



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Generación de cuestionarios  
sobre reglas de asociación  
(AssoQuiz-Generator)  
Documentación Técnica**



Presentado por Sergio Revilla Merino  
en Universidad de Burgos — 13 de febrero  
de 2023

Tutor: Juan José Rodríguez Díez



---

# Índice general

---

<b>Índice general</b>	<b>i</b>
<b>Índice de figuras</b>	<b>iii</b>
<b>Índice de tablas</b>	<b>v</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	7
<b>Apéndice B Especificación de Requisitos</b>	<b>13</b>
B.1. Introducción . . . . .	13
B.2. Objetivos generales . . . . .	13
B.3. Catálogo de requisitos . . . . .	14
B.4. Especificación de requisitos . . . . .	17
<b>Apéndice C Especificación de diseño</b>	<b>25</b>
C.1. Introducción . . . . .	25
C.2. Diseño de datos . . . . .	25
C.3. Diseño procedimental . . . . .	30
C.4. Diseño arquitectónico . . . . .	32
<b>Apéndice D Documentación técnica de programación</b>	<b>35</b>
D.1. Introducción . . . . .	35
D.2. Estructura de directorios . . . . .	35
D.3. Manual del programador . . . . .	39

D.4. Compilación, instalación y ejecución del proyecto . . . . .	43
D.5. Pruebas del sistema . . . . .	46
<b>Apéndice E Documentación de usuario</b>	<b>47</b>
E.1. Introducción . . . . .	47
E.2. Requisitos de usuarios . . . . .	47
E.3. Instalación . . . . .	48
E.4. Manual del usuario . . . . .	48
<b>Bibliografía</b>	<b>55</b>

---

# Índice de figuras

---

A.1. <i>sprint</i> 1 - Burndown report . . . . .	4
A.2. <i>Sprint</i> 2 - Burndown report . . . . .	5
A.3. <i>Sprint</i> 3 - Burndown report . . . . .	6
A.4. <i>Sprint</i> 4 - Burndown report . . . . .	7
B.1. Diagrama de casos de uso . . . . .	18
C.1. Diagrama Entidad/Relación . . . . .	26
C.2. Diagrama de clases de la interfaz . . . . .	27
C.3. Diagrama de clases del generador . . . . .	28
C.4. Diagrama de clases del traductor . . . . .	29
C.5. Diagrama de secuencia . . . . .	31
C.6. Arquitectura hexagonal . . . . .	33
D.1. Estructura de paquetes . . . . .	37
D.2. Estructura de paquetes del dominio . . . . .	37
D.3. Estructura de paquetes del generador . . . . .	38
D.4. Estructura de paquetes del traductor . . . . .	39
D.5. Estructura de paquetes de la interfaz de usuario . . . . .	39
D.6. Contenido del fichero <i>pom.xml</i> . . . . .	43
D.7. Ejecución del proyecto en Eclipse . . . . .	45
E.1. Icono <i>AssoQuiz-Generator</i> . . . . .	48
E.2. Ventana principal . . . . .	49
E.3. Ventana configuración <i>Generación Reglas Asociación</i> . . . . .	49
E.4. Ventana de guardado de un archivo . . . . .	50
E.5. Ventana finalizado . . . . .	50
E.6. Vista de la pregunta <i>Generación Item Sets</i> resuelta en <i>Moodle</i> . . . . .	51

E.7. Vista de la pregunta <i>Generación Reglas Asociación</i> resuelta en <i>Moodle</i> . . . . .	52
E.8. Vista de la pregunta <i>Ampliación Item Sets</i> resuelta en <i>Moodle</i> .	53

---

# Índice de tablas

---

A.1. Equivalencia entre <i>story points</i> y tiempo. . . . .	2
A.2. <i>Issues</i> del <i>sprint</i> 1 . . . . .	3
A.3. <i>Issues</i> del <i>sprint</i> 2 . . . . .	4
A.4. <i>Issues</i> del <i>sprint</i> 3 . . . . .	6
A.5. <i>Issues</i> del <i>sprint</i> 4 . . . . .	7
A.6. Licencias de las dependencias del proyecto. . . . .	11
B.1. CU-1 Acerca de. . . . .	19
B.2. CU-2 Ampliación Item Sets. . . . .	20
B.3. CU-3 Generación Reglas Asociación. . . . .	20
B.4. CU-4 Generación Item Sets. . . . .	21
B.5. CU-5 Configuración de Ampliación Item Sets. . . . .	21
B.6. CU-6 Configuración del conjunto de datos. . . . .	22
B.7. CU-7 Exportar XML. . . . .	23





## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

El apartado se centra en el Plan de Proyecto de Software y su importancia para el desarrollo de software de alta calidad. Se analizan dos aspectos clave del plan de proyecto: la planificación temporal y la viabilidad económica y legal. La planificación temporal establece un cronograma para el desarrollo del software y permite seguir el progreso y prever posibles desafíos. La viabilidad económica y legal evalúa la viabilidad financiera y legal del proyecto y asegura su sostenibilidad a largo plazo. Con un plan de proyecto bien estructurado y detallado, se puede establecer un camino claro hacia el éxito del software y minimizar los riesgos y los problemas a lo largo del proceso de desarrollo. [8]

### A.2. Planificación temporal

La planificación temporal es esencial para establecer un cronograma realista y efectivo para el desarrollo del software. Este cronograma incluye fechas clave para hitos importantes y la asignación de tareas a los miembros del equipo. La planificación temporal permite un seguimiento riguroso del progreso y una identificación temprana de posibles desafíos o ajustes necesarios.

Para este proyecto se decidió utilizar una metodología ágil como SCRUM. No obstante, no se ha seguido al 100 % esta metodología ya que el proyecto se ha hecho en solitario y es de carácter educativo. A continuación se especifican las tareas que se han ido desarrollando en cada *sprint* del proyecto. En este

proyecto se decidió que cada *sprint* tendría una duración de dos semanas. Una vez finalizado el *sprint*, se tenía una reunión con el tutor para revisar todo lo que se había hecho en ese periodo de tiempo. También se usaron gráficos *burndown* para monitorizar el proceso y se usaron tableros canvas.

La estimación de las tareas se realizó mediante story points. A continuación se muestra una tabla de equivalencia entre los *story points* y la estimación temporal que se decidió otorgar a cada uno.

<i>Story points</i>	Estimación temporal
1	15mins
2	45mins
3	2h
5	5h
8	12h
13	24h
21	2,5 días
40	1 semana

Tabla A.1: Equivalencia entre *story points* y tiempo.

### ***Sprint 0: hasta el 15 de Diciembre***

Esto no es un *sprint* como tal sino que es un proceso de búsqueda de información y de reuniones con el tutor para planificar el proyecto y empezar a darle forma.

Durante esta fase se aprendió a realizar los ejercicios del tema de reglas de asociación de la asignatura de *Minería de Datos* a mano, para lo cual, se tuvieron que leer los correspondientes apuntes de la asignatura y ver algún vídeo explicativo.

En esta fase se instalaron también las herramientas que se iban a utilizar y se creó el repositorio en GitHub.

### ***Sprint 1: 15 de Diciembre - 29 de Diciembre***

Siguiendo los consejos que el tutor dio a través de reuniones por Teams, lo primero que se hizo fue crear una pregunta en *Moodle* de tipo opción múltiple y exportarla a un formato .XML para tener una idea de cómo

tienen que ser los ficheros que se van a generar. Esta pregunta exportada sirvió como plantilla original para el proyecto.

Se creó la primera pregunta, que consistió en generar conjuntos de datos aleatorios de n-item sets y soluciones para los mismos, clasificándolas en válidas y no válidas. Una vez se tuvo la pregunta funcionando, se creó una clase para traducirla a un formato .XML que Moodle fuese capaz de entender, usando para ello la plantilla que se había creado anteriormente.

Durante esta fase también se realizó una búsqueda de información para aprender a crear interfaces gráficas en Java.

Índice	Descripción
#1	Investigar como hacer una interfaz gráfica
#2	Generar item sets aleatorios
#3	Generar posibles soluciones
#4	Generar respuestas correctas e incorrectas
#5	Generar plantilla para la pregunta de tipo <i>Ampliación Item Sets</i>
#6	Crear traductor para las preguntas de tipo <i>Ampliación Item Sets</i>

Tabla A.2: *Issues* del *sprint* 1

Figura A.1: *sprint* 1 - Burndown report

### ***Sprint* 2: 29 de Diciembre - 13 de Enero**

En este *sprint* se configuró el proyecto como un proyecto de *Maven* para poder utilizar las dependencias externas de la librería de *Weka*, por lo que previamente se investigó como hacerlo.

También se empezó a desarrollar la pregunta de tipo *Generación Reglas Asociación*. Para ello se hicieron pruebas usando *Weka* y posteriormente se desarrolló en el código la parte de crear el conjunto de datos aleatorios para reglas de asociación. Una vez hecho esto, se consiguió obtener soluciones y clasificarlas en válidas y no válidas.

Índice	Descripción
#7	Crear conjunto de datos para reglas de asociación
#8	Crear soluciones para la pregunta de tipo <i>Generación Reglas Asociación</i>
#9	Utilizar <i>Maven</i> para las dependencias externas

Tabla A.3: *Issues* del *sprint* 2

Figura A.2: *Sprint 2* - Burndown report

### ***Sprint 3: 13 de Enero - 27 de Enero***

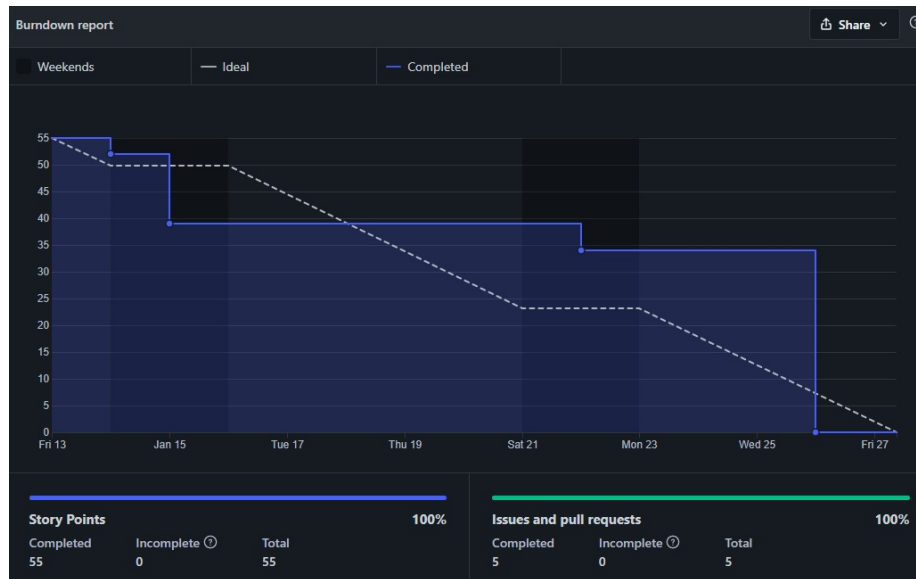
En esta fase del proyecto se solucionó un bug referente a la visualización de las reglas de asociación en *Moodle*, que no se mostraban de forma correcta. También se reorganizó el código por primera vez para tenerlo más ordenado y limpio.

Además de estos arreglos, se implementaron funcionalidades nuevas como la de la opción de que el conjunto de datos contenga columnas numéricas de forma que se puedan discretizar entre unos intervalos determinados.

