>Superación de la prueba</b>	Cuando se elige tipo de fichero de preguntas, aparecerá visible el botón de añadir preguntas con el que la aplicación añade un nuevo de panel de preguntas vacío a la aplicación, el cual hay que cumplimentar, para su posterior generación en formato XML.
<b>Comentarios</b>	El botón de añadir pregunta solo aparecerá visible cuando se haya seleccionado un tipo de fichero de preguntas en los selectores de la aplicación.

Tabla 1: PU-01

<b>Nombre</b>	<b>PU-02</b>
<b>Objetivo</b>	Generación de fichero XML
<b>Prerrequisitos de la prueba</b>	Generar un fichero de preguntas de un juego en formato XML
<b>Procedimiento</b>	Una vez se tengan todos los panels de preguntas completos, se generará un fichero con esa información en formato XML
<b>Superación de la prueba</b>	La aplicación ha detectado que todos los campos están cumplimentados y se corresponden al tipo de juego seleccionado y descarga un fichero en formato XML con dicha información.
<b>Comentarios</b>	Es necesario haber rellenado todos los campos obligatorios y haber tenido en cuenta las restricciones de cada tipo de juego.

Tabla 2: PU-02

<b>Nombre</b>	<b>PU-03</b>
<b>Objetivo</b>	Importación de fichero XML

<b>Prerrequisitos de la prueba</b>	Importar un fichero de preguntas de un juego en formato XML
<b>Procedimiento</b>	Una vez se tenga el tipo de juego seleccionado, se podrá importar las preguntas de un fichero de preguntas completo en formato XML.
<b>Superación de la prueba</b>	La aplicación ha comprobado que se ha seleccionado el tipo de juego y que el fichero de preguntas a importar es compatible con el tipo anteriormente seleccionado, con lo cual, importa todas las preguntas a la aplicación.
<b>Comentarios</b>	Es necesario haber seleccionado el tipo de juego e importar un juego compatible con el seleccionado, de lo contrario se informará al usuario de q dicho fichero no es compatible y no se importará.

Tabla 3: PU-03

<b>Nombre</b>	<b>PU-04</b>
<b>Objetivo</b>	Importación de fichero TXT
<b>Prerrequisitos de la prueba</b>	Importar un fichero de preguntas de un juego en formato TXT
<b>Procedimiento</b>	Una vez se tenga el tipo de juego seleccionado, se podrá importar las preguntas de un fichero de preguntas completo en formato TXT.
<b>Superación de la prueba</b>	La aplicación ha comprobado que se ha seleccionado el tipo de juego y que el fichero de preguntas a importar es compatible con el tipo anteriormente seleccionado, con lo cual, importa todas las preguntas a la aplicación.
<b>Comentarios</b>	Es necesario haber seleccionado el tipo de juego e importar un juego compatible con el seleccionado, de lo contrario se informará al usuario de q dicho fichero no es compatible y no se importará.

Tabla 4: PU-04

<b>Nombre</b>	<b>PU-05</b>
<b>Objetivo</b>	Conversión de fichero a PDF
<b>Prerrequisitos de la prueba</b>	Importar un fichero de preguntas de un juego en formato XML en la sección exporter para su

	conversion a formato PDF.
<b>Procedimiento</b>	En la sección exportar de la aplicación seleccionamos el fichero XML generado por la aplicación con las preguntas del juego para subirlo al navegador. Una vez lo tenemos subido, pulsando sobre el botón PDF, se descargará automáticamente un fichero en formato PDF con la información del fichero en formato XML.
<b>Superación de la prueba</b>	La aplicación comprueba que el formato del XML es el adecuado para su conversión y genera el fichero en formato PDF para su descarga automática en el navegador.
<b>Comentarios</b>	Sólo se podrá subir al navegador ficheros con formato XML que hayan sido generados por la aplicación.

Tabla 5: PU-05

<b>Nombre</b>	<b>PU-06</b>
<b>Objetivo</b>	Conversión de fichero a TXT
<b>Prerrequisitos de la prueba</b>	Importar un fichero de preguntas de un juego en formato XML en la sección exportar para su conversión a formato TXT soportado por la aplicación AJDA.
<b>Procedimiento</b>	En la sección exporter de la aplicación seleccionamos el fichero XML generado por la aplicación con las preguntas del juego para subirlo al navegador. Una vez lo tenemos subido, pulsando sobre el botón TXT, se descargará automáticamente un fichero en formato TXT con la información del fichero en formato XML.
<b>Superación de la prueba</b>	La aplicación comprueba que el formato del XML es el adecuado para su conversión y genera el fichero en formato TXT para su descarga automática en el navegador.
<b>Comentarios</b>	Sólo se podrá subir al navegador ficheros con formato XML que hayan sido generados por la aplicación.

