



# UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO  
INGENIERÍA INFORMÁTICA

## Software de apoyo para la evaluación basado en competencias

**Autor**

Rafael Delgado García-Valdecasas

**Tutores**

Daniel Sánchez Fernández

Nicolás Marín Ruiz



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

---

Granada, 24 de junio de 2024

# Índice

Índice	1
Índice de Tablas	2
Índice de figuras	2
<b>1 Introducción</b>	<b>10</b>
1.1 Motivación . . . . .	10
1.2 Objetivos . . . . .	11
1.2.1 Creación de una jerarquía de aprendizaje con preguntas. . . . .	12
1.2.2 Generación de exámenes basados en competencias . . . . .	15
1.2.3 Visualización de resultados . . . . .	15
1.3 Protección de datos personales . . . . .	16
1.4 Estudio de soluciones existentes . . . . .	16
1.4.1 Moodle <sup>™</sup> . . . . .	16
1.4.2 Otras soluciones . . . . .	18
<b>2 Solución propuesta</b>	<b>21</b>
2.1 Metadatos asociados a cada entidad . . . . .	21
2.2 Formatos de exportación e importación de archivos . . . . .	22
2.3 Requisitos Funcionales . . . . .	24
2.4 Modelo de Datos . . . . .	28
2.4.1 Diccionario de Datos . . . . .	29
2.5 Casos de Uso . . . . .	32
2.6 Implementación . . . . .	81
2.6.1 Liberación del software . . . . .	85
2.6.2 Selección de preguntas de exámenes . . . . .	86
2.7 Estimación de Recursos . . . . .	89
<b>3 Manual de usuario</b>	<b>91</b>
3.1 Manual de Instalación y Despliegue en Linux . . . . .	91
3.1.1 Manual de Instalación en Linux . . . . .	91
3.1.2 Manual de Despliegue en Linux . . . . .	92
3.2 Manual de Instalación y Despliegue en Windows . . . . .	94
3.2.1 Manual de Instalación en Windows . . . . .	94
3.2.2 Manual de Despliegue en Windows . . . . .	95
3.3 Manual de Uso . . . . .	96
<b>4 Conclusiones</b>	<b>111</b>
4.1 Posibilidades futuras . . . . .	113
<b>5 Bibliografía</b>	<b>115</b>

## Índice de Tablas

1	Tabla de configuraciones de valores de parámetros de pregunta parametrizada. . . . .	15
2	Tabla de comparación de soluciones existentes. . . . .	19
3	Tabla de Requisitos Funcionales . . . . .	25
4	CU-1: 'Registrarse' . . . . .	33
5	CU-2: 'Iniciar Sesión'. . . . .	35
6	CU-3: 'Cerrar Sesión'. . . . .	37
7	CU-4: 'Crear Asignatura'. . . . .	38
8	CU-5: 'Editar Asignatura'. . . . .	40
9	CU-6: 'Eliminar Asignatura'. . . . .	42
10	CU-7: 'Listar Asignaturas'. . . . .	44
11	CU-8: 'Ver Asignatura'. . . . .	45
12	CU-9: 'Crear Nodo'. . . . .	46
13	CU-10: 'Editar Nodo'. . . . .	48
14	CU-11: 'Eliminar Nodo'. . . . .	50
15	CU-12: 'Listar Nodos'. . . . .	52
16	CU-13: 'Ver Nodo'. . . . .	53
17	CU-14: 'Crear Pregunta'. . . . .	54
18	CU-15: 'Editar Pregunta'. . . . .	56
19	CU-16: 'Eliminar Pregunta'. . . . .	58
20	CU-17: 'Listar Preguntas'. . . . .	60
21	CU-18: 'Ver Pregunta'. . . . .	61
22	CU-19: 'Importar Banco de Preguntas'. . . . .	62
23	CU-20: 'Desactivar Pregunta'. . . . .	64
24	CU-21: 'Crear Examen'. . . . .	66
25	CU-22: 'Editar Examen'. . . . .	68
26	CU-23: 'Eliminar Examen'. . . . .	70
27	CU-24: 'Listar Exámenes'. . . . .	72
28	CU-25: 'Ver Examen'. . . . .	73
29	CU-26: 'Exportar Examen'. . . . .	74
30	CU-27: 'Importar Resultados'. . . . .	76
31	CU-28: 'Eliminar Resultados'. . . . .	78
32	CU-29: 'Listar Resultados'. . . . .	80

## Índice de figuras

1	Diagrama E/R de la Base de datos de la aplicación. . . . .	29
2	Diagrama de Interacción de CU-1: 'Registrarse' . . . . .	34
3	Diagrama de Interacción de CU-2: 'Iniciar Sesión'. . . . .	36
4	Diagrama de Interacción de CU-3: 'Cerrar Sesión'. . . . .	37
5	Diagrama de Interacción de CU-4: 'Crear Asignatura'. . . . .	39

6	Diagrama de Interacción de CU-5: 'Editar Asignatura'.	41
7	Diagrama de Interacción de CU-6: 'Eliminar Asignatura'.	43
8	Diagrama de Interacción de CU-7: 'Listar Asignaturas'.	44
9	Diagrama de Interacción de CU-8: 'Ver Asignatura'.	45
10	Diagrama de Interacción de CU-9: 'Crear Nodo'.	47
11	Diagrama de Interacción de CU-10: 'Editar Nodo'.	49
12	Diagrama de Interacción de CU-11: 'Eliminar Nodo'.	51
13	Diagrama de Interacción de CU-12: 'Listar Nodos'.	52
14	Diagrama de Interacción de CU-13: 'Ver Nodo'.	53
15	Diagrama de Interacción de CU-14: 'Crear Pregunta'.	55
16	Diagrama de Interacción de CU-15: 'Editar Pregunta'.	57
17	Diagrama de Interacción de CU-16: 'Eliminar Pregunta'.	59
18	Diagrama de Interacción de CU-17: 'Listar Preguntas'.	60
19	Diagrama de Interacción de CU-18: 'Ver Pregunta'.	61
20	Diagrama de Interacción de CU-19: 'Importar Banco de preguntas'.	63
21	Diagrama de Interacción de CU-20: 'Desactivar Pregunta'.	65
22	Diagrama de Interacción de CU-21: 'Crear Examen'.	67
23	Diagrama de Interacción de CU-22: 'Editar Examen'.	69
24	Diagrama de Interacción de CU-23: 'Eliminar Examen'.	71
25	Diagrama de Interacción de CU-24: 'Listar Exámenes'.	72
26	Diagrama de Interacción de CU-25: 'Ver Examen'.	73
27	Diagrama de Interacción de CU-26: 'Exportar Examen'.	75
28	Diagrama de Interacción de CU-27: 'Importar Resultados'.	77
29	Diagrama de Interacción de CU-28: 'Eliminar Resultados'.	79
30	Diagrama de Interacción de CU-29: 'Listar Resultados'.	80
31	Diagrama de Componentes de la aplicación web.	81
32	Diagrama de Flujo de la función de selección de preguntas.	88
33	Repositorio de GitHub en el que se aloja el proyecto.	92
34	Ejemplo del contenido del archivo ' <i>secret.py</i> '.	93
35	Captura de la pantalla de inicio de la aplicación.	94
36	Captura de la pantalla de registro de usuario.	96
37	Captura de la pantalla de inicio de sesión.	97
38	Captura de la pantalla principal.	97
39	Captura de la pantalla de añadir asignatura.	98
40	Captura de la pantalla de detalles de asignatura.	98
41	Ventana de editar asignatura.	99
42	Captura de la ventana de eliminar asignatura.	100
43	Captura de la pantalla de listado de nodos.	100
44	Ventana de añadir nodo.	101
45	Vista de detalles de nodo.	101
46	Ventana de editar nodo.	102
47	Ventana de eliminar nodo.	102
48	Captura de la pantalla de listado de preguntas.	103

---

49	Captura de la vista de creación de preguntas. . . . .	104
50	Captura de la pantalla de detalles/edición de una pregunta. . . . .	105
51	Captura de la ventana de desactivación de una pregunta. . . . .	105
52	Captura de la vista del listado de exámenes. . . . .	106
53	Captura de la vista de configuración de examen. . . . .	107
54	Captura de la vista de configuración de examen con un aviso de pregunta repetida. . . . .	107
55	Captura de la ventana de eliminar examen. . . . .	108
56	Captura de la pantalla de detalles de un examen. . . . .	108
57	Captura de la ventana de exportación de examen. . . . .	109
58	Captura de la ventana de importación de resultados. . . . .	109
59	Captura de la vista de listado de resultados. . . . .	110
60	Captura de la ventana de eliminar resultados. . . . .	110

# Software de apoyo para la evaluación basado en competencias

Rafael Delgado García-Valdecasas

**Palabras clave:** evaluación por competencias, banco de preguntas, generación de exámenes, estructura jerárquica de preguntas, asociación de preguntas a competencias, visualización de resultados de aprendizaje.

## Resumen

La formación ha vivido en constante evolución desde sus inicios, siendo la manera de evaluar más utilizada el uso de exámenes. Para continuar con la evolución de los métodos del aprendizaje y dar apoyo a este medio tan utilizado, el objetivo de este proyecto será la creación de un software para la gestión de bancos de preguntas sobre una asignatura desglosada de manera jerárquica, para la posterior configuración de exámenes en base a la jerarquía de aprendizaje de la asignatura. Dicha configuración permitirá la especificación de distintos parámetros, desde los ítems a evaluar o el número de preguntas hasta, por ejemplo, la repetición de preguntas en exámenes anteriores o la dificultad de las preguntas que lo conforman, con el objeto de ajustar el examen a las necesidades tanto de los docentes como de los examinados.

# **Support Software for Competence-based Evaluation.**

Rafael Delgado García-Valdecasas

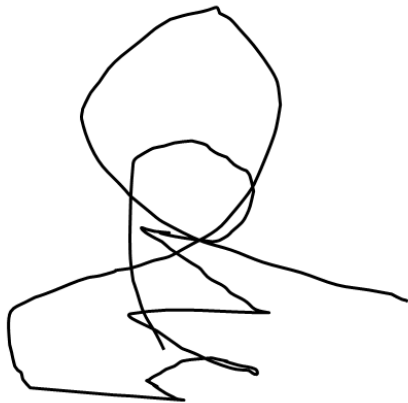
**Keywords:** competence evaluation, question bank, exam configuration, hierarchically structured questions, questions associated to competences, learning results visualization.

## **Abstract**

Learning has lived in constant evolution since its origins, counting with exams as the most used means of evaluation. To further expand the evolution of learning methods and support these common means, the objective of this project will be the creation of a software to manage question banks related to a subject that can be organized hierarchically, to later configure exams based on the subject learning hierarchy. Said configuration will allow the specifying of different parameters, ranging from the items of the hierarchy to evaluate or the number of questions in the exam to, for example, the repetition of questions in previous exams or the estimated difficulty of the questions that compose it, all with the objective of adjusting the exam to both the examiners and the examinees' needs.

---

Yo, **Rafael Delgado García-Valdecasas**, alumno de la titulación Grado de Ingeniería de Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 76653267V, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

A handwritten signature in black ink, consisting of several overlapping loops and a long horizontal stroke extending to the right.

Fdo: Rafael Delgado García-Valdecasas

Granada a 24 de junio de 2024.



---

D. **Nicolás Marín Ruiz**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

D. **Daniel Sánchez Fernández**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado ***Software de apoyo para la evaluación basado en competencias***, ha sido realizado bajo su supervisión por **Rafael Delgado García-Valdecasas**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe.

**Los tutores:**

**Nicolás Marín Ruiz      Daniel Sánchez Fernández**

## **Agradecimientos**

A mi madre, por enseñarme la valentía de sacrificar todo por lo que quieres.

A mis hermanos, tíos y primos, por enseñarme elpreciado valor de la palabra 'familia'.

A mis amigos de siempre, por ser sin dudar el apoyo incondicional cuando hacía falta.

A mis amigos de la universidad, por ser los mejores compañeros de risas y lágrimas en este viaje.

A mis tutores de este proyecto, por sus consejos y su infinita paciencia conmigo.

A mi padre, que espero que esté orgulloso allá donde me esté viendo.

## 1. Introducción

En el ámbito educativo, la evaluación de los conocimientos adquiridos es un proceso crucial para medir el nivel de aprendizaje de las competencias adquiridas o los conocimientos enseñados. Hasta ahora, el método con el que se lleva a cabo esta evaluación es la realización tradicional de exámenes. No obstante, el diseño de exámenes es un proceso que puede resultar laborioso y suponer demasiado tiempo y recursos, en concreto, para conseguir una evaluación adaptada a las circunstancias de la asignatura y de los examinados. Por todo esto, surge la necesidad de disponer de herramientas que faciliten la creación y administración de asignaturas y exámenes, con el objeto de optimizar el trabajo de los docentes y que su evaluación sea más relevante y precisa.

### 1.1. Motivación

En un primer momento, se hizo un estudio de las soluciones existentes que se desarrollará más adelante. Las herramientas encontradas tienen una gran variedad de capacidades, desde algunas más sencillas que permiten generar preguntas básicas hasta otras más complejas que posibilitan la configuración de exámenes mediante técnicas de Inteligencia Artificial. Al contar con una visión general del panorama de evaluación educativa, se pueden establecer una serie de características comunes a la mayoría de las soluciones existentes, además de características propias de una herramienta para asistir en la creación de exámenes:

- Facilidad para usuarios no expertos.
- Preguntas con distintos atributos.
- Parametrización de preguntas.
- Puntuación efectiva de las preguntas.
- Exportación o importación de exámenes.
- Generación o sugerencia de preguntas de examen.
- Visualización de resultados.