Por último se realizó la interfaz gráfica. Para ello se reorganizó primero el código de nuevo. Después se empezaron a crear las ventanas correspondientes y una vez se hizo esto, se crearon botones para enlazar la interfaz gráfica con las funciones del código de forma que todo sea más visual y fácil de usar.

En este *sprint* no se cumplieron todas las tareas que se tenían previstas, de forma que las que faltaban por completar se dejaron para el siguiente.

Índice	Descripción
#10	Las opciones de la pregunta de tipo <i>Generación Reglas Asociación</i> no se muestran correctamente en <i>Moodle</i>
#13	Reorganizar código
#14	Añadir atributos numéricos en preguntas de tipo <i>Generación Reglas Asociación</i>
#16	Editar el enunciado de la plantilla para cuando haya atributos a discretizar
#17	Interfaz gráfica

Tabla A.4: *Issues* del *sprint 3*Figura A.3: *Sprint 3* - Burndown report

### ***Sprint 4: 28 de Enero - 11 de Febrero***

En este *sprint* se completaron las tareas que habían quedado pendientes de realizar en el anterior.

También se introdujo una nueva pregunta llamada *Generación Item Sets* siguiendo las indicaciones del tutor.

Se realizó el tratamiento de excepciones y se introdujo la posibilidad de establecer valores aleatorios en la configuración.

Índice	Descripción
#11	El conjunto de datos para la pregunta de tipo <i>Generación Reglas Asociación</i> no se muestran correctamente en <i>Moodle</i>
#15	Tratamiento de excepciones
#18	Introducir opción de aleatorio
#20	El formato de los valores numericos en las opciones de reglas de asociacion aparece de forma incorrecta
#21	Introducir nueva pregunta

Tabla A.5: Issues del *sprint 4*

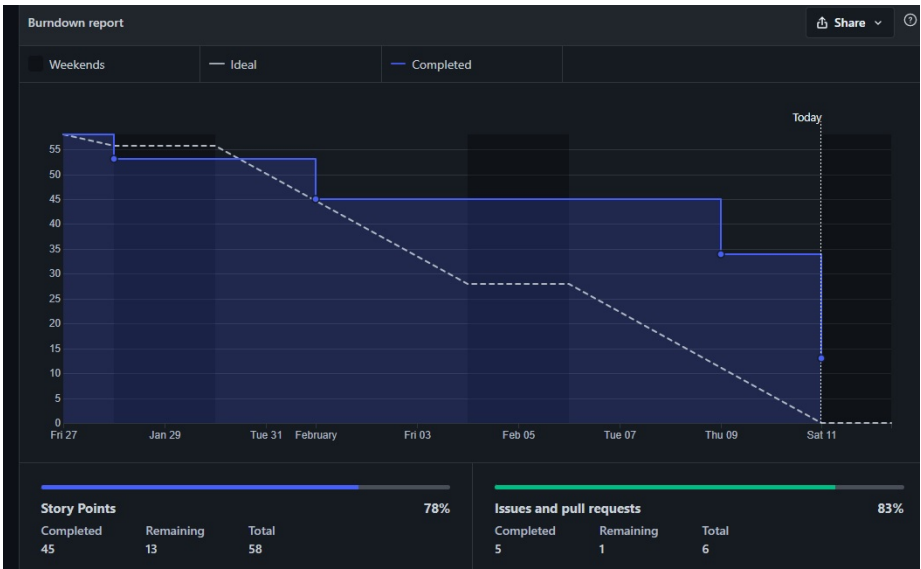


Figura A.4: *Sprint 4* - Burndown report

### A.3. Estudio de viabilidad

La viabilidad económica y legal es un aspecto fundamental del plan de proyecto de software que evalúa si el proyecto es viable desde un punto de vista financiero y legal. Este análisis incluye una revisión detallada de los costos y los ingresos proyectados, así como una revisión de las leyes y regulaciones relevantes. La viabilidad económica y legal ayuda a asegurar que el proyecto sea sostenible a largo plazo y no enfrente obstáculos legales durante su ejecución.

En este caso, dado el contexto académico del proyecto, no se requiere un estudio de viabilidad exhaustivo. No obstante, en este apartado se realizará un estudio simple de viabilidad, tanto económica como legal.

## Viabilidad económica

En este apartado se van a analizar de forma estimada los costes y beneficios que podría haber supuesto el proyecto en caso de que se hubiese realizado en una empresa real.

### Costes

#### 1. Costes de personal

Para el cálculo del coste de personal se debe tener en cuenta al desarrollador y a los dos tutores del proyecto que también deben recibir una retribución por el apoyo prestado.

- Desarrollador: Graduado en Ingeniería Informática que recibe un salario de 12€/h. Suponiendo que se han trabajado unas 6 horas al día, el desarrollador supone 504€/semana. La duración del proyecto ha sido de 12 semanas, por lo que el coste del desarrollador será de 6048€.
- Tutores: Titulados en Ingeniería Informática que suponemos que cobran 20€/h. La implicación media del tutor es de 1h/semana, y teniendo en cuenta que el proyecto ha durado 12 semanas se obtiene lo siguiente:

$$1 \text{ tutor} \times \frac{20\text{€}}{1 \text{ hora}} \times \frac{1 \text{ hora}}{1 \text{ semana}} \times 12 \text{ semanas} = 240\text{€}$$

Si se suma el coste del desarrollador y del tutor, el resultado es el siguiente:

$$6048\text{€} + 240\text{€} = 6288\text{€}$$

A este salario bruto hay que sumarle el coste de la Seguridad Social, que es, aproximadamente de un 30 % sobre el salario bruto:

$$6288\text{€} \times 0,3\text{€} = 1886,4\text{€}$$

Si sumamos el salario bruto más el coste de la Seguridad Social, obtenemos el resultado final del coste de personal:

$$6288\text{€} + 1886,4\text{€} = 8174,4\text{€}$$



## 2. Coste de software

El coste de software en este proyecto ha sido de 0€, ya que se ha trabajado con herramientas con licencia gratuita.

## 3. Coste de Hardware

El coste de hardware en este proyecto es equivalente al coste del ordenador portátil que se ha utilizado para el mismo. En este caso, se ha utilizado un portátil con 8GB de RAM con un precio de 300€.

## 4. Coste de Impresión

- Impresión de la memoria : 25€

## 5. Costes totales

Por lo tanto, si se suman todos los costes anteriores, se obtiene un coste total de 8499.4€.

## Beneficios

La aplicación desarrollada no genera ningún beneficio económico puesto que es un proyecto de carácter educativo y se va a distribuir de forma gratuita.

## Viabilidad legal

En esta sección se va a tratar el tema de las licencias de los software que se han utilizado para la creación de este proyecto.

Se va a hacer un análisis de las licencias que tiene cada dependencia que se ha utilizado, y a partir de ahí, se va a establecer una licencia para este proyecto, según lo que más se ajuste

Dado que el programa usa dependencias con licencia GPL y OTN, se ha elegido una licencia compatible con ambas. Una de las licencias compatibles sería la GPL v3.0, ya que es compatible con la GPL v2.0 (que es la licencia de la mayoría de las aplicaciones basadas en GPL) y también permite la distribución de software combinado con código de terceros con licencias distintas a la GPL. [5]

Las características generales de esta licencia son las siguientes:

- **Copyleft:** La GPL impone un copyleft fuerte, lo que significa que todas las obras derivadas de un programa licenciado bajo la GPL deben ser licenciadas bajo la misma licencia.

- **Fuente abierta:** La GPL requiere que el código fuente de un programa licenciado bajo ella sea disponible para cualquiera que lo desee.
- **Distribución libre:** La GPL permite la distribución libre y sin restricciones del software, siempre y cuando se respeten las condiciones de la licencia.
- **Modificaciones permitidas:** La GPL permite a los usuarios modificar y distribuir versiones modificadas del software licenciado bajo ella.
- **No discriminación:** La GPL no discrimina a ningún grupo de usuarios, incluyendo a los desarrolladores comerciales.
- **Aplicación a todas las partes:** La GPL se aplica a todas las partes del software, incluyendo cualquier componente o biblioteca que se incluya en el programa.
- **No garantía:** La GPL no proporciona ninguna garantía o responsabilidad por el software licenciado bajo ella.

[3]

Dependencia	Descripción	Licencia
<b>Java.Util</b>	Contiene clases y funciones útiles para realizar tareas comunes, como manejo de colecciones, formateo de fechas, generación de números aleatorios, etc.	GPL
<b>Java.Math</b>	Contiene clases para realizar operaciones matemáticas complejas.	OTN
<b>Java.IO</b>	Proporciona clases para realizar operaciones de entrada/salida básicas, como leer y escribir archivos y streams de datos.	OTN
<b>Java.Swing</b>	Biblioteca de componentes de interfaz gráfica de usuario.	OTN
<b>Java.Awt</b>	Biblioteca para crear aplicaciones gráficas de escritorio con interfaces de usuario y ventanas.	OTN
<b>Weka.Core</b>	Se usa para el procesamiento y análisis de datos en el marco de trabajo de aprendizaje automático de Weka.	GPL

Tabla A.6: Licencias de las dependencias del proyecto.



## Apéndice *B*

---

# Especificación de Requisitos

---

### B.1. Introducción

Este apéndice es una sección fundamental para el éxito de cualquier proyecto de software. En este apartado se describen los objetivos generales y se hace un catálogo exhaustivo de los requisitos necesarios para lograr esos objetivos. La especificación de requisitos es el proceso de identificar y documentar los requisitos de software que deben ser satisfechos para garantizar que el proyecto cumpla con las expectativas deseadas.

La importancia de esta sección radica en que es la base para el diseño, desarrollo, y pruebas del software. Si los requisitos no están claramente definidos, es probable que el proyecto se desvíe de su objetivo original y que los resultados finales no sean satisfactorios. Por lo tanto, la especificación de requisitos es un paso crítico en el desarrollo de software de alta calidad y es esencial para garantizar el éxito del proyecto. [6]

### B.2. Objetivos generales

Los objetivos generales del proyecto son los siguientes:

1. Generación de bancos de preguntas que contengan preguntas generadas de los siguientes tipos: *Generación Item Sets*, *Generación Reglas de Asociación* y *Ampliación Item Sets*.
2. Configuración de diferentes parámetros para la generación de los bancos de preguntas, como el número de preguntas, el número de item sets o atributos, y otros, por el usuario.

3. Capacidad de establecer valores de forma aleatoria si el usuario no define algún parámetro en la configuración.
4. Obtención de un archivo .XML compatible con la plataforma *Moodle* de forma que contenga todas las preguntas generadas de un mismo tipo.

### B.3. Catálogo de requisitos

A continuación, se enumeran los requisitos específicos derivados de los objetivos generales del proyecto.

#### Requisitos funcionales

- **RF-1 Generación de banco de preguntas:** la aplicación tiene que ser capaz de generar bancos de preguntas sobre reglas de asociación para la asignatura de minería de datos. Un banco de preguntas contiene preguntas que a su vez, tienen título, enunciado y opciones.
  - **RF-1.1 Generar banco de preguntas *Ampliación Item Sets*:** La aplicación debe ser capaz de generar un banco de preguntas que contenga todas las preguntas generadas de este tipo.
    - **RF-1.1.1: Definir el número de preguntas a generar:** el usuario tiene que ser capaz de configurar el número de preguntas de este tipo que quiere generar.
    - **RF-1.1.2: Definir el número de item sets:** el usuario tiene que poder definir el número de item sets que formarán parte del conjunto de n-item sets que va a ser generado.
    - **RF-1.1.3: Definir el tamaño de los item sets:** el usuario tiene que poder definir el tamaño de los item sets que formarán parte del conjunto de n-item sets que va a ser generado.
    - **RF-1.1.4: Establecer la configuración de forma aleatoria:** el usuario no tiene que estar obligado a definir la configuración de todos los parámetros anteriores, de forma que si no establece un valor determinado para un parámetro, este valor se generará de forma aleatoria.
  - **RF-1.2 Generar banco de preguntas *Generación Reglas de Asociación*:** La aplicación debe ser capaz de generar un banco de preguntas que contenga todas las preguntas generadas de este tipo.