Tabla 6: PU-06

## 5.2.2 Pruebas de integración

<b>Nombre</b>	<b>PI-01</b>
<b>Objetivo</b>	Importación de tipo de fichero de preguntas distinto al seleccionado pero compatible y generación en XML.
<b>Prerrequisitos de la prueba</b>	Importar un fichero de preguntas de un juego compatible con el que tenemos actualmente seleccionado en el selector de tipos.
<b>Procedimiento</b>	Se importa un fichero compatible, se añaden más preguntas y/o se modifican las importadas. Se pulsa el botón generar fichero el cual hace las validaciones pertinentes para comprobar que todas las preguntas cumplen las propiedades del tipo de fichero seleccionado en el selector para su posterior generación en XML.
<b>Superación de la prueba</b>	La aplicación es capaz de comprobar que el fichero importado es de tipo compatible y al pulsar sobre el botón generar fichero, comprueba que todas las preguntas cumplen las propiedades del tipo elegido y las que no, manda un aviso al usuario para que éste la adapte conforme las propiedades del tipo seleccionado.
<b>Comentarios</b>	Cuando una pregunta no cumple las propiedades del tipo elegido, se mostrará un mensaje de error en la parte superior derecha de la pantalla indicando el número de preguntas con errores y se marcarán en la aplicación las preguntas que sean erróneas.

Tabla 7: PI-01

<b>Nombre</b>	<b>PI-02</b>
<b>Objetivo</b>	Conversión a formato PDF o TXT de un fichero en formato XML previamente generado por el usuario en la aplicación
<b>Prerrequisitos de la prueba</b>	Generar un fichero en formato XML desde esta aplicación web, por ejemplo uno como en el caso de la prueba de integración anterior.
<b>Procedimiento</b>	En la sección exportar de la aplicación seleccionamos el fichero XML generado por la aplicación con las

	preguntas del juego para subirlo al navegador. Una vez lo tenemos subido, pulsando sobre el botón PDF o TXT, se validará que el contenido está en el formato correcto creado por esta aplicación y se descargará automáticamente un fichero en formato PDF o TXT con la información del fichero en formato XML.
<b>Superación de la prueba</b>	La aplicación comprueba que el formato del XML es el adecuado para su conversión y genera el fichero en formato PDF o TXT para su descarga automática en el navegador.
<b>Comentarios</b>	Cuando se importa un XML a esta sección y pasa el filtro de extensión de fichero importado, si la aplicación no detecta que el contenido de dicho fichero está en el formato generado por esta aplicación no se procederá a su conversión.

Tabla 8: PI-02

### 5.2.3 Pruebas de validación

<b>Nombre</b>	<b>PV-01</b>
<b>Objetivo</b>	Se generan ficheros de preguntas de juegos con los que el usuario puede utilizarlos para elaborar pruebas manuscritas o para importarlos a la aplicación AJDA.
<b>Prerrequisitos de la prueba</b>	<ul style="list-style-type: none"> <li>• Generar un fichero en formato XML desde esta aplicación web.</li> <li>• Exportar dicho fichero a formato TXT (para su uso en AJDA) o PDF para su uso en pruebas manuscritas.</li> </ul>
<b>Procedimiento</b>	El usuario puede crear/modificar ficheros de preguntas de juegos para su posterior exportación al formato que más le convenga.
<b>Superación de la prueba</b>	Se deberá generar un fichero de preguntas de juegos, si todo está bien, en el formato elegido XML, PDF o TXT.
<b>Comentarios</b>	

Tabla 9: PV-01



# 6 CONCLUSIONES

---

*“El placer más noble es el júbilo de comprender.”*

*-Leonardo da Vinci.*

Con la elaboración de este proyecto, llegamos a la conclusión de que con esta aplicación web podríamos hacerle ahorrar al usuario de la aplicación AJDA mucho trabajo de aprendizaje sobre cómo funciona internamente AJDA, ya que no es necesario que dicho usuario comprenda el funcionamiento de los ficheros generados en formato TXT cuando quiera hacer uso de uno de ellos para modificarlo o para mezclarlo con otro tipo de fichero de preguntas en la aplicación AJDA.

Como conclusiones propias de este proyecto, he de destacar la línea de aprendizaje que he tenido. Partiendo de la base obtenida en la asignatura Fundamentos de Aplicaciones y Servicios Telemáticos, he podido añadir a mi conocimiento una nueva tecnología de programación como es JSF y el uso de PrimeFaces usando la documentación en línea.