Estas características serán los pilares fundamentales del proyecto; no obstante, es preciso destacar dos ideas:

- Por una parte, la gran mayoría de herramientas encontradas están orientadas a la realización de exámenes en la misma plataforma, resultando en una experiencia muy encapsulada en la misma plataforma y sin mucha conexión con otras plataformas. La realización de exámenes no será un punto en el proyecto a desarrollar.
- Por otra parte, ninguna de las herramientas contaban con una manera de relacionar las preguntas creadas a distintas competencias o contenidos de la asignatura en cuestión. Esta funcionalidad resulta muy importante, y una nueva plataforma con esta característica puede marcar una gran diferencia con respecto a la oferta actual.

La segunda idea será el eje central de la plataforma a desarrollar, no solo por su novedad, sino por su utilidad: relacionar cada pregunta con distintos ítems de la asignatura puede resultar muy útil para realizar exámenes más ajustados a las necesidades de los docentes.

## 1.2. Objetivos

El proyecto se enfoca en desarrollar una plataforma que permita la generación de exámenes basados en un conjunto organizado de ítems de aprendizaje, estructurado a partir de los conocimientos, competencias y relaciones de los mismos en la asignatura. Esta plataforma no solo permitirá a los docentes reflejar los contenidos de la materia en preguntas concretas sobre cierto nivel de detalle de la asignatura u organizar las distintas competencias asociadas a la misma, sino que también facilitará la creación de exámenes personalizados y análisis de resultados obtenidos, buscando proporcionar versatilidad y eficiencia a la hora de evaluar una asignatura.

En una primera aproximación, una posible estructura para organizar los ítems de una asignatura sería de manera jerárquica, permitiendo que los ítems estuvieran englobados en ítems más generales y, en consecuencia, que los ítems puedan desarrollarse en otros ítems más específicos.

De forma general, se pueden establecer una serie de objetivos para el desarrollo de la plataforma:

- Crear una plataforma que pueda comunicarse por algún medio con plataformas de aprendizaje extendidas como Moodle<sup>TM</sup>.
- Permitir la gestión de forma organizada de una asignatura según desee el usuario, sin restringir la estructura únicamente, por ejemplo, al contenido de la asignatura.
- Implementar medidas de seguridad para permitir acceso a recursos únicamente al propietario.
- Presentar la información de los distintos objetos de la herramienta de forma clara y concisa.
- Integrar tecnologías de uso sencillo, para permitir un entendimiento rápido del código.
- Modularizar correctamente los distintos componentes de la plataforma, para encapsular las funciones y datos de cada componente.
- Disponer de una visualización sencilla, que permita al usuario realizar todas las funcionalidades disponibles explicando concisamente las consecuencias de las distintas acciones.
- Contar con una plataforma cohesionada, tanto estructural como visualmente, con el objetivo de dar una experiencia de usuario mejor.

Además de los objetivos intrínsecos de la plataforma, es importante establecer unos objetivos de lo que conlleva el desarrollo de este proyecto. Estos añadidos no sólo enriquecen el proyecto en sí, sino también el proceso de creación del mismo:

- Realizar una solución siguiendo una metodología Ágil, para llevar un ritmo de trabajo constante.
- Documentar exhaustivamente todo el trabajo realizado a través de una memoria.
- Estudiar múltiples alternativas para la herramienta, para seleccionar de entre una variedad de posibilidades.
- A pesar de no ser un caso real, realizar un estudio estimado de los recursos necesarios para realizar el proyecto.
- Estudiar los casos límite de las distintas funcionalidades de la plataforma.
- Establecer la herramienta de manera que esté abierta a posibles ampliaciones en un futuro.

El proyecto se podría desglosar en tres grandes partes que se desarrollarán a continuación:

- **Creación de una jerarquía de aprendizaje con preguntas.** Utilizando el material de la asignatura (guías docentes, guión de la asignatura...), el docente transformará los ítems de aprendizaje de la asignatura en una jerarquía de aprendizaje. En base a la jerarquía creada, se creará un banco de preguntas las cuales serán asociadas a los distintos nodos de la jerarquía, además de almacenar otros datos propios de cada pregunta, (complejidad, duración estimada, respuesta correcta...).
- **Generación de exámenes basados en ítems.** Una vez creado el banco de preguntas, el docente podrá generar un examen especificando los ítems a evaluar en la prueba en lugar de las preguntas concretas. Asimismo, se podrán especificar otros parámetros, como la duración, la dificultad o el tipo de pregunta. La plataforma generará automáticamente un examen según los parámetros proporcionados, permitiendo la configuración de exámenes más ajustados a los criterios de los docentes.
- **Análisis de resultados.** una vez realizados los exámenes, el docente podrá ver un análisis de los resultados de los mismos.

### 1.2.1. Creación de una jerarquía de aprendizaje con preguntas.

Para permitir generar preguntas de aprendizaje, el contenido de la asignatura deberá estar organizado en una jerarquía de ítems basados en las competencias y resultados de aprendizaje de la asignatura, en la que el nodo raíz de la jerarquía es el nivel de menor

detalle (es decir, para preguntas sobre la asignatura en general), y los nodos hoja de la jerarquía son el nivel de mayor detalle. Este nivel ha de establecerlo el propio profesor, ya que el nivel puede corresponder a temas completos, secciones del mismo, epígrafes, o incluso líneas de epígrafe. En el caso de organizar el contenido por competencias, el nivel de mayor detalle podría ser, por ejemplo, competencias o subcompetencias entre otros.

Se supone el siguiente caso; para organizar la asignatura de Periféricos y Dispositivos de Interfaz Humana, el docente podría establecer la siguiente jerarquía de aprendizaje en base al contenido de la asignatura:

- Periféricos y Dispositivos de Interfaz Humana
  - Tema 1: Gestión a Bajo Nivel
    - BIOS
    - API del Sistema Operativo
    - Acceso desde ensamblador
    - Acceso desde C
  - Tema 2: Interfaz Hardware
    - Ranuras de expansión
    - Dispositivos RAID
    - Conectores
  - Tema 3: Dispositivos de Entrada
    - Teclado
    - Ratón
    - Digitalizador
    - Lectores de códigos de barras
    - Escáner
  - Tema 4: Dispositivos de Salida
    - Impresora
    - Sistemas de visualización
  - Tema 5: Dispositivos de Ocio
    - Tarjetas de sonido
    - Altavoces
    - Dispositivos de videojuegos
    - Formatos gráficos
  - Tema 6: Sensores
    - Transductores
    - Sensores
  - Tema 7: Dispositivos de Ayuda
    - Dispositivos para discapacidad visual

- Dispositivos para discapacidad cognitiva
- Dispositivos para discapacidad auditiva
- Prótesis
- Tema 8: Dispositivos de Almacenamiento
  - Almacenamiento magnético
  - Almacenamiento óptico
  - Almacenamiento eléctrico

En este caso, el docente ha considerado oportuno dejar el nivel de mayor detalle en secciones de temas. Cabe mencionar que el docente podría haber organizado la asignatura basándose en las competencias de la asignatura (apreciar las ventajas de distintas configuraciones RAID, conocer las características de los dispositivos fundamentales de entrada y salida, identificar las ventajas de dispositivos multimedia...).

Definida ya la jerarquía de aprendizaje, es posible crear preguntas para añadir al banco de preguntas. Estas han de estar asociadas, al menos, a uno de los nodos de la jerarquía de aprendizaje.

A continuación, siguiendo con el ejemplo, el profesor crea la siguiente pregunta:

¿Cuál de las siguientes no es una tarjeta de expansión?

- A. EISA.
- B. ISA.
- C. VESA.
- D. DESA.

Observando la jerarquía de aprendizaje, esta pregunta estaría asignada al nodo 'Ranuras de expansión'. A continuación, el profesor crea la siguiente pregunta:

¿Cuál es una desventaja del almacenamiento eléctrico respecto al óptico?

- A. El almacenamiento eléctrico es superior al óptico en todos los aspectos.
- B. El almacenamiento eléctrico es volátil, aunque tome mucho tiempo que los datos guardados se pierdan.
- C. El almacenamiento eléctrico solo permite una escritura, mientras que el almacenamiento óptico permite varias escrituras.

Observando la jerarquía de aprendizaje establecida, esta pregunta podría asignarse a los nodos 'Almacenamiento eléctrico' y 'Almacenamiento óptico'.

Cabe destacar que, además de las preguntas más tradicionales (preguntas de tipo test, preguntas de desarrollo...), se permitirá crear preguntas parametrizadas, en las que se permitirá la inclusión de parámetros en el enunciado y/o las respuestas de la pregunta. El objeto de este tipo de preguntas sería contar con una representación más condensada de un conjunto de preguntas con una estructura similar pero con versiones distintas entre sí, permitiendo una gestión de este conjunto de preguntas más rauda y eficiente. Estos parámetros vendrán especificados como "##param1##", "##param2##",

”##param3##”... Por otra parte, para poder dar distintos valores a los parámetros, se han de especificar también las distintas configuraciones de parámetros que se quieren establecer, para saber qué valores de cada parámetro se relacionan con otros valores de otros parámetros. Por ejemplo:

¿Cuál es la interrupción de acceso a kernel en ##param1##?

A. La interrupción ##param2##.

B. La interrupción ##param3##.

C. La interrupción ##param4##.

D. No existe ninguna interrupción para acceder al kernel.

##param1##	##param2##	##param3##	##param4##
Linux	INT 80h	INT 21h	syscall
MSDOS	INT 21h	INT 80h	syscall
MacOS	syscall	INT 80h	INT 21h

Tabla 1: Tabla de configuraciones de valores de parámetros de pregunta parametrizada.

### 1.2.2. Generación de exámenes basados en competencias

Una vez creadas las preguntas, el docente podrá crear exámenes; para ello, se deberán especificar el número de secciones en las que dividir el examen, y para cada sección, los nodos relacionados con las preguntas y el número de preguntas. Opcionalmente, se podrán especificar otros parámetros para cada sección, entre otros:

- Dificultad aproximada de cada pregunta.
- Tiempo aproximado de cada pregunta.
- Tipo de las preguntas.
- Bloqueo/Fijación de preguntas concretas.

Al generar los exámenes, para elegir las preguntas, se tendrían en cuenta ciertas preguntas antes que otras (preguntas que hayan tenido menos usos, preguntas más cercanas a la dificultad pedida...). Asimismo, el docente ha de ser consciente de que, en algunos casos, no será posible cumplir todos los parámetros indicados a la hora de generar un examen (por ejemplo, no se podrá generar un examen de diez preguntas de un ítem si ese ítem tiene asociadas menos de diez preguntas).

### 1.2.3. Visualización de resultados

Realizados los exámenes y obtenidos los resultados, el último bloque de este proyecto visualizaría los resultados, relacionándolos con la jerarquía de aprendizaje: de esta manera, el docente podría observar qué contenidos de aprendizaje han tenido mejores



resultados de los alumnos y qué contenidos necesitan ser reforzados. Cabe mencionar que la idea de este apartado sería únicamente la visualización de los resultados, y no sería en ningún momento un apartado para llevar a cabo exámenes ni para evaluar la corrección de los mismos. Esta herramienta contaría con la funcionalidad de producir los exámenes, que estos fueran realizados por los examinados a través de otros medios (entre otros, impresos o escritos en papel, o a través de plataformas externas como Moodle<sup>TM</sup>) y, tras recoger los resultados de los mismos, que los resultados se importaran y visualizaran en la plataforma.

### 1.3. Protección de datos personales

Dado que se almacenan los datos de resultados de exámenes, cabe preguntarse por los aspectos legales relativos a la protección de datos. Es importante tener en cuenta que el proyecto a desarrollar tiene como objetivo apoyar la evaluación del contenido de una asignatura, y no realizar la gestión de la evaluación de la misma. Dicho de otra manera, dentro de los objetivos de ese proyecto no se espera bajo ningún concepto conducir la realización de exámenes o la evaluación de examinados a través de exámenes generados por la plataforma, sino que se espera que dichos exámenes sean realizados por medios ajenos a la plataforma, y que los resultados se recojan en la plataforma únicamente para visualizar los resultados de exámenes ya realizados y para complementar la configuración de nuevos exámenes, la edición de exámenes previos y la creación de nuevas preguntas. Para evitar problemas generados por un uso incorrecto de la plataforma, es importante recalcar que, debido a la Ley de Protección de Datos Vigente, no podrán almacenarse datos identificativos (DNI, nombre, apellidos, entre otros), o deberán ser anonimizados (por ejemplo, para identificar a un examinado se podrá usar un ID generado automáticamente que no tenga que ver con los datos previamente mencionados) [12] .

### 1.4. Estudio de soluciones existentes

En internet existen varios servicios con similitudes a este proyecto; por tanto, se comentarán a continuación los más destacables. Cabe resaltar que la gran mayoría de las herramientas que se van a comentar cuentan con funcionalidades para llevar a cabo el examen en la misma plataforma, pero esta funcionalidad no es esperada de la herramienta a desarrollar.

#### 1.4.1. Moodle<sup>TM</sup>

Moodle<sup>TM</sup> [15] es un servicio muy cercano a esta aplicación, además de uno de los servicios de educación más usados alrededor del mundo, por lo que merece una especial mención en este trabajo.

Moodle<sup>TM</sup> es una plataforma de gestión de aprendizaje que permite la creación de cursos, exámenes y otro tipo de ejercicios. Pese a que cuenta con bastantes funcionalidades en su forma gratuita, la gran mayoría de utilidades que tiene requieren de un

conocimiento algo avanzado en informática, puesto que la plataforma delega las responsabilidades de instalación del sistema, gestión de la base de datos y conectividad, gestión de usuarios, gestión de cursos, y otras funcionalidades al administrador. Esto hace que un usuario que no tiene este tipo de habilidades tenga bastantes dificultades para utilizar de forma gratuita todos los servicios que ofrece esta plataforma. No obstante, esta plataforma ofrece ciertos servicios de pago; en concreto, ofrece la posibilidad de contar con la asistencia de un experto en la plataforma que aconseja el tipo de instalación que se debería crear dependiendo de las necesidades del administrador y su instancia, actuando este como asistencia del administrador.