- **RF-1.2.1: Definir el número de preguntas a generar:** el usuario tiene que ser capaz de configurar el número de preguntas de este tipo que quiere generar.
  - **RF-1.2.2: Definir el número de atributos:** el usuario tiene que poder definir el número de atributos que formarán parte del conjunto de datos que va a ser generado.
  - **RF-1.2.3: Definir el número de instancias:** el usuario tiene que poder definir el número de instancias que formarán parte del conjunto de datos que va a ser generado.
  - **RF-1.2.4: Definir el soporte mínimo:** el usuario tiene que poder definir el valor del soporte mínimo del problema.
  - **RF-1.2.5: Definir la confianza mínima:** el usuario tiene que poder definir el valor de la confianza mínima del problema.
  - **RF-1.2.6: Especificar si quiere atributos numéricos para ser discretizados:** el usuario tiene que poder establecer si quiere atributos numéricos para que sean discretizados.
  - **RF-1.2.7: Definir el número de intervalos a discretizar:** el usuario tiene que poder definir el número de intervalos en los que se va a discretizar el atributo numérico.
  - **RF-1.2.8: Establecer la configuración de forma aleatoria:** el usuario no tiene que estar obligado a definir la configuración de todos los parámetros anteriores, de forma que si no establece un valor determinado para un parámetro, este valor se generará de forma aleatoria.
- **RF-1.3 Generar banco de preguntas *Generación Item Sets*:** La aplicación debe ser capaz de generar un banco de preguntas que contenga todas las preguntas generadas de este tipo.
    - **RF-1.3.1: Definir el número de preguntas a generar:** el usuario tiene que ser capaz de configurar el número de preguntas de este tipo que quiere generar.
    - **RF-1.3.2: Definir el número de atributos:** el usuario tiene que poder definir el número de atributos que formarán parte del conjunto de datos que va a ser generado.
    - **RF-1.3.3: Definir el número de instancias:** el usuario tiene que poder definir el número de instancias que formarán parte del conjunto de datos que va a ser generado.
    - **RF-1.3.4: Definir el soporte mínimo:** el usuario tiene que poder definir el valor del soporte mínimo del problema.

- **RF-1.3.5: Definir la confianza mínima:** el usuario tiene que poder definir el valor de la confianza mínima del problema.
  - **RF-1.3.6: Especificar si quiere atributos numéricos para ser discretizados:** el usuario tiene que poder establecer si quiere atributos numéricos para que sean discretizados.
  - **RF-1.3.7: Definir el número de intervalos a discretizar:** el usuario tiene que poder definir el número de intervalos en los que se va a discretizar el atributo numérico.
  - **RF-1.3.8: Establecer la configuración de forma aleatoria:** el usuario no tiene que estar obligado a definir la configuración de todos los parámetros anteriores, de forma que si no establece un valor determinado para un parámetro, este valor se generará de forma aleatoria.
- **RF-2 Exportación de problemas:** la aplicación tiene que ser capaz de exportar todos los problemas que ha generado.
  - **RF-2.1 Introducir un nombre para el fichero:** el usuario tiene que poder introducir el nombre que desea dar a su fichero que va a generar.
  - **RF-2.2 Especificación de la ruta:** el usuario tiene que poder especificar la ruta en la que quiere que se guarde su fichero .XML generado.
  - **RF-2.3 Exportación en formato Moodle XML:** la aplicación tiene que ser capaz de exportar todos los problemas generados en un fichero .XML importable en la plataforma Moodle.
- **RF-3 Obtener información de la aplicación:** el usuario tiene que poder obtener información acerca de la aplicación.

## Requisitos no funcionales

- **RNF-1 Usabilidad:** la aplicación debe ser fácil de usar, con una interfaz gráfica sencilla de forma que no requiera aprendizaje.
- **RNF-2 Rendimiento:** la aplicación tiene que ser capaz de hacer sus funciones de forma rápida y sin estancarse.
- **RNF-3 Escalabilidad:** la aplicación tiene que estar diseñada para añadir nuevas funcionalidades de forma sencilla.
- **RNF-4 Disponibilidad:** la aplicación debe de estar disponible para su uso en todo momento, de forma que no dependa de una conexión a internet.



- **RNF-5 Mantenibilidad:** la aplicación debe soportar cambios y mejoras sin que su funcionamiento general se vea afectado.
- **RNF-6 Fiabilidad:** la aplicación debe de funcionar de manera confiable y tiene que ser capaz de recuperarse ante fallos no deseados.

## B.4. Especificación de requisitos

En esta sección se muestra el diagrama de casos de uso resultante y se desarrollar cada uno de ellos.

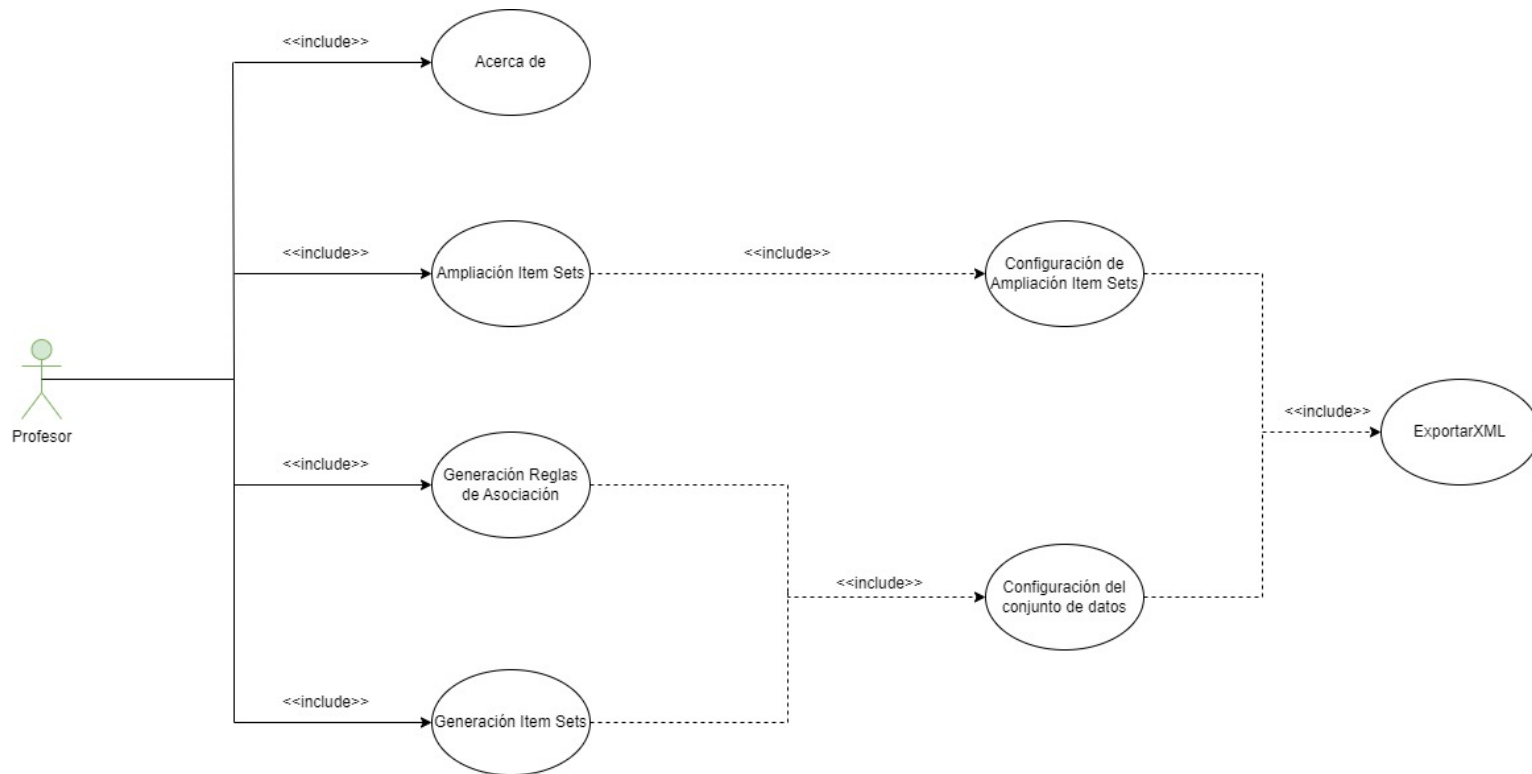


Figura B.1: Diagrama de casos de uso

## Actores

Solo interactuará con el sistema un actor, que en este caso será un profesor.

## Casos de uso

CU-1	Acerca de
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Revilla Merino
<b>Requisitos asociados</b>	RF-3
<b>Descripción</b>	Permite al usuario obtener información de la aplicación.
<b>Precondición</b>	Iniciar la aplicación.
<b>Acciones</b>	<ol style="list-style-type: none"><li>1. Pulsar el botón de <i>Acerca de</i>.</li></ol>
<b>Postcondición</b>	Se abre un panel con información de la aplicación.
<b>Excepciones</b>	-
<b>Importancia</b>	Media

Tabla B.1: CU-1 Acerca de.

<b>CU-2</b>	<b>Ampliación Item Sets</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Revilla Merino
<b>Requisitos asociados</b>	RF-1, RF-1.1, RF-1.1.1, RF-1.1.2, RF-1.1.3, RF-1.1.4
<b>Descripción</b>	Permite al usuario acceder a una ventana de configuración de este tipo de pregunta.
<b>Precondición</b>	Iniciar la aplicación.
<b>Acciones</b>	1. Pulsar el botón
<b>Postcondición</b>	Se abre la ventana para configurar este tipo de pregunta.
<b>Excepciones</b>	-
<b>Importancia</b>	Alta

Tabla B.2: CU-2 Ampliación Item Sets.

<b>CU-3</b>	<b>Generación Reglas Asociación</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Revilla Merino
<b>Requisitos asociados</b>	RF-1, RF-1.2, RF-1.2.1, RF-1.2.2, RF-1.2.3, RF-1.2.4, RF-1.2.5, RF-1.2.6, RF-1.2.7, RF-1.2.8
<b>Descripción</b>	Permite al usuario acceder a una ventana de configuración de este tipo de pregunta.
<b>Precondición</b>	Iniciar la aplicación
<b>Acciones</b>	1. Pulsar el botón
<b>Postcondición</b>	Se abre la ventana para configurar este tipo de pregunta.
<b>Excepciones</b>	-
<b>Importancia</b>	Alta

Tabla B.3: CU-3 Generación Reglas Asociación.