También me gustaría destacar que este proyecto me ha servido para enfrentarme por primera vez a un problema software con los requisitos pertinentes del cliente de forma individual.

Por último, me gustaría resaltar los siguientes puntos sobre lecciones aprendidas durante la elaboración de este proyecto y que sin duda me serán de gran ayuda de cara al futuro:

- Los comentarios en el código son importantísimos a la hora de elaborar, depurar y mantener un proyecto.
- El control de versiones es un requisito indispensable en proyectos software de este tipo y tamaño.
- Organización. Primero hay que pensar en qué vas a hacer, cómo y por qué y luego ponerte a ello, se ahorra mucho tiempo y futuros problemas.
- El inglés es una herramienta esencial para la elaboración de proyectos de este tipo ya que toda la documentación disponible en línea está en inglés.
- Una buena bibliografía es esencial en cualquier proyecto software. Es importante saber dónde buscar y cómo buscarlo, las palabras claves en la búsqueda pueden diferenciar un resultado exitoso de otro que no lo es.



# 7 LÍNEAS DE MEJORA

---

*“La persistencia es muy importante.*

*No debes renunciar al menos que te veas obligado a renunciar.”*

*-Elon Musk.*

**E**L marco de este proyecto es muy amplio y admite varias líneas de mejora. Las cuales ayudarían bastante al usuario de cara al futuro y facilitarían el uso de la aplicación de AJDA a usuarios con poco conocimiento en el campo de la informática:

- **Conversiones a formato a Word, Excel, etc:** La conversión de estos ficheros de preguntas procedentes de la aplicación AJDA en formato TXT a formato Word o Excel, por ejemplo, facilitarían mucho la labor del usuario de la aplicación a la hora de elaborar nuevos ficheros de preguntas o a reutilizar información contenida en otros ficheros del mismo tipo.
- **Almacenamiento en Base de datos:** Una vez se tengan generados los ficheros de preguntas en el formato deseado, una línea de mejora podría ser el almacenamiento de dichos ficheros en una base de datos, que además de recoger los ficheros de preguntas, también recopile el nombre del creador y si ese fichero puede ser utilizado por otros usuarios de la aplicación o no.
- **Buscador de ficheros en base de datos:** Otra posible línea de mejora podría ser la elaboración de un buscador de ficheros de la base de datos anteriormente mencionada, con la finalidad de poder filtrar la búsqueda por nombre, palabra coincidente, materia, si el fichero es público o privado, etc.
- **Conversión directa al formato deseado:** Esta línea de mejora permitiría al usuario generar directamente el fichero de preguntas en el formato que mas le conveniese de manera directa, sin tener que generarlo previamente en formato XML para su conversión.



# REFERENCIAS Y BIBLIOGRAFÍA

---

- [1] Universidad de Alicante, " Modelo vista controlador (MVC) ". [Recurso en línea]. Disponible en: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- [2] Universidad de Alicante, " El MVC en JavaServer Faces". [Recurso en línea]. Disponible en: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion02-apuntes.html>
- [3] Wikipedia, " Primefaces". [Recurso en línea]. Disponible en: <https://es.wikipedia.org/wiki/PrimeFaces>
- [4] Primefaces.org, " Primefaces". [Recurso en línea]. Disponible en: <https://www.primefaces.org/showcase/>
- [5] Richfaces.jboss, " Richfaces". Disponible en: <http://richfaces.jboss.org/>
- [6] ICESOFT technologies inc, " ICEfaces". [Recurso en línea]. Disponible en: <http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>
- [7] Tutorialspoint, "JSF - Managed Beans ". [Recurso en línea]. Disponible en: [https://www.tutorialspoint.com/jsf/jsf\\_managed\\_beans.htm](https://www.tutorialspoint.com/jsf/jsf_managed_beans.htm)
- [8] Wikipedia, "Pruebas de software". [Recurso en línea]. Disponible en: [https://es.wikipedia.org/wiki/Pruebas\\_de\\_so](https://es.wikipedia.org/wiki/Pruebas_de_so)
- [9] Itextpdf, "iTetx". [Recurso en línea]. Disponible en: <https://itextpdf.com/>
- [10] "Documentación HTML ". [Recurso en línea]. Disponible en: <https://www.w3schools.com/html/>
- [11] Jesús. M. Muñoz Calle et al., "ADJA, Aplicación de juegos didácticos en el aula." [Recurso en línea]. Disponible en: <http://newton.proyectedescartes.org/juegosdidacticos/index.php>