Moodle™ cuenta con una gran cantidad de módulos (gestión de cursos y usuarios, comunicación, evaluación y calificación, seguridad, mantenimiento...) que se complementan entre sí para dotar a la plataforma de una experiencia mucho más cohesionada y diversa. Los módulos básicos de las instancias de Moodle™ cuentan con pocas funcionalidades de las que buscamos en este proyecto (contando simplemente con la puntuación efectiva de preguntas y la visualización de resultados), aunque la inclusión de módulos adicionales en la instancia hace que las funcionalidades aumenten en cantidad y calidad, como exportar exámenes y añadir atributos varios a las preguntas. Sin embargo, la inclusión de más módulos en la instancia puede llevar a complicar el mantenimiento de la instancia.

Por otra parte, Moodle™ es una plataforma de código libre, por lo que se permite ampliar el código de la instancia inicial siempre que sea necesario. La documentación de Moodle™ es muy extensa, dando mucha información de los diferentes módulos extra que ofrece la plataforma, e incluso cuenta con un apartado de preguntas de la comunidad, en la que se pueden hacer preguntas que otros administradores pueden responder. Esta extensa documentación permite que los administradores que no conocen la plataforma puedan familiarizarse fácilmente con ella.

Moodle™, como muchos otros programas, tiene la arquitectura de una pila LAMP: como sistema operativo se vale de Linux (aunque, lógicamente, muchos de los programas que surgen de Moodle™ se pueden ejecutar sin mucho problema en otros sistemas operativos), utiliza un servidor Apache, tiene como base de datos MariaDB y se vale de PHP para crear la interfaz web con la que el usuario final interactúa. Estos componentes se comunican entre sí para crear la instancia de Moodle™.

Se ha querido destacar Moodle™ por separado ya que, debido a su extenso uso en más de cien mil páginas repartidas por más de 200 países del mundo, es innegable que muchas organizaciones de formación (entre las que se encuentran muchas universidades españolas) lo conocen y usan a diario. La plataforma que se va a desarrollar en este Trabajo de Fin de Grado busca, además de servir como plataforma independiente, ser una ayuda para la evaluación de asignaturas en Moodle™, para que dichas organizaciones (como, por ejemplo, la Universidad de Granada) tengan una herramienta más para ayudar al desarrollo académico de sus alumnos. Sería de gran utilidad que la herramienta a desarrollar cuente con alguna forma de comunicación con Moodle™, para permitir el apoyo a una plataforma con tanto uso.

### 1.4.2. Otras soluciones

- En primer lugar, existe un **proyecto de GitHub** (<https://javierfpanadero.github.io/generador-examenes/>) de Javier F. Panadero que crea exámenes de tipo test, exportables a XML para subirlos a plataformas como Moodle™ o un formato de texto para impresión. El programa es muy sencillo: a partir de un examen ejemplo, se generan varias preguntas del mismo tipo. Este servicio tiene varias diferencias con el proyecto: por una parte, el servicio no está orientado a personas sin ningún conocimiento de informática. Por otra parte, el proyecto no da ningún tipo de análisis sobre los resultados de los exámenes, por lo que se desconoce si las preguntas seleccionadas son efectivas o no, teniendo en cuenta que no se guarda ninguna información sobre la pregunta (tipo, dificultad...) [8].
- En segundo lugar, **Kahoot** [10] presenta un servicio de creación de pruebas tipo test, así como estadísticas de las preguntas de cada Kahoot. Un Kahoot es una prueba formada por preguntas tipo test, cuya puntuación varía según la velocidad de respuesta, el número de respuestas seguidas y, por supuesto, el acierto o fallo. La principal diferencia que supone Kahoot frente al proyecto es la variabilidad de las pruebas: un Kahoot mantiene siempre las mismas preguntas, variando únicamente el orden de preguntas y respuestas, por lo que no podría cambiar cantidad, dificultad o categoría de las preguntas, entre otros. Además, el sistema de puntuación de Kahoot es muy laborioso, ya que premia la velocidad al mismo nivel que la exactitud de la respuesta, a diferencia de los exámenes de tipo test tradicionales. Asimismo, Kahoot tiene varios servicios de suscripciones que permiten nuevas versiones de test.
- En tercer lugar, FlexiQuiz [3] presenta un servicio muy cercano a este proyecto: FlexiQuiz recibe preguntas, crea tests de múltiple opción, y crea automáticamente informes de los resultados de los mismos. Asimismo, está respaldado por una transmisión segura con encriptación SSL y una fácil accesibilidad tanto para creadores de test como para personas que respondan. La principal diferencia que tiene el proyecto con FlexiQuiz es la posibilidad de clasificar las preguntas según su dificultad y la categoría; la única posibilidad que nos ofrece FlexiQuiz es la selección manual de ciertas preguntas. Asimismo, FlexiQuiz ofrece varios tipos distintos de preguntas, desde las de tipo test de elección múltiple, pasando por las de coincidencia de texto hasta incluso subida de archivos. FlexiQuiz también tiene un servicio de suscripción de pago con mejoras para los tests, así como la posibilidad de conducir la realización de tests en la propia plataforma.
- En cuarto lugar, la aplicación Fyrexbox [7] tiene un servicio muy similar al anterior: el servicio recibe preguntas tipo test que conforman un examen. La particularidad de esta aplicación web es la generación automática de preguntas: dado un tema, crea algunas preguntas tipo test; a esto se le añade la posibilidad de utilizar un archivo CSV para insertar las preguntas. El problema de Fyrexbox, como el de otros servicios ya comentados, es la poca variabilidad de las preguntas: las preguntas

son siempre las mismas, sin dar margen a que las preguntas sean clasificadas por su jerarquía o su dificultad. Además, Fyrexbox no genera ningún tipo de análisis estadístico de las preguntas más allá del porcentaje de acierto.

- Por último, la aplicación web TestWizard [19] permite crear exámenes y otro tipo de ejercicios de repaso de preguntas de Educación Primaria y Secundaria. El sistema tiene acceso a una base de datos de muchas asignaturas (matemáticas, biología, inglés, español, historia...) para crear estas preguntas. El problema de TestWizard es que se remite a las categorías ya mencionadas, y resulta bastante laborioso añadir preguntas, sin tener posibilidad de añadir preguntas de categorías nuevas (por ejemplo, materias a nivel universitario). Además, como otras plataformas, no provee ningún tipo de análisis estadístico de los resultados, ya que se remite a la creación de exámenes y ejercicios.

En la tabla 2 se puede observar una comparativa de todas las plataformas mencionadas:

	Proyecto Github	Kahoot	Flexiquiz	Fyrexbox	TestWizard	Moodle™
Facilidad para usuarios no expertos	No	Sí	Sí	Sí	Sí	Sí (tareas generales)
Preguntas con distintos atributos (dificultad, tipo...)	No	No	Sí	No	Sí	Sí
Parametrización de preguntas	Sí	No	No	Sí	No	No
Puntuación efectiva de las preguntas	No	No	Sí	Sí	Sí	Sí
Exámenes exportables o importables	Sí	No	No	Sí	Sí	Sí
Generación o sugerencia de preguntas del examen	No	No	Sí	Sí	Sí	No
Análisis de resultados	No	Sí	Sí	No	No	No
Preguntas relacionadas a ítems de la asignatura	No	No	No	No	No	No

Tabla 2: Tabla de comparación de soluciones existentes.

Al observar las características de las plataformas disponibles, la plataforma a desarrollar resulta novedosa en ciertos aspectos, como al dar soporte a un generador de exámenes que se base en los ítems de una asignatura, o la relación de las preguntas de un banco con los nodos que representan dichos ítems. Como se ha mencionado, la organización en base a ítems que representan el contenido o las competencias de una asignatura y la generación de exámenes en base a esta organización será el núcleo del proyecto a de-

---

sarrollar, con el objeto de presentar una plataforma con un enfoque distinto e innovador en la configuración de exámenes en comparación a la oferta actual.

## 2. Solución propuesta

Teniendo en cuenta las necesidades de la aplicación, los objetivos y las funcionalidades que se piden, se presenta una propuesta de solución que cumple lo que se busca. No obstante, cabe destacar que la solución propuesta es una posibilidad de múltiples existentes, como se demostrará más adelante, aunque se ha seleccionado la solución que, a nuestro parecer, se adapta mejor a lo que se pide.

### 2.1. Metadatos asociados a cada entidad

En este apartado se expondrán los metadatos asociados a cada una de las entidades de la plataforma. Estos datos serán, entre otros, los que se pidan al ser añadidos o modificados, o los que se muestren al pedir ver un objeto.

- **Metadatos de un usuario:**

- Nombre.
- Correo electrónico.
- Contraseña.

- **Metadatos de una asignatura:**

- Nombre.

- **Metadatos de un nodo de la jerarquía de una asignatura:**

- Nombre.
- Padre. Nodo de la misma jerarquía del que es hijo.

- **Metadatos de una pregunta:**

- Enunciado de la pregunta.
- Tipo de la pregunta, uno de los siguientes:
  - Test. Se deberán especificar posibles respuestas y su puntuación en  $[-100,100]$ .
  - Desarrollo. Se podrá especificar tanto posibles respuestas como indicaciones de la respuesta a dar.
- Nodo o nodos de la jerarquía a los que pertenece.
- Dificultad estimada, en  $[1, 10]$ .
- Tiempo estimado de realización, en minutos.

Opcionalmente:

- Respuestas adicionales (en el caso de las preguntas de tipo test, se deben especificar al menos dos).

- Puntuación de cada respuesta (en el intervalo  $[-100, 100]$ ).
- Parámetros de pregunta. Se especificarán el número de configuraciones de parámetros y el número de parámetros en cada una.

■ **Metadatos de un examen:**

- Título del examen.
- Número de secciones, y por cada sección:
  - Nodo o nodos de la jerarquía a los que se asocian las preguntas de la sección.
  - Número de preguntas de la sección.

Opcionalmente, para cada sección:

- Dificultad aproximada de cada pregunta.
- Tiempo aproximado de cada pregunta.
- Tipo prioritario de las preguntas.
- La prioridad de preguntas parametrizadas sobre no parametrizadas.
- Bloqueo/Fijación de preguntas concretas.

■ **Metadatos de un resultado de un examen:**

- ID del examinado.
- Tiempo de realización.
- Puntuación obtenida.

## 2.2. Formatos de exportación e importación de archivos

Como se explicará más adelante, algunas de las funcionalidades de la solución que se propone incluyen la exportación o importación de archivos. En este apartado se describirán las características de los distintos archivos que se pueden importar y exportar en la aplicación. Previo a esto, es importante mencionar los formatos de algunos archivos menos convencionales:

- **Aiken.** Aiken [4] es un formato nativo de Moodle<sup>™</sup>, en el que se dispone un listado en texto plano de preguntas de tipo test con una estructura concreta que se puede comprender fácilmente. Estos formatos serían empleados para importar preguntas y exportar exámenes. En el caso de la exportación de exámenes, realmente se exporta un conjunto de preguntas, que funcionan como un examen. Es importante que el archivo esté guardado como un archivo de texto plano, y usando caracteres ASCII. La estructura de una pregunta en un archivo Aiken se vería así:

*¿Qué es AGP?*  
*A. Las siglas del Aeropuerto de Madrid.*  
*B. Una ranura de una tarjeta de expansión.*  
*C. Un tipo de conector externo.*  
*D. Ninguna de los anteriores es correcta.*  
**ANSWER: C**

- **GIFT.** GIFT [5] es también un formato de Moodle™, en el que se dispone de un listado en texto plano de preguntas con menos restricciones, pudiendo incluir más información de la pregunta, como diferentes tipos de pregunta, diferentes puntuaciones para una respuesta o un título para la pregunta. Al igual que con los archivos Aiken, es importante que el archivo esté guardado como un archivo de texto plano, y usando caracteres ASCII. De la misma manera, este archivo serviría también para exportar exámenes siguiendo la misma lógica que los archivos Aiken. Un ejemplo del contenido de un archivo en GIFT es el siguiente:

```

::Pregunta 90::¿Qué es una impresora 3D? {
~%0%Una impresora que crea documentos.
~%100%Una impresora que crea objetos tridimensionales.
~%0%Una impresora que envía faxes.
~%-100%Un dispositivo para escuchar la radio.
}

::Pregunta 91::Explica cómo se ajusta una imagen en una pantalla {
}

```

La pregunta 90 sería una pregunta de tipo test, en la que la segunda respuesta tendría una puntuación de -100%, la cuarta con una puntuación de 100%, y la primera y tercera con una puntuación de 0%. La pregunta 91 correspondería a una pregunta de desarrollo sin respuestas.

- **MoodleXML.** MoodleXML [6] es un archivo también de Moodle™, que utiliza la estructura de XML. Este formato es algo más difícil de interpretar por humanos, pero permite también mucha variedad en las preguntas. Se recurriría a estos archivos para la exportación de exámenes, aunque con estos archivos sí se determina que el archivo contiene un examen, no un conjunto de exámenes. Este es un ejemplo del contenido de un archivo en MoodleXML:

```

1 <quiz>
2   <question type="multichoice">
3     <name>
4       <text>Pregunta 92.</text>
5     </name>
6     <questiontext format="html">
7       <text>Indique el dispositivo para almacenar datos de forma permanente.
          </text>

```



```

8      </questiontext>
9      <answer fraction="0">
10         <text>Memoria RAM</text>
11      </answer>
12      <answer fraction="0">
13         <text>Procesador</text>
14      </answer>
15      <answer fraction="0">
16         <text>Pantalla</text>
17      </answer>
18      <answer fraction="100">
19         <text>Disco Duro</text>
20      </answer>
21 </question>
22 <question type="essay">
23     <name>
24         <text>Pregunta 93</text>
25     </name>
26     <questiontext format="html">
27         <text>Describe el funcionamiento de las pantallas LCD.</text>
28     </questiontext>
29 </question>
30 </quiz>

```

Además de los formatos mencionados, se utilizan otros formatos más convencionales:

- **PDF.** Se dispone de estos archivos para exportar exámenes en un formato más orientado a la realización presencial, con el objetivo de usar el propio archivo generado para la realización del examen.
- **ODT.** El propósito de este formato es el mismo que el de PDF, generar exámenes para su realización en papel, aunque permitiendo que el docente pueda editar el contenido del examen por diversos motivos (añadir una cabecera oficial, cambiar la distribución del examen...).
- **CSV.** El uso de CSV se da tanto al importar un banco de preguntas como para importar los resultados. Por una parte, se buscaba un formato más convencional para importar preguntas, además de con mayor flexibilidad, para permitir, por ejemplo, importar preguntas de desarrollo o preguntas de tipo test con respuestas con puntuaciones distintas a una única respuesta correcta y el resto sin ninguna puntuación. Por otra parte, por razones muy parecidas, para importar los resultados este formato encajaba con las necesidades de importar resultados, al permitir la inserción de todos los atributos necesarios para crear un resultado con un formato de archivo más convencional.