CU-4	Generación Item Sets
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Revilla Merino
<b>Requisitos asociados</b>	RF-1, RF-1.3, RF-1.3.1, RF-1.3.2, RF-1.3.3, RF-1.3.4, RF-1.3.5, RF-1.3.6, RF-1.3.7, RF-1.3.8
<b>Descripción</b>	Permite al usuario acceder a una ventana de configuración de este tipo de pregunta.
<b>Precondición</b>	Iniciar la aplicación.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Pulsar el botón</li> </ol>
<b>Postcondición</b>	Se abre la ventana para configurar este tipo de pregunta.
<b>Excepciones</b>	-
<b>Importancia</b>	Alta

Tabla B.4: CU-4 Generación Item Sets.

CU-5	Configuración de Ampliación Item Sets
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Revilla Merino
<b>Requisitos asociados</b>	RF-1, RF-1.1, RF-1.1.1, RF-1.1.2, RF-1.1.3, RF-1.1.4
<b>Descripción</b>	Permite al usuario configurar una pregunta de este tipo.
<b>Precondición</b>	Pulsar el botón de <i>Ampliación Item Sets</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Seleccionar un número de preguntas a generar.</li> <li>2. Seleccionar o no el tamaño del conjunto de item sets y el tamaño de los item sets.</li> </ol>
<b>Postcondición</b>	-
<b>Excepciones</b>	-
<b>Importancia</b>	Alta

Tabla B.5: CU-5 Configuración de Ampliación Item Sets.

CU-6	Configuración del conjunto de datos
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Revilla Merino
<b>Requisitos asociados</b>	RF-1, RF-1.2, RF-1.2.1, RF-1.2.2, RF-1.2.3, RF-1.2.4, RF-1.2.5, RF-1.2.6, RF-1.2.7, RF-1.2.8, RF-1.3, RF-1.3.1, RF-1.3.2, RF-1.3.3, RF-1.3.4, RF-1.3.5, RF-1.3.6, RF-1.3.7, RF-1.3.8
<b>Descripción</b>	Permite al usuario configurar las preguntas de <i>Generación Reglas Asociación</i> y <i>Generación Item Sets</i> .
<b>Precondición</b>	Pulsar el botón de <i>Generación Reglas Asociación</i> o el de <i>Generación Item Sets</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Seleccionar un número de preguntas a generar.</li> <li>2. Seleccionar o no el número de instancias a generar, el número de atributos del conjunto de datos, el valor del soporte mínimo, el valor de la confianza, si se quieren atributos numéricos y el número de intervalos.</li> </ol>
<b>Postcondición</b>	-
<b>Excepciones</b>	-
<b>Importancia</b>	Alta

Tabla B.6: CU-6 Configuración del conjunto de datos.

CU-7	Exportar XML
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Revilla Merino
<b>Requisitos asociados</b>	RF-2, RF-2.1, RF-2.2, RF-2.3
<b>Descripción</b>	Traduce y exporta el banco de preguntas en la ruta seleccionada.
<b>Precondición</b>	Se ha pulsado el botón de <i>Exportar</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Darle un nombre al fichero que se va a generar.</li> <li>2. Seleccionar la ruta deseada.</li> <li>3. Pulsar el botón <i>Guardar</i></li> <li>4. Se genera el banco de preguntas en base a la configuración establecida.</li> <li>5. Se traduce el banco de preguntas a formato XML.</li> </ol>
<b>Postcondición</b>	Se genera un fichero .XML en la ubicación establecida y con el nombre establecido.
<b>Excepciones</b>	No se ha introducido ningún nombre para el fichero (mensaje).
<b>Importancia</b>	Alta

Tabla B.7: CU-7 Exportar XML.





## *Apéndice C*

---

# **Especificación de diseño**

---

### **C.1. Introducción**

En este apartado se describen los detalles técnicos y funcionales clave del proyecto desarrollado. Se enfocará en tres áreas esenciales: diseño de datos, diseño procedimental y diseño arquitectónico. En el diseño de datos, se definirá y estructurará la organización de los datos utilizados en el proyecto, con diagramas para visualizar las relaciones entre ellos y su relación con las diferentes entidades en el sistema. En el diseño procedimental, se describirán los procesos y tareas del proyecto, con un diagrama de secuencia que muestra la secuencia de interacciones entre los componentes del sistema. En el diseño arquitectónico, se describirá la estructura general del software, que utiliza una arquitectura hexagonal para permitir una mayor flexibilidad, escalabilidad y modularidad.

### **C.2. Diseño de datos**

El diseño de datos es un componente clave a la hora de realizar un proyecto de software. En este apartado, se enfoca en la definición y estructuración de los datos que se utilizan en el proyecto. Para ilustrar y describir esta estructura, se incluyen diagramas que permiten visualizar la organización de los datos, las relaciones entre ellos y cómo se relacionan con las diferentes entidades en el sistema. El diseño de datos es una herramienta esencial para garantizar la eficiencia, la fiabilidad y la seguridad de los datos en el software.

[1]

La aplicación cuenta con las siguientes entidades:

- **BancoPreguntas:** tiene una lista de preguntas. Almacena todas las preguntas que se han generado.
- **Pregunta:** tiene un título, enunciado y una lista de opciones. Almacena los campos que forman una pregunta.
- **Opcion** tiene un peso y un texto. El peso es la puntuación que esa opción tiene asignada, de forma que si es verdadera es positiva, y si es falsa, negativa. Su valor depende del número de opciones verdaderas y falsas.

## Diagrama E/R

Un diagrama E/R (Entidad-Relación) es una técnica gráfica que permite representar y modelar entidades y las relaciones entre ellas en un sistema o proyecto. Un diagrama E/R se compone de dos elementos principales: entidades y relaciones. Las entidades son objetos que se consideran relevantes para el sistema o proyecto que se está modelando. Las relaciones describen cómo las entidades están vinculadas entre sí. En un diagrama E/R, las entidades se representan como rectángulos y las relaciones como líneas conectadas entre ellos. Este tipo de diagrama es útil para representar y comprender la estructura de un sistema y para planificar su desarrollo.

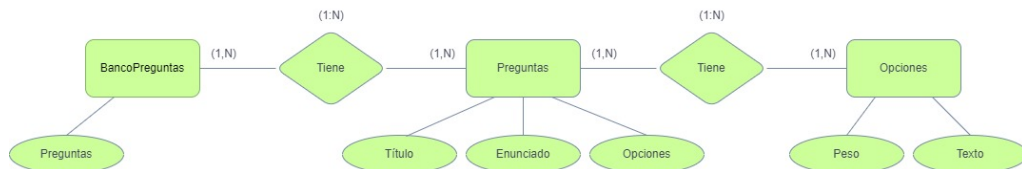


Figura C.1: Diagrama Entidad/Relación

## Estructura de la interfaz de usuario

Se va a explicar la estructura de la interfaz gráfica para la pregunta *GeneradorReglasAsociacion*. Esta estructura es la misma para todos los tipos de preguntas que la aplicación puede implementar.

La ventana principal es una clase abstracta, cuyo método es implementado por la clase *VentanaReglasAsociacion*. Esta clase instancia un objeto de tipo *BancoPreguntas*. Un banco de preguntas tiene una lista de *Preguntas*. Cada objeto de esta lista, tiene a su vez una lista de *Opciones*

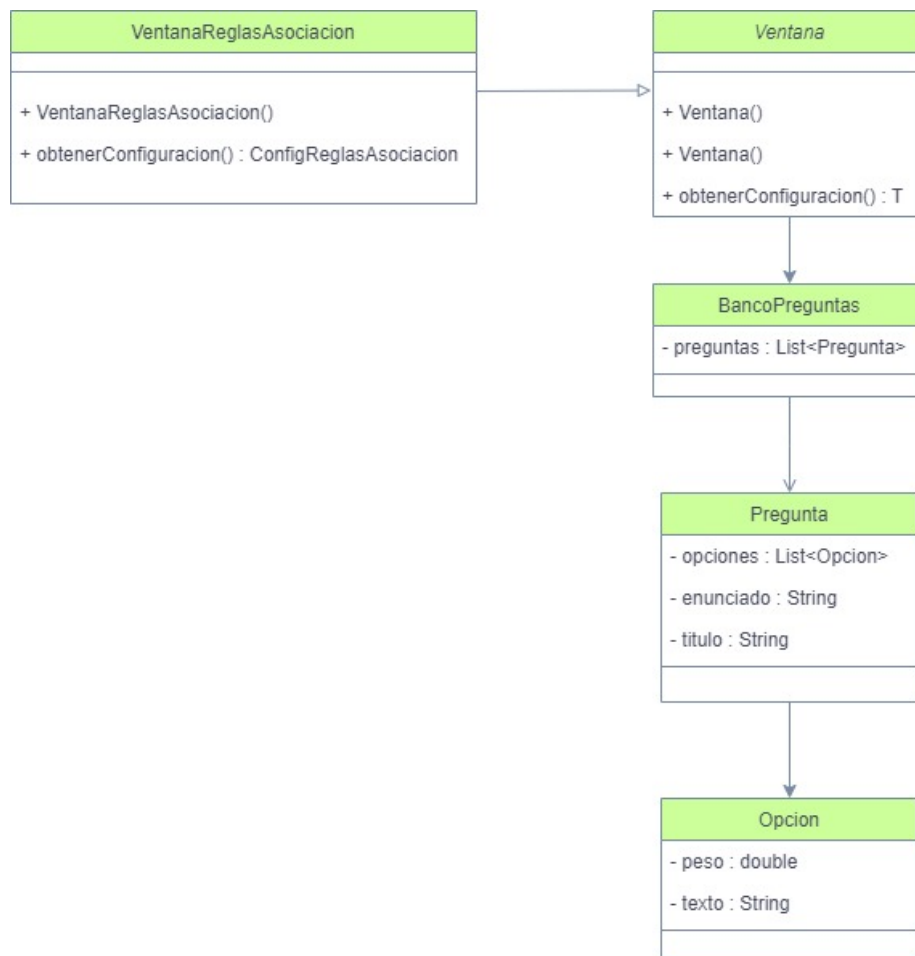


Figura C.2: Diagrama de clases de la interfaz

## Estructura del generador

Se va a explicar la estructura del generador para la pregunta *GeneradorReglasAsociacion*. Esta estructura es la misma para todos los tipos de preguntas que la aplicación puede implementar.

El generador tiene una interfaz llamada *GeneradorBancoPreguntas* que tiene un método que recibe un objeto genérico. Este método, es implementado por la clase *GeneradorBancoPreguntasReglasAsociacion*. Esta clase, además, utiliza un objeto de tipo *GeneradorConjuntoDatos* para obtener objetos de tipo *ConjuntoDatos*. La clase *ConfigReglasAsociacion* es usada por varias clases que acceden a los valores de un objeto de este tipo.