### 2.3. Requisitos Funcionales

Aunque la plataforma cuenta con algunas funcionalidades extra, el funcionamiento básico de la misma es el siguiente:

El usuario, después de haber registrado una cuenta en la plataforma, inicia sesión con su cuenta. En la pantalla de inicio, puede observar sus asignaturas, las cuales puede

crear, borrar o seleccionar para visualizar o editar. Al seleccionar una, podrá ver varios elementos de la asignatura: en primer lugar, la jerarquía de aprendizaje, que podrá modificar añadiendo, editando y borrando nuevos nodos de la misma. Por otra parte, el banco de preguntas, en el que podrá crear preguntas nuevas, y ver las preguntas (tanto el listado de todas las preguntas como cada pregunta por separado si se selecciona) y podrá borrar y modificar preguntas existentes. Por último, la sección de exámenes, en la que podrá configurar nuevos exámenes, ver los exámenes (tanto el listado de exámenes completo como cada examen por separado si uno concreto es seleccionado), que podrá modificar, borrar y exportar, además de poder importar resultados del examen en cuestión.

En consecuencia de lo anterior, los requisitos funcionales de la aplicación son los siguientes:

Tabla 3: Tabla de Requisitos Funcionales

RF	Nombre	Descripción
<b>Autenticación</b>		
RF1	Registro de Usuario	El usuario debe poder registrarse en la plataforma a través de un formulario de registro. Será necesario especificar los metadatos de un usuario.
RF2	Inicio de Sesión de usuario	El usuario debe poder iniciar sesión a través de su correo y su contraseña.
RF3	Cierre de sesión	El usuario debe poder cerrar sesión en la plataforma.
<b>Gestión de asignaturas</b>		
RF4	Crear asignatura	El usuario debe poder crear una asignatura, especificando los metadatos de la asignatura. Al crear una asignatura, se crea la jerarquía asociada, formada sólo por el nodo raíz, con el nivel de menor detalle (la asignatura).
RF5	Editar asignatura	El usuario debe poder cambiar el título de la asignatura.
RF6	Borrar asignatura	El usuario debe poder borrar una asignatura que ha creado, teniendo en cuenta que se borrarán todos los objetos asociados a la misma (jerarquía de aprendizaje, banco de preguntas, exámenes, resultados e informes).

RF	Nombre	Descripción
RF7	Listar asignaturas	El usuario debe poder ver una lista de sus asignaturas creadas.
RF8	Ver asignatura	El usuario debe poder ver los metadatos de una asignatura concreta de sus asignaturas creadas, para ver detalles de la misma.
<b>Gestión de jerarquía de aprendizaje</b>		
RF9	Añadir nodo de la jerarquía	El usuario debe poder añadir un nodo a una jerarquía de aprendizaje. Este sólo podrá ser un nodo hijo del nodo al que se relaciona, por lo que no se podrá crear un nuevo nodo relacionado directamente con un nodo de mayor detalle. El usuario deberá especificar los metadatos del nodo.
RF10	Editar nodo de la jerarquía	El usuario debe poder editar el nombre de un nodo de una jerarquía de aprendizaje.
RF11	Eliminar nodo de la jerarquía	El usuario debe poder eliminar un nodo de una jerarquía de aprendizaje. El usuario sólo podrá eliminar nodos hoja, es decir, nodos que no tengan nodos de mayor detalle directamente relacionados.
RF12	Ver jerarquía de aprendizaje	El usuario debe poder ver la jerarquía de aprendizaje de una asignatura al completo, visualizando las distintas relaciones entre los nodos.
RF13	Ver nodo de la jerarquía	El usuario debe poder ver un nodo concreto de la jerarquía de aprendizaje de una asignatura, y poder acceder a los detalles del mismo.
<b>Gestión de preguntas</b>		
RF14	Crear pregunta del banco	El usuario debe poder crear una pregunta. Se deberán especificar los metadatos de la pregunta, además de los parámetros opcionales que se consideren oportunos.

RF	Nombre	Descripción
RF15	Editar pregunta	El usuario debe poder editar una pregunta. Sólo se permitirá editar preguntas sin recursos asociados. Se deberán especificar los nuevos metadatos de la pregunta, además de los nuevos parámetros opcionales que se consideren oportunos.
RF16	Borrar pregunta del banco	El usuario debe poder borrar una pregunta. Una pregunta sólo podrá ser borrada en caso de no estar asociada a ningún examen.
RF17	Ver banco de preguntas	El usuario debe poder ver el banco de preguntas correspondiente a una de sus asignaturas.
RF18	Ver pregunta	El usuario debe poder visualizar los detalles de una pregunta. Se deberán mostrar todos los metadatos de la pregunta.
RF19	Importar banco de preguntas	El usuario debe poder importar un banco de preguntas. Deberá importarse en un formato compatible (Aiken o CSV).
RF20	Desactivar pregunta	El usuario debe poder desactivar una pregunta. La pregunta, una vez desactivada, no podrá volver a activarse, y, aunque podrá salir en resultados, no será elegible al generar nuevos exámenes.
<b>Gestión de exámenes</b>		
RF21	Crear examen	El usuario debe poder crear un examen. El usuario ha de especificar el conjunto de metadatos del examen a crear, así como los metadatos opcionales que considere relevantes.
RF22	Editar examen	El usuario debe poder editar un examen sin resultados asociados, pudiendo cambiar todos los metadatos del examen.
RF23	Borrar examen	El usuario debe poder borrar un examen. Un examen sólo podrá borrarse si no tiene resultados asociados.
RF24	Listar exámenes	El usuario debe poder ver una lista de los exámenes creados para una asignatura, así como un resumen de cada una.

RF	Nombre	Descripción
RF25	Ver examen	El usuario debe poder ver los detalles de un examen. El usuario podrá ver todos los metadatos del examen.
RF26	Exportar examen	El usuario debe poder exportar un examen. El usuario recibirá el examen en un formato que facilitará su uso o edición en otras aplicaciones (Aiken, GIFT, MoodleXML, ODT o PDF).
RF27	Comprobar preguntas	El usuario debe poder comprobar la repetición de las preguntas de un examen en exámenes anteriores. El usuario deberá especificar los exámenes en los que se quiere comprobar la repetición.
<b>Gestión de resultados</b>		
RF28	Importar resultados de un examen	El usuario debe poder importar resultados. Los resultados se reciben en un formato compatible (CSV) y se añaden a la plataforma, siendo asociados a un examen.
RF29	Borrar resultados de un examen	El usuario debe poder borrar el archivo de resultados de un examen. Al borrar los resultados, al examen se le podrá asociar un nuevo archivo de resultados.
RF30	Ver resultados de un examen	El usuario debe poder ver los resultados de un examen. En ellos podrá ver estadísticas de distintos datos asociados al examen.

## 2.4. Modelo de Datos

Para la base de datos se propone el uso de una base de datos relacional. Esto se debe a que el proyecto no cuenta con unas necesidades muy específicas, ni las entidades o las relaciones entre ellas son muy complejas. No obstante, es importante mencionar que una alternativa habría sido una base de datos basada en grafos, especialmente para facilitar las operaciones con los distintos ítems de la jerarquía de aprendizaje. A pesar de esto, las jerarquías asociadas a las asignaturas no son de gran tamaño y sus operaciones se pueden resolver fácilmente en una base de datos relacional, por lo que no se consideró muy relevante y se descartó la idea de esgrimir una base de datos basada en grafos. Otra razón de peso para decantarse por una base de datos relacional es que son las más comunes, tanto en general como en las plataformas mencionadas anteriormente que se ofrecen ahora mismo. La popularidad de este tipo de bases de datos puede ser beneficioso en el caso de necesitar algún tipo de ayuda en caso de un problema. Para representar el diseño de la base de datos, el diagrama Entidad/Relación de la aplicación es el siguiente:

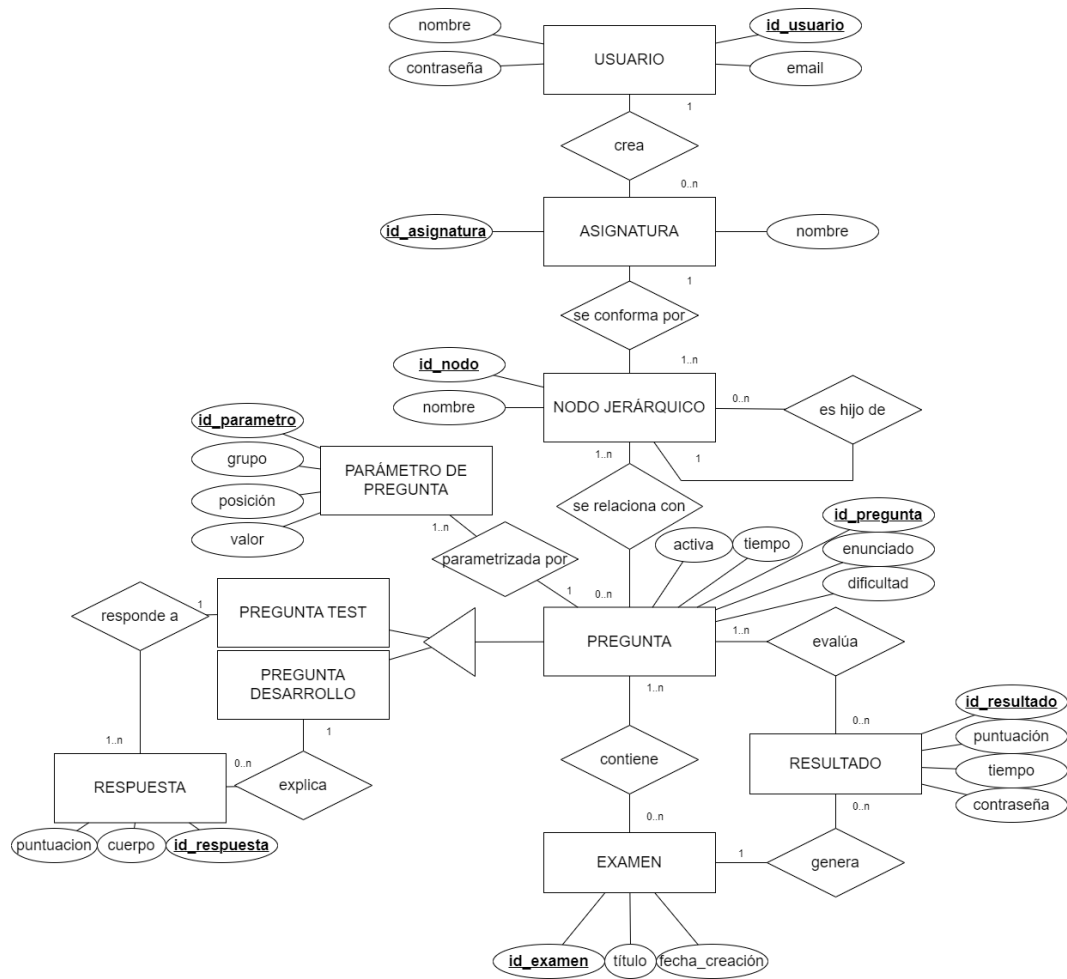


Figura 1: Diagrama E/R de la Base de datos de la aplicación.

#### 2.4.1. Diccionario de Datos

En este apartado se repasarán todas las entidades que se han presentado, para explicar sus atributos, así como las relaciones entre entidades:

- **Usuario:** Usuario de la plataforma.

- *ID de usuario.* Entero que identifica unívocamente a un usuario. Es la clave primaria de la entidad.
- *Email.* Correo electrónico único para cada usuario.
- *Nombre.* Cadena alfanumérica que representa el nombre del usuario.
- *Contraseña.* Cadena alfanumérica que guarda la contraseña del usuario. Por motivos de seguridad, el valor que se guarda está hashado para evitar tener el valor real de la contraseña.

- **Asignatura:** Asignatura de un usuario.
  - *ID de asignatura.* Entero que identifica unívocamente a una asignatura. Es la clave primaria de la entidad.
  - *Nombre de la asignatura.* Cadena alfanumérica que representa el título de la asignatura.
- **Nodo jerárquico:** Nodo de la jerarquía de aprendizaje de una asignatura. Un nodo puede representar un ítem, una sección de la asignatura o la asignatura al completo.
  - *ID de nodo.* Entero que identifica unívocamente a un nodo. Es la clave primaria de la entidad.
  - *Nombre del nodo.* Cadena alfanumérica que representa el nombre del nodo.
- **Pregunta:** Pregunta del banco de preguntas de una asignatura.
  - *ID de pregunta.* Entero que identifica unívocamente a una pregunta. Es la clave primaria de la entidad.
  - *Enunciado de la pregunta.* Cadena alfanumérica que representa el enunciado de la pregunta.
  - *Dificultad estimada.* Entero en  $[1, 10]$  que representa la dificultad estimada.
  - *Tiempo estimado.* Entero que representa el tiempo aproximado en minutos que se tarda en responder a la pregunta.
  - *Activa.* Valor booleano que representa si una pregunta esta activada o desactivada.
- **Pregunta Test:** Pregunta del banco de preguntas de tipo test. La diferencia con las Preguntas de Desarrollo está en el tratamiento de las respuestas.
- **Pregunta Desarrollo:** Pregunta del banco de preguntas de tipo desarrollo.
- **Respuesta:** Respuesta a una pregunta. Una pregunta de tipo test tiene que tener al menos dos respuestas con puntuación. Una pregunta de tipo desarrollo puede no tener respuestas y, si las tiene, estas son más bien posibles respuestas o directrices de lo que se busca en la respuesta.
  - *ID de respuesta.* Entero que identifica unívocamente a una respuesta. Es la clave primaria de la entidad.
  - *Cuerpo.* Cadena alfanumérica que representa el contenido de la respuesta.
  - *Puntuación.* Entero en  $[-100, 100]$  que representa el porcentaje de la puntuación de la pregunta que se obtendría al seleccionar la respuesta.
- **Parámetros de pregunta:** Parámetros de una pregunta parametrizada. Una pregunta parametrizada es una pregunta con una serie de variables en el enunciado

y/o respuestas que tiene diferentes posibles combinaciones de valores. De esta manera, se permite que varias preguntas con una estructura muy similar puedan fusionarse en una sola. Una pregunta es parametrizada solo si tiene parámetros de pregunta asociados.