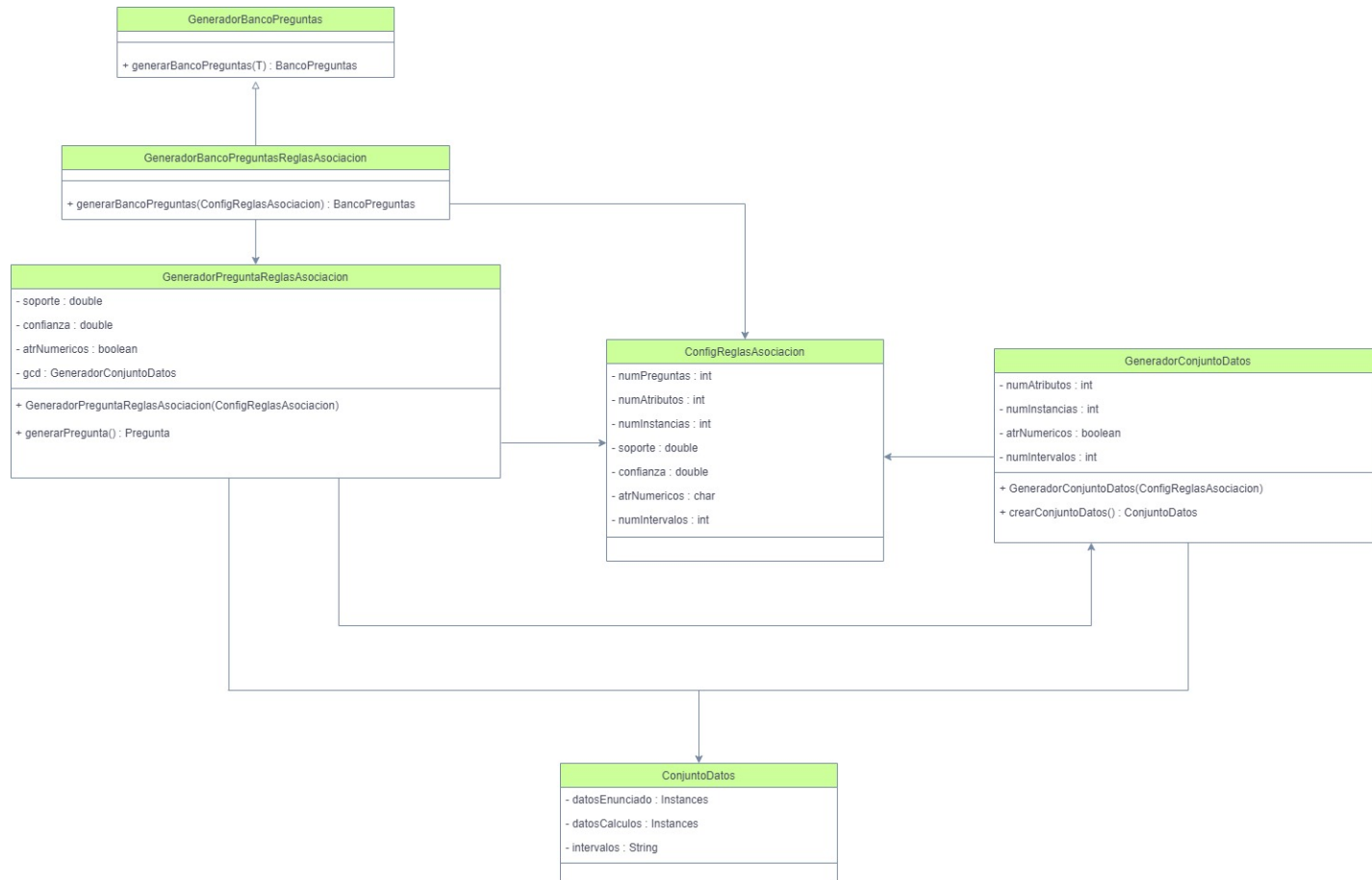


Figura C.3: Diagrama de clases del generador

## Estructura del traductor

Se va a explicar la estructura del traductor para la pregunta *GeneradorReglasAsociacion*. Esta estructura es la misma para todos los tipos de banco de preguntas que la aplicación puede traducir.

El traductor tiene una interfaz *Traductor* que tiene un método que traduce un banco de preguntas que recibe por parámetro. La clase *TraductorXML* implementa este método para traducir el banco de preguntas a un formato .XML importable en *Moodle*. Para ello, utiliza la clase *Plantilla*.

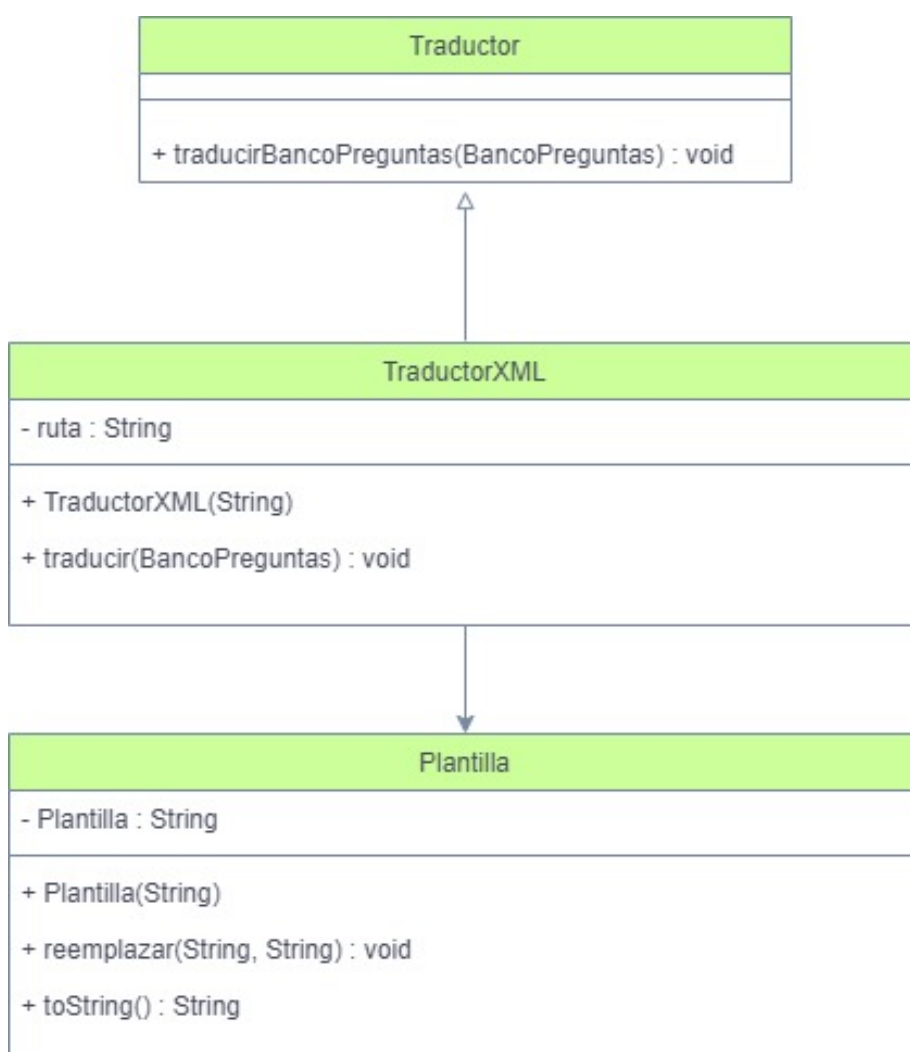


Figura C.4: Diagrama de clases del traductor

### **C.3. Diseño procedimental**

El diseño procedimental se centra en describir cómo se realizan las diferentes tareas y procesos en el proyecto. Para ilustrar estos procesos, se incluye un diagrama de secuencia, que muestra la secuencia de interacciones entre los objetos y componentes del sistema y cómo se realizan las tareas y procesos. Este diagrama de secuencia es una herramienta valiosa para entender cómo se realizarán las tareas en el software y cómo se relacionan entre ellas. Por lo tanto, es esencial para garantizar la eficiencia y la claridad en el diseño y la implementación del software. [2]

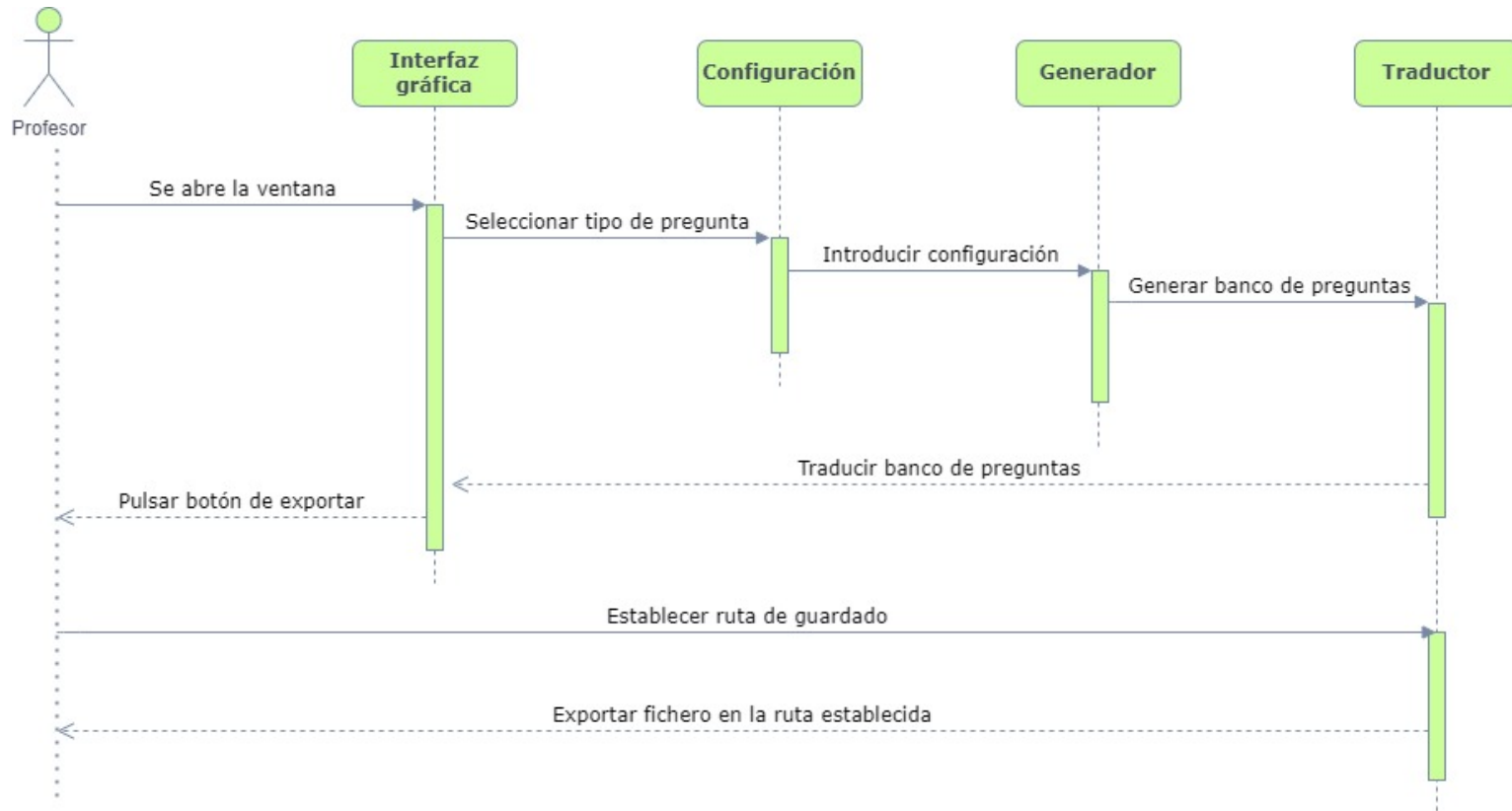


Figura C.5: Diagrama de secuencia

## C.4. Diseño arquitectónico

El diseño arquitectónico se centra en describir la estructura y organización general del software. En este caso, se ha elegido utilizar una arquitectura hexagonal.