- *ID de parámetro*. Entero que identifica unívocamente a un parámetro. Es la clave primaria de la entidad.
- *Grupo*. Entero que representa la configuración de parámetros a la que pertenece el parámetro.
- *Posición*. Entero que representa la posición del parámetro al que se da valor.
- *Valor*. Cadena alfanumérica que representa el valor a sustituir.

- **Examen:** Examen de una asignatura.

- *ID de examen*. Entero que identifica unívocamente a un examen. Es la clave primaria de la entidad.
- *Título*. Cadena alfanumérica que representa el título del examen.
- *Fecha de creación*. Fecha que representa el día de creación del examen. Este valor se añade automáticamente al crear el examen.

- **Resultado:** Resultado de un examinado en una pregunta concreta de un examen de la asignatura.

- *ID de resultado*. Entero que identifica unívocamente a un resultado. Es la clave primaria de la entidad.
- *Puntuación*. Entero en  $[-100, 100]$  que representa el porcentaje de la puntuación de la pregunta obtenido.
- *Tiempo estimado*. Entero que representa el tiempo estimado en minutos que se ha tardado en responder.
- *ID de examinado*. Entero que representa a un usuario. Como se ha mencionado anteriormente, este valor no puede ser identificativo de un examinado, sino que sirve únicamente para agrupar distintos resultados que pertenecen a una misma persona anonimizada.

Teniendo en cuenta las entidades, se va a proceder a hablar de las relaciones entre las entidades:

- **Crea (Usuario-Asignatura):** Relación entre Usuario y Asignatura. Un usuario puede crear varias asignaturas o ninguna, y una asignatura solo puede ser creada por un usuario.

- **Se conforma por (Asignatura-Nodo Jerárquico):** Una asignatura puede estar conformada por uno o varios nodos, y un nodo solo puede pertenecer a una asignatura. Una asignatura sólo se relaciona con un nodo jerárquico raíz, y un nodo jerárquico raíz sólo puede pertenecer a una asignatura.



- **Es hijo de (Nodo Jerárquico-Nodo Jerárquico):** Un nodo puede ser padre de varios nodos o ninguno, y un nodo solo puede ser hijo de otro nodo (excepto en el caso de los nodos raíz, que no tienen padre).
- **Se relaciona con (Nodo Jerárquico-Pregunta):** Un nodo puede asociarse a varias preguntas, y una pregunta puede asociarse a varios nodos.
- **Explica (Pregunta Desarrollo-Respuesta):** Una pregunta puede tener varias respuestas, y una respuesta solo puede pertenecer a una pregunta.
- **Responde a (Pregunta Test-Respuesta):** Una pregunta puede tener varias respuestas, y una respuesta solo puede pertenecer a una pregunta.
- **Parametrizada por (Pregunta-Parámetros de pregunta):** Una pregunta puede tener varios o ninguno, y un parámetro solo puede pertenecer a una pregunta.
- **Contiene (Pregunta-Examen):** Una pregunta puede estar contenida en varios exámenes, y un examen puede contener varias preguntas.
- **Genera (Examen-Resultado):** Un examen puede generar varios resultados o ninguno, y un resultado solo puede ser generado por un examen.
- **Evalúa (Pregunta-Resultado):** Una pregunta puede ser evaluada por varios resultados o ninguno, y un resultado solo puede evaluar una pregunta.

## 2.5. Casos de Uso

En vista de los Requisitos Funcionales establecidos anteriormente, se procede a desglosar en detalle los Casos de Uso, que definirán las acciones a hacer en la plataforma, acompañados de un Diagrama de Interacción para comprender mejor su funcionamiento:

Caso de Uso	Registrarse			CU-1
Actor	Usuario			
Tipo	Primario Real			
Referencias	RF1			
Precondición				
Postcondición	El usuario queda registrado en el sistema a través de su email.			
Autor	Rafael Delgado García-Valdecasas	Versión	1.0	
Propósito				
Crear una cuenta en el sistema con las credenciales proporcionados.				
Resumen				
Pide credenciales al usuario y crea una cuenta con las mismas.				
Curso Normal (Básico)				
1	Sistema: Pide datos de registro.	2	Usuario: Proporciona credenciales.	
3	Sistema: Comprueba las credenciales en el sistema.	4	Sistema: Crea la cuenta y muestra un mensaje de éxito.	
Cursos Alternos				
2a	Usuario: Proporciona datos no válidos.			
2b	Sistema: Avisa al usuario de la invalidez de los datos y pide credenciales de nuevo.			
3a	Sistema: las credenciales no son válidos.			
3b	Sistema: Muestra un mensaje de error.			
3c	Fin del Caso de Uso.			
Comentarios				

Tabla 4: CU-1: 'Registrarse'

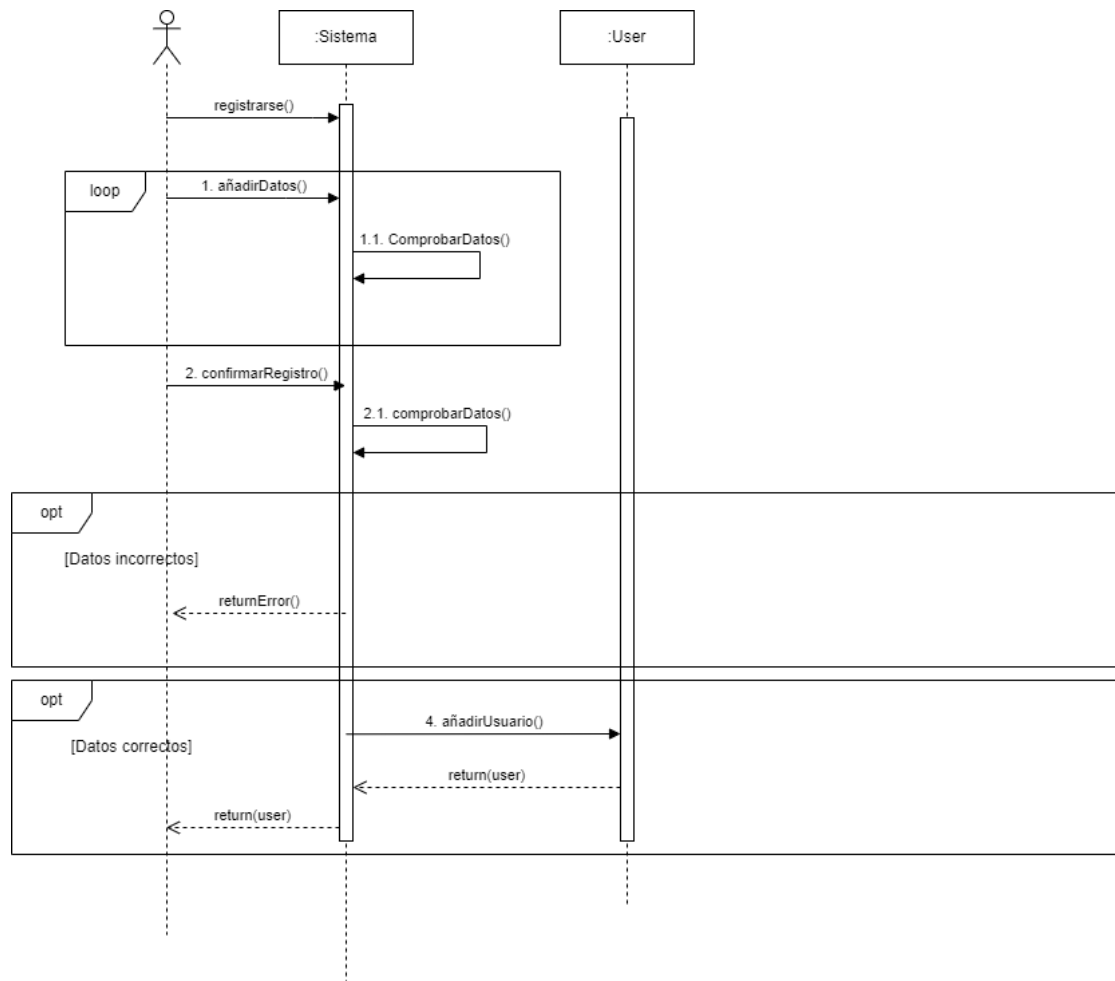


Figura 2: Diagrama de Interacción de CU-1: 'Registrarse'

Caso de Uso		Iniciar Sesión		CU-2
Actor		Usuario		
Tipo		Primario Real		
Referencias		RF2	CU-1	
Precondición		El usuario deber tener una cuenta en el sistema.		
Postcondición		El usuario inicia sesión en el sistema.		
Autor		Rafael Delgado García-Valdecasas	Versión	1.0
Propósito				
Inicia sesión para poder gestionar sus recursos asociados.				
Resumen				
Se piden credenciales de acceso al usuario y se inicia sesión en la plataforma.				
Curso Normal (Básico)				
1	Sistema: Pide credenciales de inicio de sesión.		2	Usuario: Proporciona credenciales de acceso.
3	Sistema: Comprueba las credenciales en el sistema.		4	Sistema: Crea un token de acceso, lo guarda y muestra un mensaje de éxito.
Cursos Alternos				
2a	Usuario: proporciona datos no válidos.			
2b	Sistema: avisa al usuario de la invalidez de los datos y pide credenciales de nuevo.			
3a	Sistema: no encuentra la cuenta o está desactivada.			
3b	Sistema: Muestra un mensaje de error.			
3c	Fin del Caso de Uso.			
Comentarios				

Tabla 5: CU-2: 'Iniciar Sesión'.

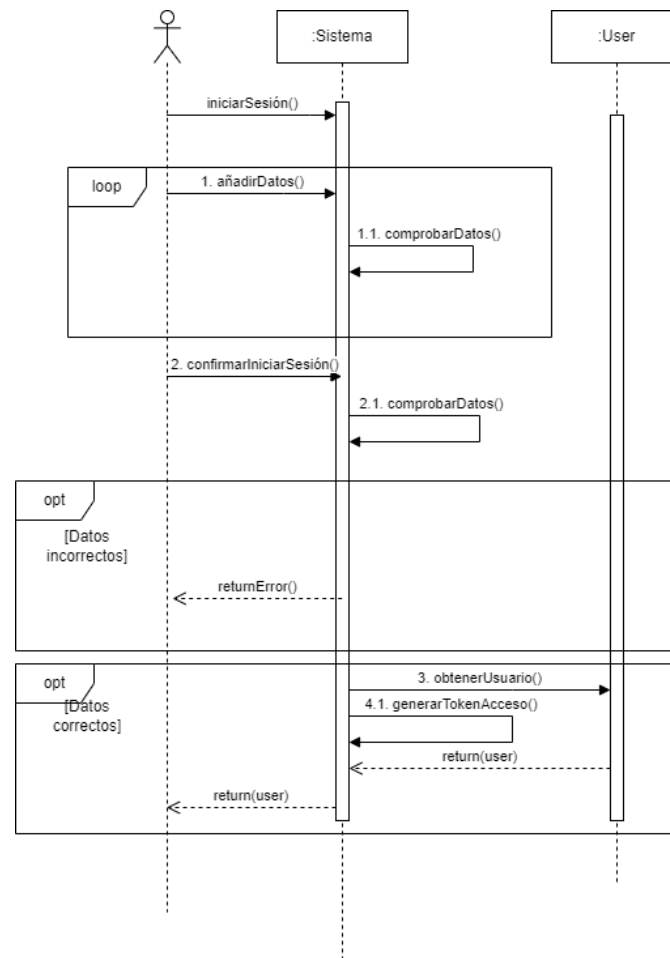


Figura 3: Diagrama de Interacción de CU-2: 'Inciar Sesión'.

<b>Caso de Uso</b>	<b>Cerrar sesión</b>	<b>CU-3</b>
<b>Actor</b>	<b>Usuario</b>	
<b>Tipo</b>	<i>Secundario Real</i>	
<b>Referencias</b>	<i>RF3</i>   CU-2	
<b>Precondición</b>	El usuario debe haber iniciado sesión.	
<b>Postcondición</b>	<i>El usuario cierra sesión.</i>	
<b>Autor</b>	<i>Rafael Delgado García-Valdecasas</i>	<b>Versión</b>   1.0
<b>Propósito</b>		
<i>Cerrar sesión.</i>		
<b>Resumen</b>		
<i>Se pide cerrar sesión y se borra el token de acceso.</i>		
<b>Curso Normal (Básico)</b>		
<b>1</b>	Usuario: Pide cerrar sesión.	<b>2</b> Sistema: Elimina el token de acceso.
<b>3</b>	Sistema: Muestra mensaje de éxito.	<b>4</b>
<b>Cursos Alternos</b>		
<b>Comentarios</b>		
En el caso de que el token hubiera sido borrado antes manualmente o hubiera expirado, el curso sería igual.		

Tabla 6: CU-3: 'Cerrar Sesión'.

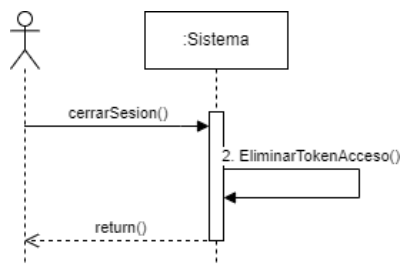


Figura 4: Diagrama de Interacción de CU-3: 'Cerrar Sesión'.

Caso de Uso	Crear Asignatura			CU-4
Actor	Usuario			
Tipo	Primario Real			
Referencias	RF4	CU-2		
Precondición	El usuario debe haber iniciado sesión en el sistema.			
Postcondición	El usuario crea una asignatura.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Crear una asignatura con los datos proporcionados.				
Resumen				
Se piden datos de la asignatura al usuario y se crea la asignatura.				
Curso Normal (Básico)				
1	Sistema: Pide nombre de la asignatura.		2	Usuario: Proporciona datos.
3	Sistema: Añade asignatura			
Cursos Alternos				
2a	Usuario: Proporciona datos no válidos.			
2b	Sistema: Muestra un mensaje de aviso de la no validez de los datos y pide datos de nuevo.			
Comentarios				
Puesto que el único dato que se pide es el nombre de la asignatura y este se puede repetir, no se realiza ninguna comprobación en el sistema.				

Tabla 7: CU-4: 'Crear Asignatura'.

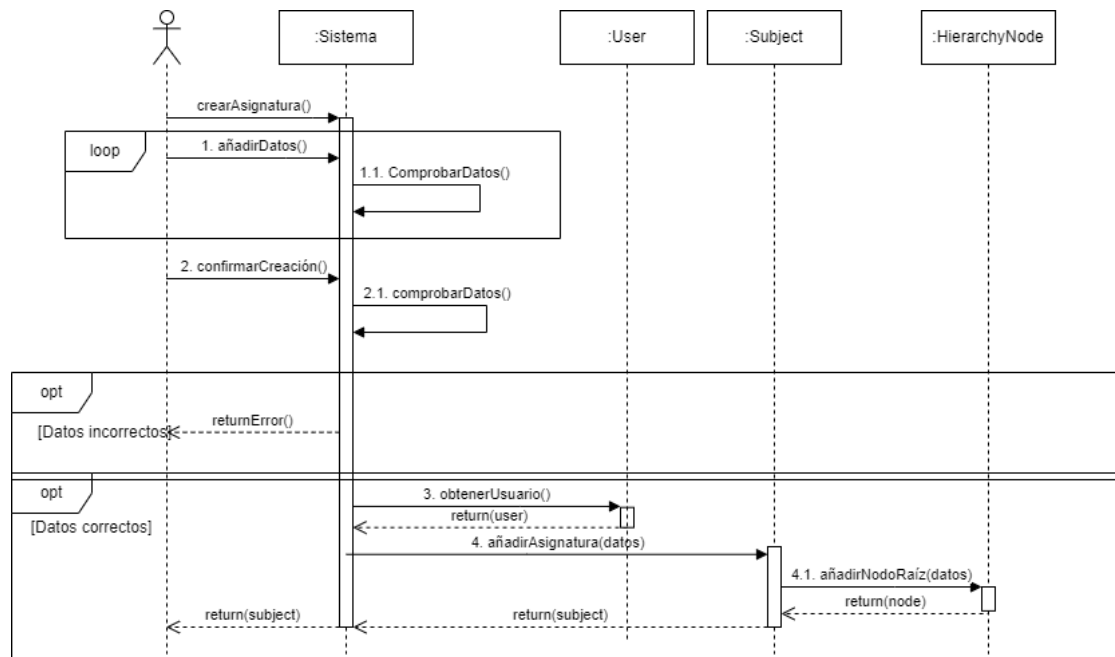


Figura 5: Diagrama de Interacción de CU-4: 'Crear Asignatura'.



Caso de Uso	Editar Asignatura			CU-5
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF5	CU-2, CU-4		
Precondición	El usuario debe de haber iniciado sesión y haber creado una asignatura.			
Postcondición	El nombre de la asignatura es modificado.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Cambiar el nombre de la asignatura.				
Resumen				
Se piden datos para actualizar la asignatura y se actualiza.				
Curso Normal (Básico)				
1	Usuario: Proporciona asignatura a editar.	2	Sistema: Comprueba existencia de la asignatura.	
3	Sistema: Pide nuevo nombre de la asignatura.	4	Usuario: Proporciona nombre de la asignatura.	
5	Sistema: Modifica la asignatura y muestra mensaje de éxito.			
Cursos Alternos				
1a	Usuario: Proporciona datos no válidos o la asignatura no existe.			
1b	Sistema: Muestra mensaje de error.			
1c	Fin de caso de uso.			
4a	Usuario: Proporciona datos no válidos.			
4b	Sistema: Muestra un mensaje de aviso y pide datos de nuevo.			
Comentarios				

Tabla 8: CU-5: 'Editar Asignatura'.

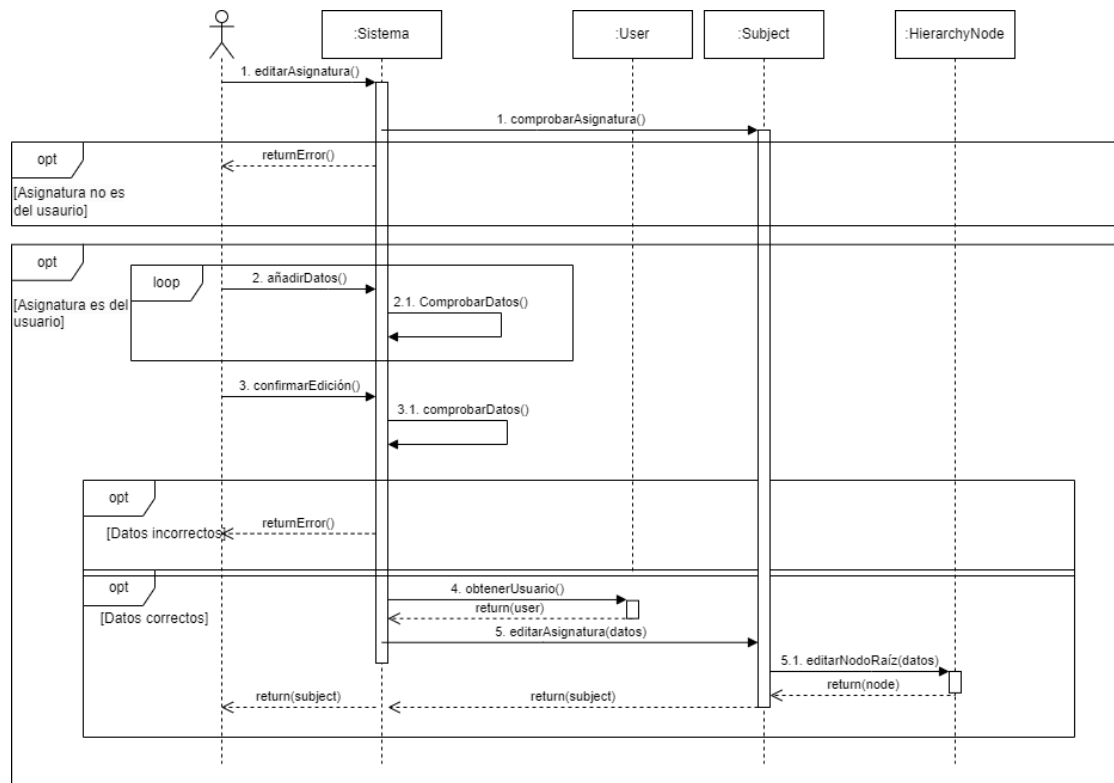


Figura 6: Diagrama de Interacción de CU-5: 'Editar Asignatura'.

Caso de Uso		Eliminar asignatura		CU-6	
Actor		Usuario			
Tipo		Secundario Real			
Referencias		RF6	CU-2, CU-4		
Precondición		El usuario debe haber iniciado sesión y haber creado una asignatura.			
Postcondición		La asignatura es eliminada del sistema.			
Autor		Rafael Delgado García-Valdecasas		Versión	1.0
Propósito					
Eliminar los datos de una asignatura.					
Resumen					
Se pide borrar una asignatura y se eliminan los recursos asociados y la asignatura.					
Curso Normal (Básico)					
1	Usuario: Proporciona asignatura a borrar.		2	Sistema: Pide confirmación del usuario.	
3	Usuario: Confirma la decisión.		4	Sistema: Comprueba los datos proporcionados.	
5	Sistema: Elimina los recursos asociados.		6	Sistema: Muestra un mensaje de éxito.	
Cursos Alternos					
3a	Usuario: Cancela el proceso.				
3b	Fin del Caso de Uso.				
4a	Sistema: No encuentra la asignatura o no pertenece al usuario.				
4b	Sistema: Muestra un mensaje de error.				
4c	Fin del Caso de Uso.				
5a	Sistema: No encuentra recursos asociados a la asignatura.				
5b	Sistema: Muestra un mensaje de éxito				
5c	Fin del Caso de Uso.				
Comentarios					

Tabla 9: CU-6: 'Eliminar Asignatura'.

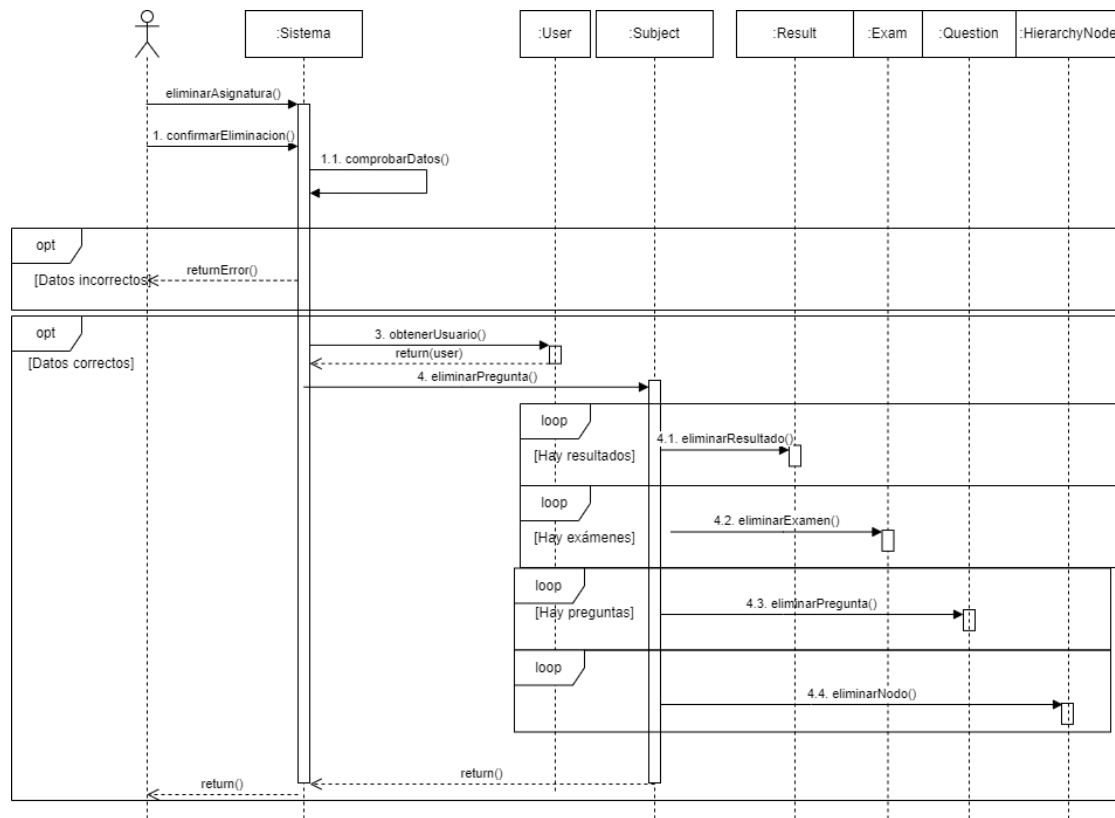


Figura 7: Diagrama de Interacción de CU-6: 'Eliminar Asignatura'.

Caso de Uso	Listar Asignaturas			CU-7
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF7	CU-2		
Precondición	El usuario debe de haber iniciado sesión.			
Postcondición	El usuario ve la lista de sus asignaturas.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Acceder a un resumen de las asignaturas.				
Resumen				
Se piden las asignaturas del usuario y se muestran.				
Curso Normal (Básico)				
1	Usuario: Pide lista de sus asignatu- ras.		2	Sistema: Muestra las asignaturas del usuario.
Cursos Alternos				
2a	Sistema: No encuentra asignaturas propias del usuario.			
2b	Sistema: Muestra un mensaje de aviso.			
2c	Fin del caso de uso.			
Comentarios				

Tabla 10: CU-7: 'Listar Asignaturas'.

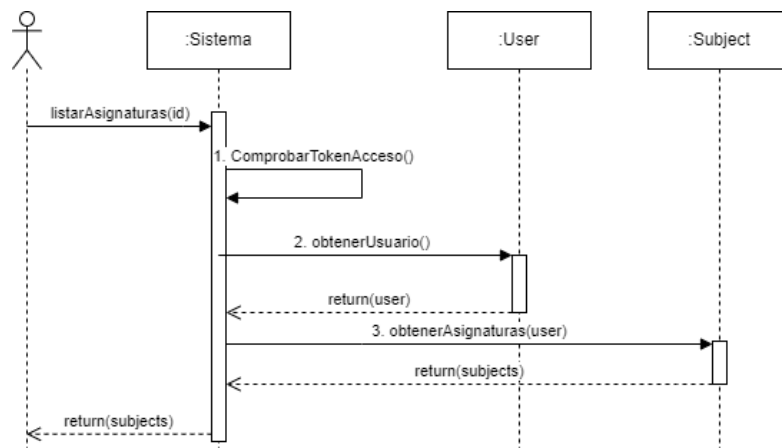


Figura 8: Diagrama de Interacción de CU-7: 'Listar Asignaturas'.