La arquitectura hexagonal es un enfoque para diseñar aplicaciones que se centra en la separación clara de los componentes de una aplicación en diferentes capas. La idea detrás de la arquitectura hexagonal es proporcionar una estructura clara y modular que permita separar los diferentes aspectos del sistema y mejorar la escalabilidad, la mantenibilidad y la testabilidad del código. Al hacerlo, se pueden realizar cambios en una parte del sistema sin afectar a otras partes, lo que facilita el desarrollo y la mejora continua del software. [4]

Para este proyecto se ha seguido esta arquitectura al separar la interfaz gráfica del usuario, el dominio y la lógica de negocio (generación de preguntas y creación de archivos .XML) en diferentes capas.

La capa de interfaz gráfica se encarga de presentar la información al usuario y recibir sus selecciones y configuraciones. La capa de generación de preguntas se encarga de crear las preguntas basadas en las selecciones y configuraciones del usuario. Por último, la capa de creación de archivos .XML se encarga de convertir las preguntas en un formato compatible con la plataforma Moodle.



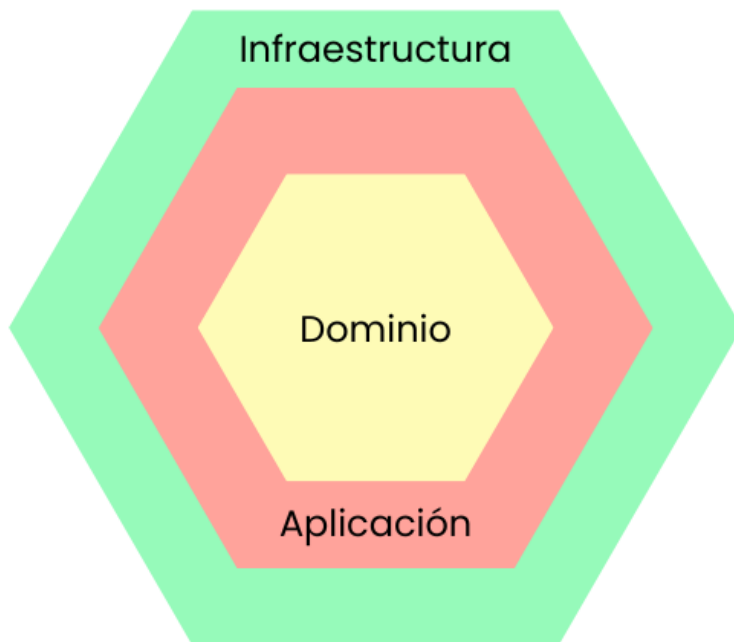


Figura C.6: Arquitectura hexagonal



## *Apéndice D*

---

# Documentación técnica de programación

---

## D.1. Introducción

La documentación técnica de programación es un elemento esencial para el correcto desarrollo y mantenimiento de un proyecto software. En este apéndice, se describen los aspectos técnicos relacionados con el proyecto, incluyendo la estructura de directorios, el manual del programador, la compilación, la instalación y ejecución del proyecto y las pruebas del sistema. Esta documentación es fundamental para garantizar la consistencia y la eficacia del software y para facilitar la tarea de los programadores y usuarios que interactúan con él. La información contenida en este apéndice es crucial para el éxito y la escalabilidad del proyecto a largo plazo.

## D.2. Estructura de directorios

El repositorio del proyecto se distribuye de la siguiente manera:

- `/`: contiene el fichero `pom.xml` de Maven, el fichero `README`, el `.gitignore` y la copia de la licencia.
- `/src/`: es el módulo correspondiente a la aplicación.
- `/src/es/ubu/inf/tfg`: contiene todas las carpetas principales del proyecto.
- `/src/es/ubu/inf/tfg/ui/`: contiene todas las clases relacionadas con la interfaz de usuario, entre las que se encuentra el `main` del programa.

- `/src/es/ubu/inf/tfg/dominio/`: contiene las clases que forman parte del dominio del proyecto.
- `/src/es/ubu/inf/tfg/generador`: contiene una interfaz común para todas las preguntas y dos subpaquetes: uno para la generación de datos y otro para la generación de preguntas.
- `/src/es/ubu/inf/tfg/generador/datos`: contiene un subpaquete por cada conjunto de datos que es necesario generar.
- `/src/es/ubu/inf/tfg/generador/datos/conjuntoitemsets`: contiene las clases necesarias para generar un conjunto de item sets.
- `/src/es/ubu/inf/tfg/generador/datos/conjuntodatos`: contiene las clases necesarias para generar un conjunto de datos.
- `/src/es/ubu/inf/tfg/generador/preguntas`: contiene un subpaquete por cada tipo de pregunta que se puede generar.
- `/src/es/ubu/inf/tfg/generador/preguntas/ampliacionitemsets`: contiene las clases necesarias para generar una pregunta de tipo "Ampliación Item Sets"
- `/src/es/ubu/inf/tfg/generador/preguntas/reglasasociaicon`: contiene las clases necesarias para generar una pregunta de tipo "Generación Reglas de Asociación"
- `/src/es/ubu/inf/tfg/generador/preguntas/itemsets`: contiene las clases necesarias para generar una pregunta de tipo "Generación Item Sets"
- `/src/es/ubu/inf/tfg/traductor`: contiene una interfaz para traducir las preguntas, una clase Plantilla y las plantillas en formato .XML necesarias para ejecutar el código desde Eclipse.
- `/src/es/ubu/inf/tfg/traductor/xml`: contiene la clase necesaria para traducir un banco de preguntas a formato .xml.
- `/resources/`: es el módulo correspondiente a las plantillas en formato .XML.
- `/resources/es/ubu/inf/tfg/traductor`: contiene las plantillas en formato .XML necesarias para ejecutar el archivo .jar.

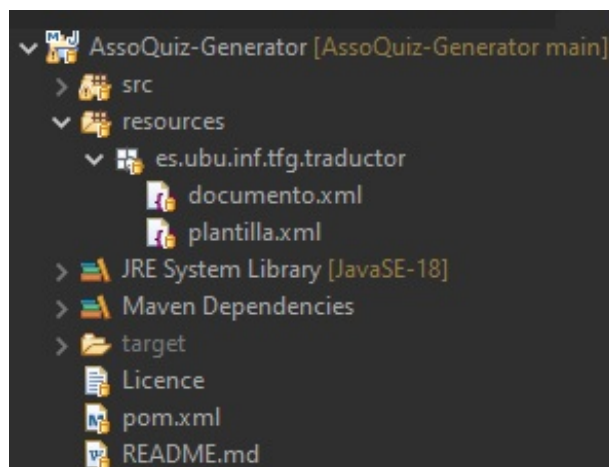


Figura D.1: Estructura de paquetes

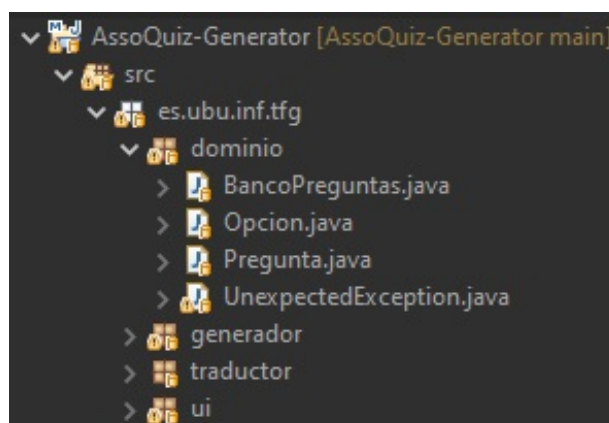


Figura D.2: Estructura de paquetes del dominio

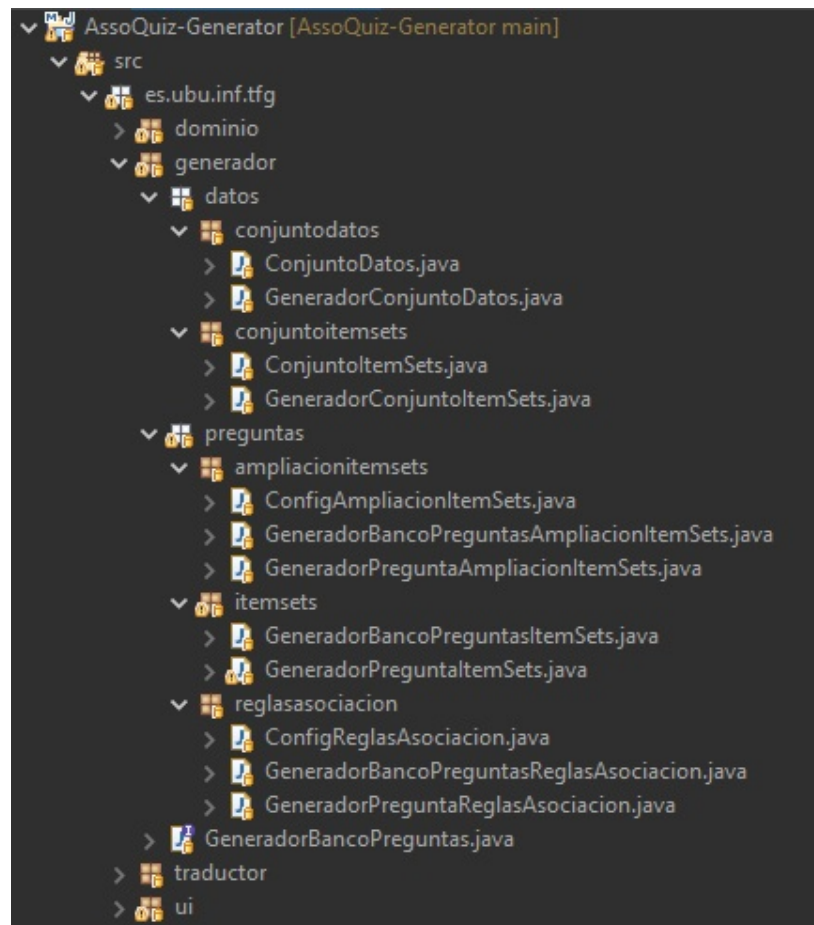


Figura D.3: Estructura de paquetes del generador

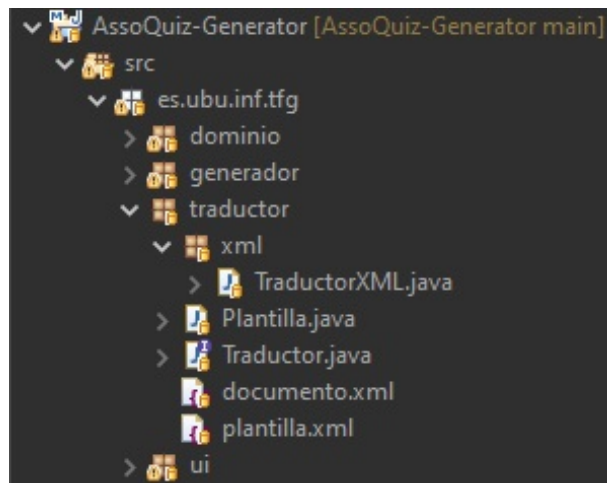


Figura D.4: Estructura de paquetes del traductor

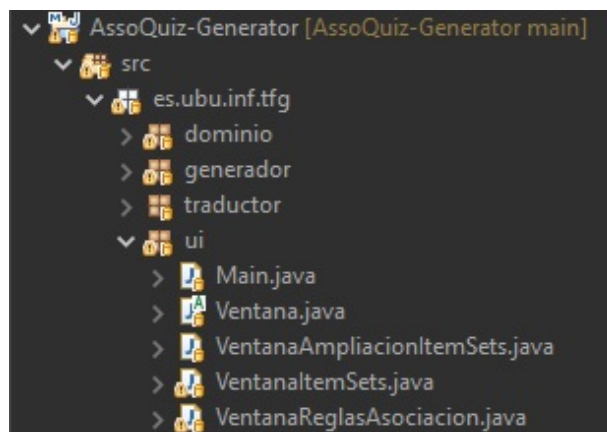


Figura D.5: Estructura de paquetes de la interfaz de usuario

## D.3. Manual del programador

Este manual está diseñado para guiar a programadores futuros que trabajen en la aplicación. Se describen los pasos para obtener las herramientas necesarias, configurar el entorno de desarrollo y añadir nuevas características a la aplicación.