Caso de Uso	Ver Asignatura			CU-8
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF8	CU-2, CU-4		
Precondición	El usuario debe haber iniciado sesión y creado una asignatura.			
Postcondición	El usuario ve detalles de su asignatura.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Ver detalles de una asignatura concreta del usuario.				
Resumen				
El usuario pide ver una asignatura y se le muestra.				
Curso Normal (Básico)				
1	Usuario: Pide ver una asignatura del listado de asignaturas.		2	Sistema: Comprueba los datos proporcionados.
3	Sistema: Muestra los datos de la asignatura.			
Cursos Alternos				
2a	Sistema: Detecta que la asignatura no existe o no pertenece al usuario.			
2b	Sistema: Muestra un mensaje de error.			
2c	Fin del Caso de Uso.			
Comentarios				

Tabla 11: CU-8: 'Ver Asignatura'.

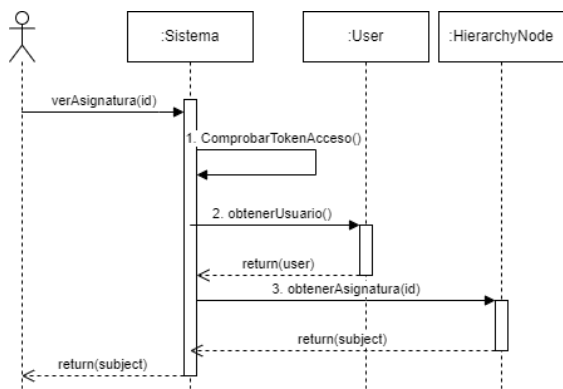


Figura 9: Diagrama de Interacción de CU-8: 'Ver Asignatura'.

Caso de Uso		Crear Nodo		CU-9	
Actor		Usuario			
Tipo		Primario Real			
Referencias		RF9	CU-2, CU-4		
Precondición		El usuario debe haber iniciado sesión en el sistema y creado una asignatura.			
Postcondición		El usuario crea un nodo.			
Autor		Rafael Delgado García-Valdecasas		Versión	1.0
Propósito					
Crear un nodo con los datos proporcionados.					
Resumen					
Se piden datos del nodo y se crea el nodo.					
Curso Normal (Básico)					
1	Sistema: Pide datos para el nodo.			2	Usuario: Proporciona datos.
3	Sistema: Comprueba datos proporcionados.			4	Sistema: Crea nodo y muestra mensaje de éxito.
Cursos Alternos					
2a	Usuario: Proporciona datos no válidos.				
2b	Sistema: Muestra un mensaje de aviso de la no validez de los datos y pide datos de nuevo.				
3a	Sistema: Detecta datos no válidos o el nodo padre no existe.				
3b	Sistema: Muestra mensaje de error.				
3c	Fin de Caso de Uso.				
Comentarios					

Tabla 12: CU-9: 'Crear Nodo'.

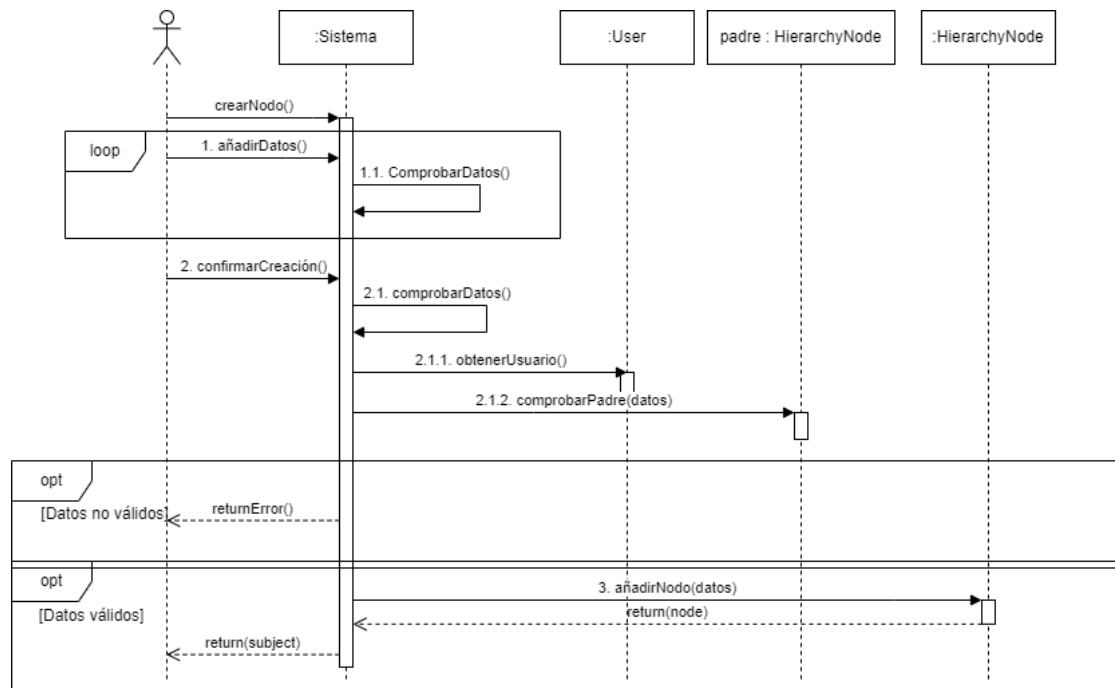


Figura 10: Diagrama de Interacción de CU-9: 'Crear Nodo'.



Caso de Uso	Editar Nodo			CU-10
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF10	CU-2, CU-4, CU-9		
Precondición	El usuario debe de haber iniciado sesión y haber creado una asignatura y un nodo.			
Postcondición	El nombre del nodo es modificado.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Cambiar el nombre del nodo.				
Resumen				
Se piden datos para actualizar el nodo y se actualiza.				
Curso Normal (Básico)				
1	Usuario: Proporciona nodo a editar.	2	Sistema: Comprueba existencia del nodo.	
3	Sistema: Pide nuevo nombre del nodo.	4	Usuario: Proporciona nombre del nodo.	
5	Sistema: Modifica el nodo y muestra mensaje de éxito.			
Cursos Alternos				
1a	Usuario: Proporciona un nodo inexistente o de otro usuario.			
1b	Sistema: Muestra mensaje de error.			
1c	Fin de caso de uso.			
4a	Usuario: Proporciona datos no válidos.			
4b	Sistema: Muestra un mensaje de aviso y pide datos de nuevo.			
Comentarios				

Tabla 13: CU-10: 'Editar Nodo'.

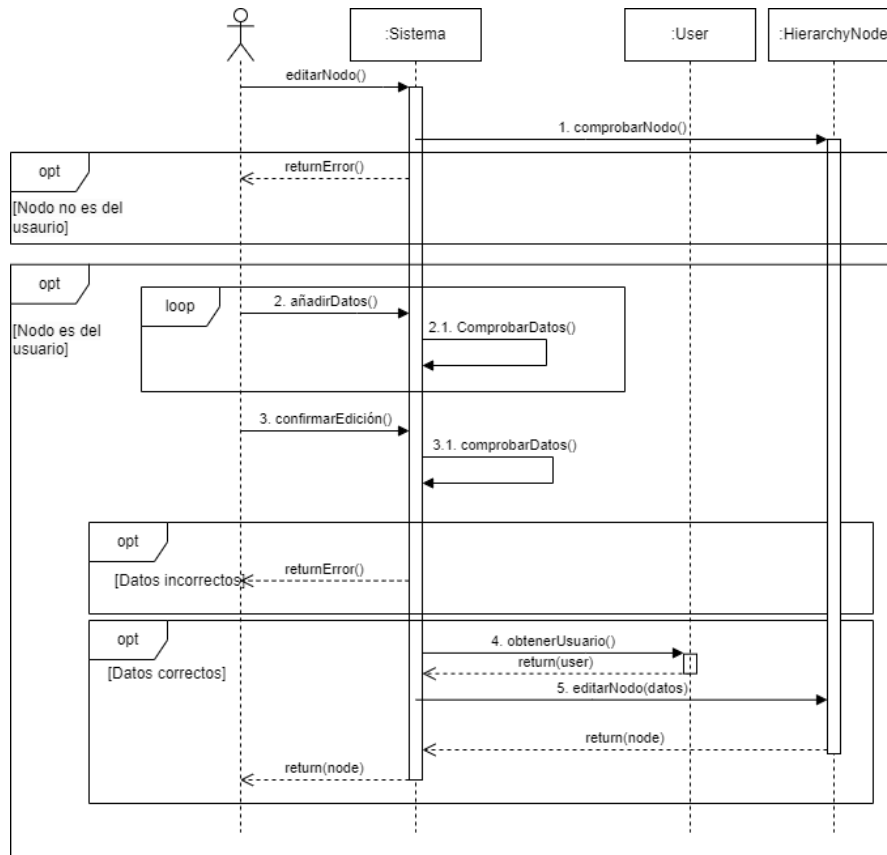


Figura 11: Diagrama de Interacción de CU-10: 'Editar Nodo'.

Caso de Uso		Eliminar nodo		CU-11	
Actor		Usuario			
Tipo		Secundario Real			
Referencias		RF11	CU-2, CU-4, CU-9		
Precondición		El usuario debe haber iniciado sesión y creado una asignatura y un nodo.			
Postcondición		El nodo es eliminado del sistema.			
Autor		Rafael Delgado García-Valdecasas		Versión	1.0
Propósito					
Eliminar los datos de un nodo de una asignatura.					
Resumen					
Se pide borrar un nodo y se eliminan los recursos asociados al mismo y el nodo.					
Curso Normal (Básico)					
1	Usuario: Proporciona nodo a borrar.		2	Sistema: Pide confirmación del usuario.	
3	Usuario: Confirma la decisión.		4	Sistema: Comprueba los datos proporcionados.	
5	Sistema: Elimina los recursos asociados.		6	Sistema: Muestra un mensaje de éxito.	
Cursos Alternos					
3a	Usuario: Cancela el proceso.				
3b	Fin del Caso de Uso.				
4a	Sistema: Detecta que el nodo no existe, no se puede eliminar por no ser un nodo hoja o por tener recursos asociados.				
4b	Sistema: Muestra un mensaje de error.				
4c	Fin del Caso de Uso.				
5a	Sistema: No encuentra recursos asociados al nodo.				
5b	Sistema: Muestra un mensaje de éxito				
5c	Fin del Caso de Uso.				
Comentarios					

Tabla 14: CU-11: 'Eliminar Nodo'.

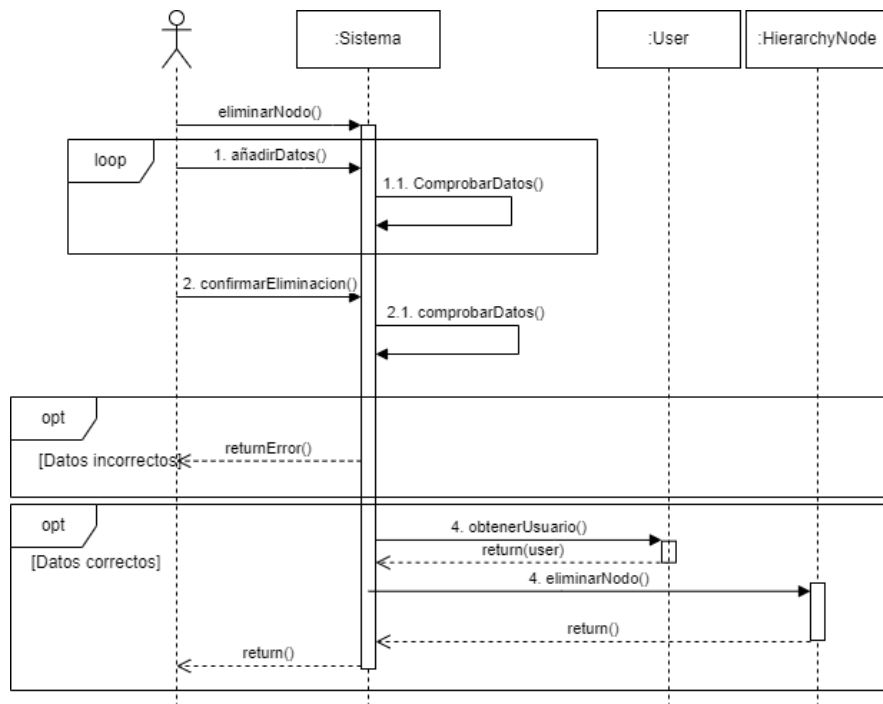


Figura 12: Diagrama de Interacción de CU-11: 'Eliminar Nodo'.

Caso de Uso		Listar Nodos		CU-12
Actor		Usuario		
Tipo		Secundario Real		
Referencias		RF12	CU-2, CU-4	
Precondición		El usuario debe de haber iniciado sesión y creado una asignatura.		
Postcondición		El usuario ve la lista de nodos de una de sus asignaturas.		
Autor		Rafael Delgado García-Valdecasas	Versión	1.0
Propósito				
Acceder al árbol de nodos de una de las asignaturas.				
Resumen				
Se piden los nodos de una asignatura del usuario y se muestran.				
Curso Normal (Básico)				
1	Usuario: Proporciona una asignatura.	2	Sistema: Comprueba los datos dados.	
3	Sistema: Muestra los nodos de la asignatura y un mensaje de éxito.			
Cursos Alternos				
2a	Sistema: Detecta que la asignatura es inexistente o de otro usuario.			
2b	Sistema: Muestra un mensaje de error.			
2c	Fin del caso de uso.			
3a	Sistema: No encuentra nodos propios del usuario.			
3b	Sistema: Muestra un mensaje de aviso.			
3c	Fin del caso de uso.			
Comentarios				

Tabla 15: CU-12: 'Listar Nodos'.