### Entorno de desarrollo

Para trabajar en el proyecto se necesita tener instalado lo siguiente:

- Java JDK 19.
- Eclipse.
- Git.

A continuación, se indica como instalar y configurar correctamente cada uno de ellos.

## Java JDK 19

Java es uno de los lenguajes de programación más utilizados. Se puede descargar la última versión desde <https://www.oracle.com/es/java/technologies/downloads/>. Es importante seleccionar adecuadamente el sistema operativo y la arquitectura del ordenador. [7]

En este caso se va a explicar el proceso de instalación en Windows, ya que es el sistema operativo que se ha utilizado para este proyecto:

1. Ejecutar el archivo de instalación y seguir las instrucciones en pantalla para completar la instalación.
2. Verificar que la instalación de Java se haya realizado correctamente abriendo el símbolo del sistema y escribiendo *java -version*.
3. Hacer clic en el botón *Inicio* y escribir *variables de entorno* en el cuadro de búsqueda.
4. Seleccionar *Editar las variables de entorno del sistema* en los resultados de la búsqueda.
5. En la ventana *Configuración avanzada del sistema*, hacer clic en el botón *Variables de entorno*.
6. En la sección *Variables del sistema*, buscar la variable PATH y hacer clic en *Editar*.
7. Agregar la ruta de la carpeta *lib* de Java a la variable PATH, asegurándose de separar las entradas con un punto y coma. Por ejemplo, si Java está instalado en `C:\Program Files\Java\jdk-19`, agregar `;C:\Program Files\Java\jdk-19\lib` al final de la variable PATH.
8. Hacer clic en *Aceptar* para guardar los cambios.



## Eclipse

Para descargar Eclipse en Windows se han seguido los siguientes pasos:

1. Descargar el archivo de instalación desde <https://www.eclipse.org/downloads/>.
2. Descomprimir el archivo de instalación en la carpeta deseada.
3. Hacer clic en el archivo *eclipse.exe* para iniciarlo.
4. Seleccionar una carpeta de *Workspace* qué es donde se guardarán los proyectos.

## Git

Pra descargar Git en Windows se han seguido los siguientes pasos:

1. Descargar el archivo de instalación desde <https://git-scm.com/downloads>.
2. Ejecutar el archivo de instalación y seguir las instrucciones indicadas.
3. Verificar la instalación abriendo el símbolo del sistema y escribiendo *git --version*. Se tiene que obtener una respuesta que indique la versión de Git que se ha instalado.

La instalación de Git también incluye una interfaz gráfica de usuario, como Git Bash o Git GUI, que permite interactuar con Git de manera más amigable.

## Añadir nuevas características a la aplicación

Una vez se ha importado el proyecto en Eclipse, ya es posible añadir nuevas características.

### Añadir nuevas preguntas

Para ello hay que seguir los siguientes pasos:

1. Crear un nuevo paquete dentro de */src/es/ubu/inf/tfg/generador/preguntas* que se llame como el nombre de la nueva pregunta.
2. Dentro de este paquete, crear, al menos, dos clases:

- Un generador de banco de preguntas: esta clase tiene que implementar la interfaz *GeneradorBancoPreguntas* y tiene que sobrescribir su método *generarBancoPreguntas(T config)* pasándole la configuración de la nueva pregunta.
  - Un generador de pregunta: esta clase tiene que tener un método *generarPregunta()* que devuelva una pregunta nueva. Para ello se tendrán que crear, además, métodos que establezcan su enunciado, título y opciones.
  - Una configuración de pregunta: esta clase es opcional, puesto que puede que una nueva pregunta utilice un conjunto de datos común para varias preguntas, por lo que no sería necesario crear una configuración nueva. Esta clase tiene que iniciar en el constructor los valores necesarios para generar el conjunto de datos que vaya a usar.
3. Crear una nueva clase dentro de `/src/es/ubu/inf/tfg/ui` que se corresponda con la ventana de la nueva pregunta en la interfaz gráfica. Además de eso, editar el *Main* para añadir un nuevo botón que llame a la nueva clase.

### Añadir nuevo traductor

Estos son los pasos que hay que seguir en caso de que se quiera añadir un nuevo traductor para traducir el banco de preguntas a otro lenguaje:

1. Crear un nuevo paquete dentro de `/src/es/ubu/inf/tfg/traductor` que se llame como el lenguaje al que se va a traducir.
2. Dentro de este paquete, crear una nueva clase que implemente la interfaz *Traductor* y sobrescriba el método *traducir(BancoPreguntas bancoPreguntas)*. Este método tiene que hacer lo necesario para traducir el banco de preguntas que recibe al lenguaje deseado.
3. Añadir dentro del paquete `/src/es/ubu/inf/tfg/traductor` las plantillas en el formato del lenguaje que se desea traducir. Estas plantillas tienen que ser utilizadas por la nueva clase para traducir las preguntas.

Por último, es necesario subir las actualizaciones al repositorio de GitHub. Para ello, hay que hacer lo siguiente en la terminal:

1. Situar en el directorio del proyecto mediante el comando `cd` seguido de la ruta del directorio.

2. Añadir al *stage* los archivos modificados mediante *git add*.
3. Hacer *commit* mediante *git commit -m "nombre que se quiera dar al commit"*.
4. Subir los cambios a la rama principal del repositorio mediante *git push*.

## Actualizar dependencias

Es importante que las dependencias utilizadas en el proyecto estén actualizadas para el correcto mantenimiento del mismo.

En este proyecto se utiliza Maven como herramienta para gestionar las dependencias externas. Las dependencias se definen dentro del fichero *pom.xml*.

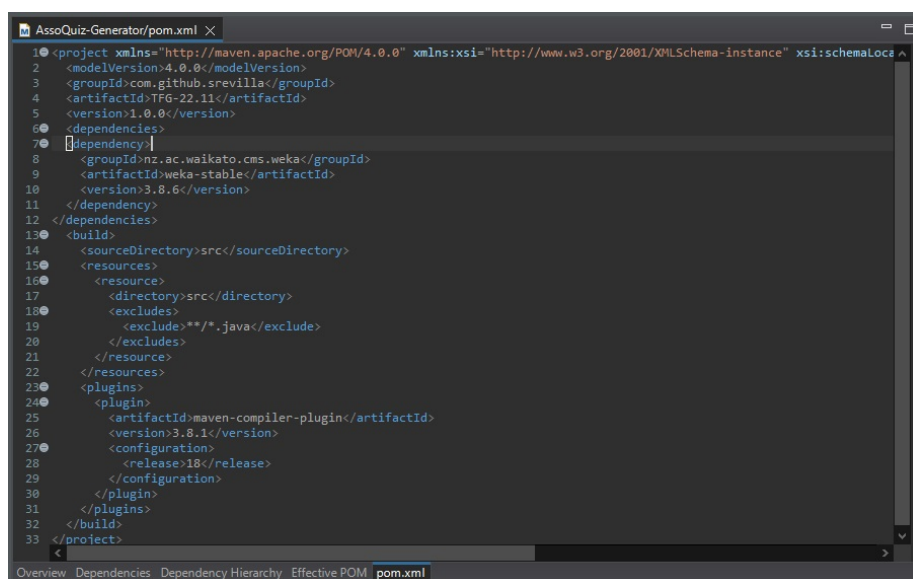


Figura D.6: Contenido del fichero *pom.xml*

## D.4. Compilación, instalación y ejecución del proyecto

En esta sección se describen los pasos para obtener el código fuente del proyecto, compilarlo, ejecutarlo y exportarlo.

## Obtención del código fuente

Para el desarrollo del proyecto se ha utilizado un repositorio Git alojado en GitHub. Para obtener una copia del repositorio hay que seguir los siguientes pasos:

1. Abrir la terminal de Git Bash.
2. Colocarse en el directorio donde se desee copiar el repositorio (mediante el comando `cd`).
3. Introducir el siguiente comando:  

```
git clone https://github.com/srevilla/TFG-22.11.git
```
4. Cuando finalice la descarga ya estará disponible una copia completa del repositorio.

## Importar proyecto en Eclipse

Para importar un proyecto en Eclipse, hay que seguir estos pasos:

1. Abrir Eclipse y seleccionar *File -> Import*.
2. Seleccionar la opción *Existing Projects into Workspace* y hacer clic en *Next*.
3. Seleccionar la ubicación del proyecto que se desea importar.
4. Verificar que el proyecto aparece en la lista de proyectos seleccionados y hacer clic en *Finish*.

Una vez hecho esto, el proyecto estará disponible en el árbol de proyectos de Eclipse y puede ser abierto y modificado desde allí.

## Compilación del código

Para compilar el proyecto en Eclipse, es necesario seguir estos pasos:

1. Hacer clic en el botón *Project* en la barra de herramientas y seleccionar *Build Project* en el menú desplegable.
2. Alternativamente, se puede presionar *Ctrl + B* en el teclado.

De esta forma, Eclipse compila el código fuente y muestra cualquier error o advertencia en la pestaña *Problems*. Si no hay errores, se crea automáticamente un archivo compilado o ejecutable en la carpeta *bin* del proyecto.

## Ejecución del proyecto

Para ejecutar el código en Eclipse, es necesario seguir estos pasos:

1. Hacer clic derecho en el archivo principal del proyecto (en este caso, la clase *Main.java* situada en */src/es/ubu/inf/tfg/ui/*) y seleccionar *Run As* en el menú desplegable.
2. Seleccionar *Java Application* en el menú desplegable.

De esta forma, Eclipse compila y ejecuta el código, y se abre una ventana con el menú principal de la aplicación.

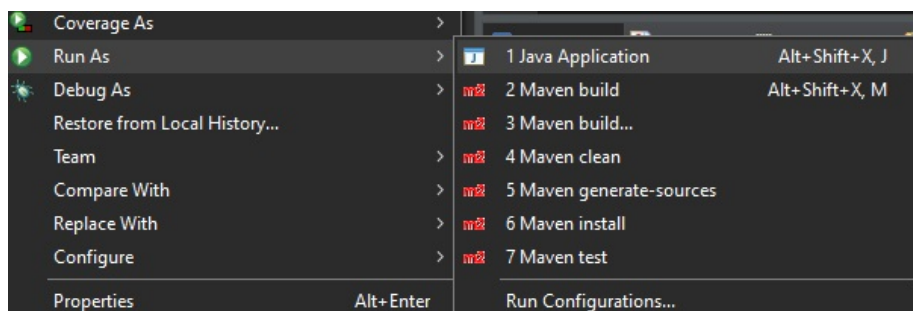


Figura D.7: Ejecución del proyecto en Eclipse

## Exportar aplicación

Para crear un archivo ejecutable del proyecto es necesario seguir estos pasos:

1. Hacer clic derecho en el proyecto en el panel *Project Explorer* y seleccionar *Export* en el menú desplegable.
2. Seleccionar *Java* en la categoría *Export* y luego *JAR file*.
3. Seguir las instrucciones en el asistente de exportación para especificar la ubicación y las opciones del archivo JAR.

4. Una vez que se ha creado el archivo JAR, ejecutarlo desde la línea de comandos utilizando el comando `"java -jar nombre.jar"`.

## D.5. Pruebas del sistema

Se ha comprobado manualmente el funcionamiento del programa. Se han utilizado diferentes casos de prueba para cada uno de los tipos de preguntas y se ha verificado que el programa proporciona resultados correctos y precisos. Aunque no se han implementado pruebas unitarias, se está seguro de que el programa cumple con los requisitos y estándares necesarios para ser utilizado eficientemente.

## *Apéndice E*

---

# Documentación de usuario

---

### E.1. Introducción

El manual de usuario es un componente fundamental de cualquier proyecto software, ya que proporciona la información necesaria para que los usuarios puedan utilizar el producto de manera efectiva. Este apéndice incluye tres secciones importantes: Requisitos de usuarios, Instalación y Manual del usuario.

### E.2. Requisitos de usuarios

En esta sección se describen todos los requisitos técnicos y de hardware necesarios para utilizar el programa de manera óptima. Aquí se incluyen los requisitos mínimos y recomendados de hardware y software, así como cualquier otra información relevante para asegurarse de que el programa funcione correctamente en el sistema del usuario.

Los requisitos mínimos son:

- Sistema operativo: Se requiere un sistema operativo compatible con Java, como Windows, MacOS o Linux.
- Java: Se requiere tener instalada la versión mínima de Java SE (Standard Edition) necesaria para ejecutar el programa. La versión recomendada es Java SE 19 ya que es la más reciente.

- Espacio en disco: Se requiere una cantidad mínima de espacio en disco para almacenar el programa y cualquier otro archivo necesario para su ejecución.
- Memoria RAM: Se requiere una cantidad mínima de memoria RAM para ejecutar el programa de manera óptima.

### E.3. Instalación

En esta sección se explica como instalar y configurar el programa en el sistema del usuario.

Para ejecutar *AssoQuiz-Generator* se dispone de un fichero .jar ejecutable. El programa se iniciará al ejecutar este fichero.



Figura E.1: Icono *AssoQuiz-Generator*

### E.4. Manual del usuario

En esta sección se proporciona una descripción detallada de cómo utilizar el programa, Aquí se incluyen capturas de pantalla y ejemplos prácticos para ilustrar cómo utilizar el programa de manera efectiva.

Al ejecutar *AssoQuiz-Generator* aparece una ventana con los tipos de preguntas disponibles.



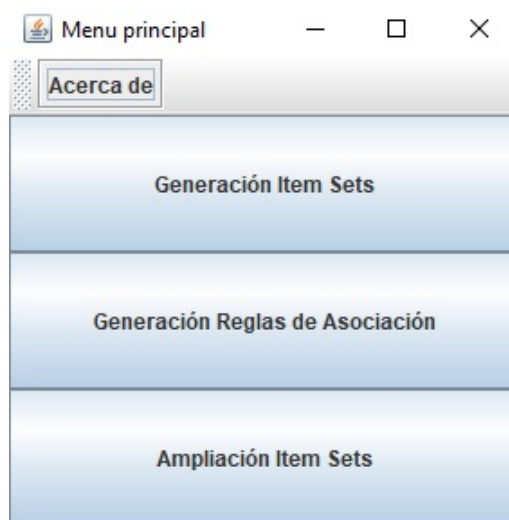
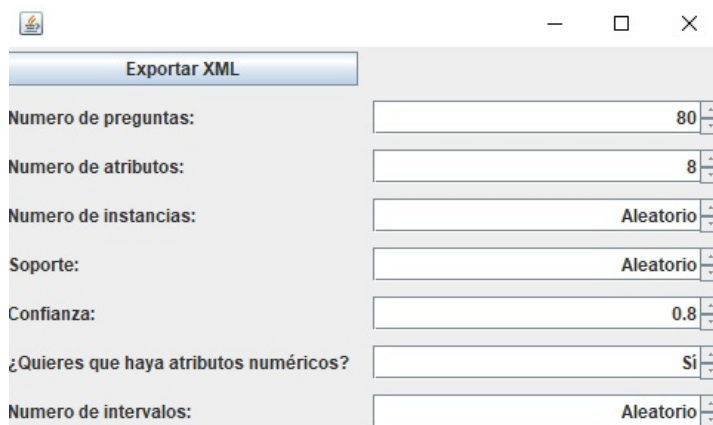


Figura E.2: Ventana principal

Si se pulsa en una opción, se abre una ventana en la que se da opción a configurar un banco de preguntas. Se pueden seleccionar la configuración de los parámetros entre unos intervalos establecidos previamente en el código. También da la opción de generar aleatoriamente estos parámetros para cada pregunta generada.

Figura E.3: Ventana configuración *Generación Reglas Asociación*

En esta ventana hay un botón de exportar, que al pulsarlo se abre un explorador en el que se tiene que elegir una carpeta para guardar el archivo

.XML que se va a generar. Es necesario introducir un nombre para el archivo, de lo contrario, el programa no deja guardarlo.

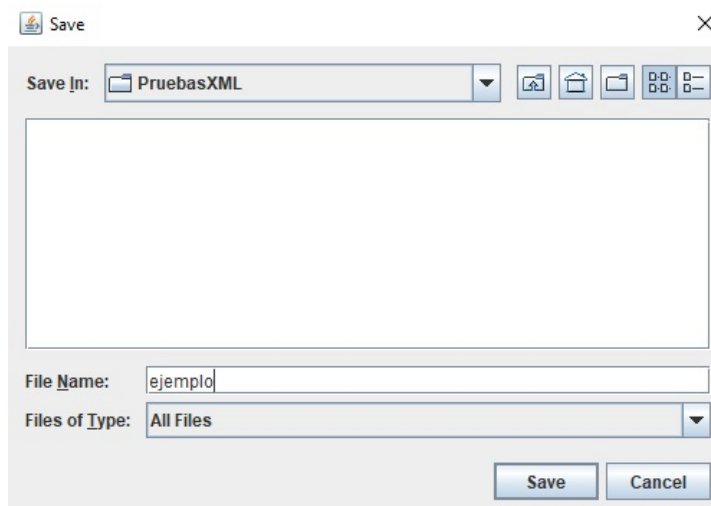


Figura E.4: Ventana de guardado de un archivo

Mientras se genera el banco de preguntas, sale un mensaje avisando de que se está generando. Cuando ya ha terminado, se avisa al usuario mediante esta ventana:

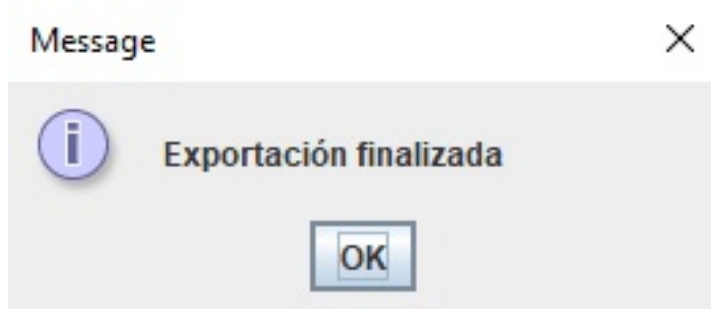


Figura E.5: Ventana finalizado

Una vez hecho esto, el fichero está listo para ser importado en *Moodle*.

Para ello es necesario poder acceder a la plataforma de aprendizaje de *Moodle*. Los pasos que se deben seguir son:

1. Ir a la página principal del curso.

2. Pulsar en *Banco de preguntas* desde el bloque de Administración.
3. Pulsar en *Importar archivos*.
4. Seleccionar la opción de *formato XML Moodle*.
5. Desde la siguiente ventana pulsar en *Examinar*.
6. Seleccionar el fichero .XML desde el ordenador.
7. Pulsar en *Importar* este archivo.

Los puntos 5 y 6 se pueden realizar arrastrando el archivo al recuadro que se muestra.

**Pregunta 1**  
Respuesta guardada  
Puntúa como 1,00

Sea el siguiente conjunto de datos:

	A	B	C	D	E	F
1	T	F	F	T	F	T
2	F	F	T	T	F	F
3	F	T	F	F	T	T
4	F	F	F	F	F	T
5	F	T	F	T	T	F
6	T	F	F	F	T	F
7	F	T	F	T	F	T
8	F	F	F	T	F	F
9	T	F	F	F	F	F
10	T	F	T	F	F	F

Seleccione los item sets con soporte mayor o igual que 3.

Seleccione una o más de una:

- ☒ a. [B=F, D=F, F=F]
- ☐ b. [B=T, D=T]
- ☒ c. [A=F, C=F]
- ☐ d. [B=T, E=T]

Figura E.6: Vista de la pregunta *Generación Item Sets* resuelta en *Moodle*

**Pregunta 1**  
Correcta  
Se puntúa 1,00 sobre 1,00

Sea el siguiente conjunto de datos:

ABCDEFGHIX	
T T T F T F F T	42.8
F T F F F F T T	73.7
T T F F T T F F	94.7
T F T T T F F T	45.9
F F T T T F F F	63.0
T F F T T T F F	60.9
T T T T F T T T	10.4
F F T F F T T F	69.7
F F F T T F T T	67.3
F T F T F F F T	21.5
T T T F F F F F	84.1
F F T T T F T T	98.2

Seleccione las reglas de asociacion con soporte y confianza mayores o iguales que, respectivamente, 3 y 80%. El atributo X debe discretizarse entre los siguientes intervalos: (- ; 39.7], [39.8 ; 68.9], [69.0 ; -)

Seleccione una o más de una:

- ☒ a. [B=F, E=T] ==> [D=T] ✓
- ☒ b. [A=T] ==> [G=F] ✓
- ☐ c. [E=T] ==> [X='(39.7-68.9)']
- ☐ d. [D=T] ==> [E=T]

Figura E.7: Vista de la pregunta *Generación Reglas Asociación* resuelta en Moodle

**Pregunta 1**  
Respuesta guardada  
Puntúa como 1,00

Se dispone de los siguientes 3-item sets: [[A, B, G], [A, C, D], [A, C, F], [A, C, G], [A, D, F], [A, D, G], [A, E, F], [B, D, F], [C, D, F], [C, E, F], [C, F, G], [D, E, F]] ¿Cuales son los posibles 4-item sets?

Seleccione una o más de una:

- ☒ a. [A, C, D, F]
- ☐ b. [A, E, F, G]
- ☐ c. [A, B, C, G]
- ☐ d. [B, C, F, G]

Figura E.8: Vista de la pregunta *Ampliación Item Sets* resuelta en *Moodle*



---

## Bibliografía

---

- [1] Diseño de datos. [Online; accedido 07-febrero-2023].
- [2] Diseño procedimental. [Online; accedido 09-febrero-2023].
- [3] GNU. Licencias gnu, 2022. [Online; accedido 11-febrero-2023].
- [4] Edu Salguero. Arquitectura hexagonal, 2018. [Online; accedido 09-febrero-2023].
- [5] Open Source. The legal side of open source. [Online; accedido 10-febrero-2023].
- [6] Wikipedia. Especificación de requisitos, 2022. [Online; accedido 07-febrero-2023].
- [7] Wikipedia. Java jdk, 2022. [Online; accedido 05-febrero-2023].
- [8] Wikiversity. Plan de proyecto software, 2014. [Online; accedido 09-febrero-2023].