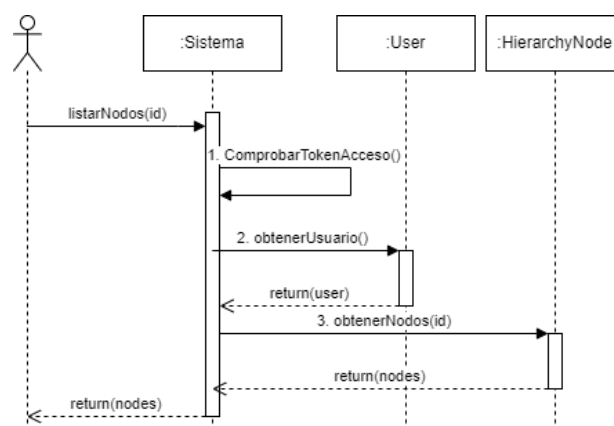


Figura 13: Diagrama de Interacción de CU-12: 'Listar Nodos'.

Caso de Uso	Ver Nodo		CU-13
Actor	Usuario		
Tipo	Secundario Real		
Referencias	RF13	CU-2, CU-4, CU-9	
Precondición	El usuario debe haber iniciado sesión y creado una asignatura y un nodo.		
Postcondición	El usuario ve los detalles de un nodo de la jerarquía.		
Autor	Rafael Delgado García-Valdecasas	Versión	1.0
Propósito			
Ver detalles de un nodo concreto de una asignatura del usuario.			
Resumen			
El usuario pide ver un nodo de su asignatura y se le muestra.			
Curso Normal (Básico)			
1	Usuario: Pide ver un nodo de la lista de nodos.	2	Sistema: Comprueba los datos proporcionados.
3	Sistema: Muestra los datos del nodo.		
Cursos Alternos			
2a	Sistema: Detecta que el nodo no existe o no es del usuario.		
2b	Sistema: Muestra un mensaje de error.		
2c	Fin del Caso de Uso.		
Comentarios			

Tabla 16: CU-13: 'Ver Nodo'.

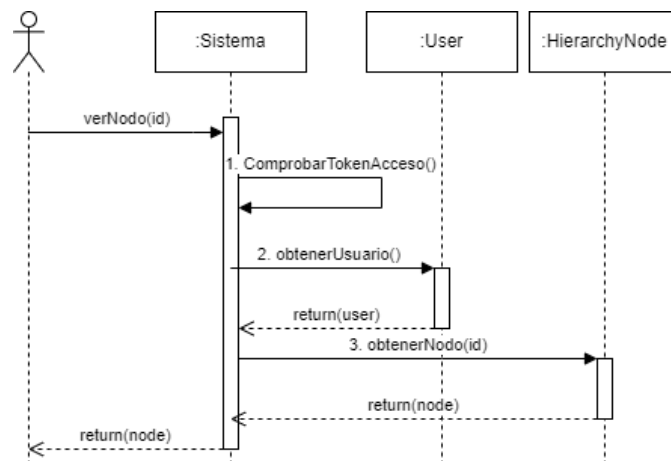


Figura 14: Diagrama de Interacción de CU-13: 'Ver Nodo'.

Caso de Uso	Crear Pregunta			CU-14
Actor	Usuario			
Tipo	Primario Real			
Referencias	RF14	CU-2, CU-4, CU-9		
Precondición	El usuario debe haber iniciado sesión en el sistema y creado una asignatura y un nodo.			
Postcondición	El usuario crea una pregunta.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Crear una pregunta con los datos proporcionados.				
Resumen				
Se piden datos de la pregunta y se crea.				
Curso Normal (Básico)				
1	Sistema: Pide datos para la pregunta.		2	Usuario: Proporciona datos.
3	Sistema: Comprueba datos proporcionados.		4	Sistema: Crea pregunta y muestra mensaje de éxito.
Cursos Alternos				
2a	Usuario: Proporciona datos no válidos.			
2b	Sistema: Muestra un mensaje de aviso de la no validez de los datos y pide datos de nuevo.			
3a	Sistema: detecta datos inexistentes o no válidos (por ejemplo, asociación a nodos no existentes o que no son del usuario).			
3b	Sistema: Muestra mensaje de error.			
3c	Fin de Caso de Uso.			
Comentarios				

Tabla 17: CU-14: 'Crear Pregunta'.

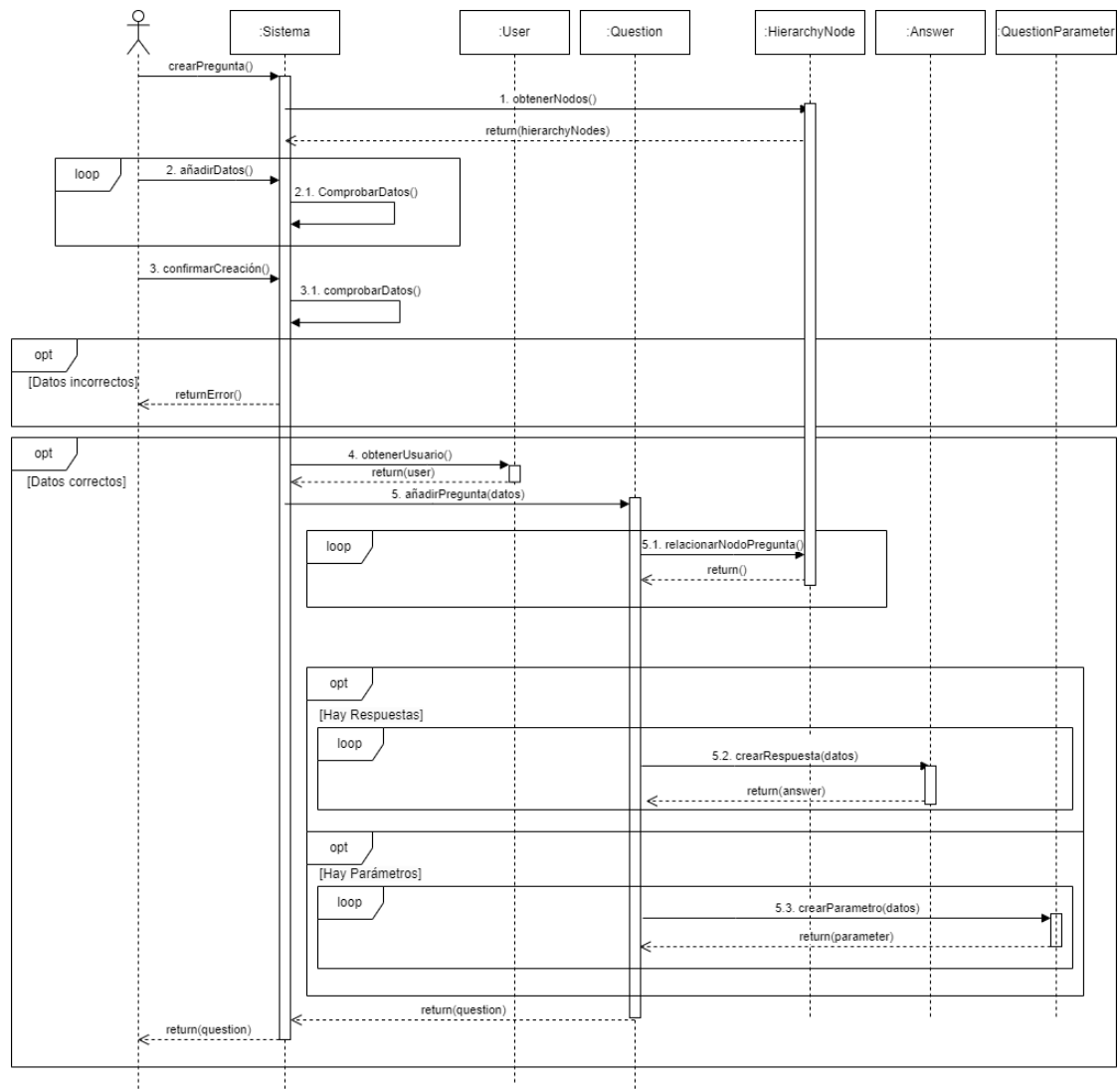


Figura 15: Diagrama de Interacción de CU-14: 'Crear Pregunta'.



Caso de Uso	Editar Pregunta			CU-15
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF15	CU-2, CU-4, CU-9, CU-14		
Precondición	El usuario debe de haber iniciado sesión y haber creado una asignatura, un nodo y una pregunta sin recursos asociados.			
Postcondición	Los datos de la pregunta son modificados.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Cambiar los datos de una pregunta sin recursos asociados.				
Resumen				
Se piden datos para actualizar la pregunta y se actualiza.				
Curso Normal (Básico)				
1	Usuario: Proporciona pregunta a editar.	2	Sistema: Comprueba existencia y conectividad de la pregunta.	
3	Sistema: Pide nuevos datos de la pregunta.	4	Usuario: Proporciona nuevos datos de la pregunta.	
5	Sistema: Modifica la pregunta y muestra mensaje de éxito.			
Cursos Alternos				
1a	Usuario: Proporciona datos no válidos o una pregunta no existente o con recursos asociados.			
1b	Sistema: Muestra mensaje de error.			
1c	Fin de caso de uso.			
4a	Usuario: Proporciona datos no válidos (por ejemplo, asociación a nodos no existentes o que no son del usuario).			
4b	Sistema: Muestra un mensaje de aviso y pide datos de nuevo.			
Comentarios				

Tabla 18: CU-15: 'Editar Pregunta'.

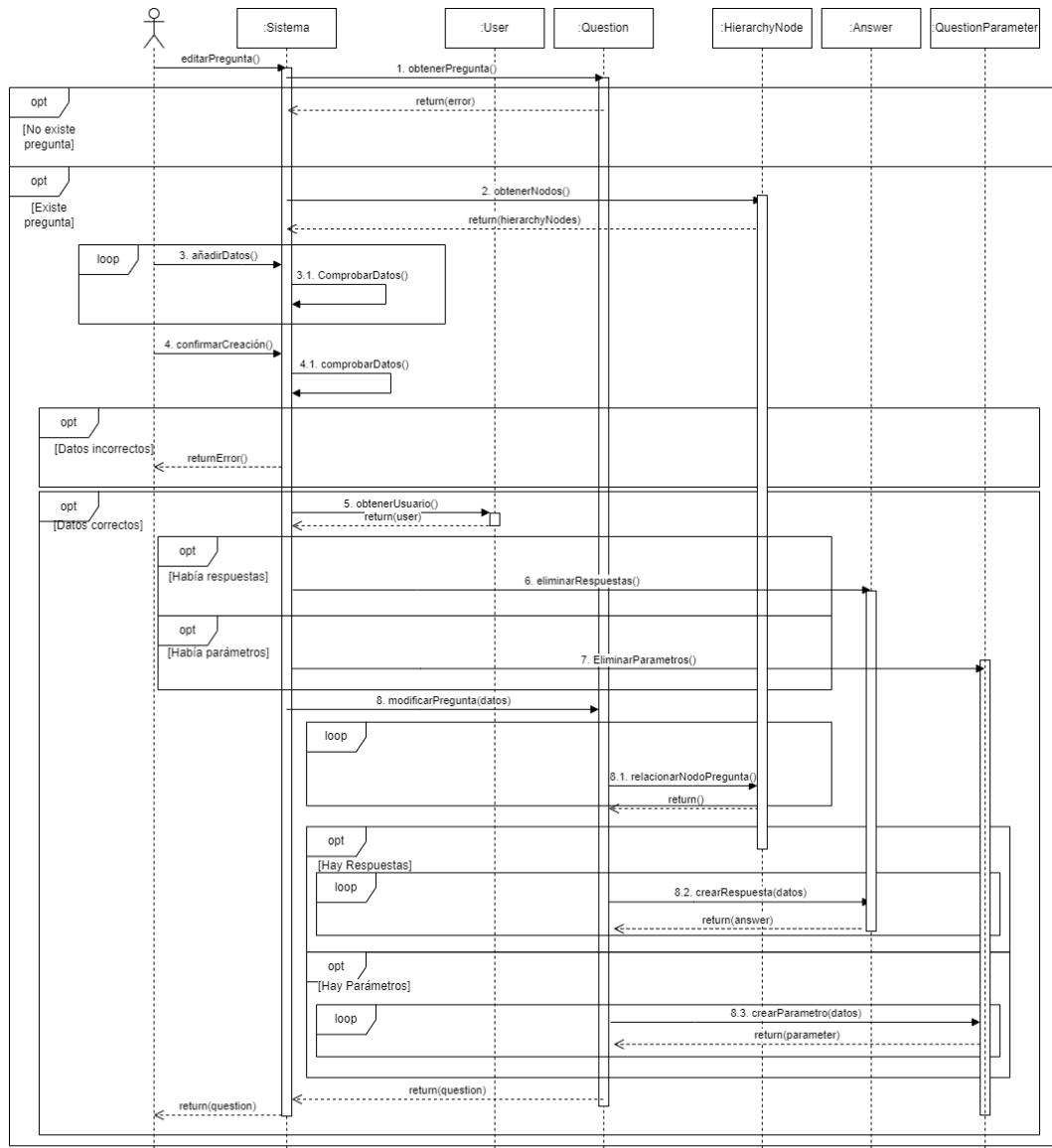


Figura 16: Diagrama de Interacción de CU-15: 'Editar Pregunta'.

Caso de Uso		Eliminar pregunta		CU-16
Actor		Usuario		
Tipo		Secundario Real		
Referencias		RF16	CU-2, CU-4, CU-9, CU-14	
Precondición		El usuario debe haber iniciado sesión y creado una asignatura, un nodo y una pregunta.		
Postcondición		La pregunta es eliminada del sistema.		
Autor		Rafael Delgado García-Valdecasas	Versión	1.0
Propósito				
Eliminar los datos de una pregunta de una asignatura.				
Resumen				
Se pide borrar una pregunta y se eliminan los recursos asociados a la misma y la pregunta.				
Curso Normal (Básico)				
1	Usuario: Proporciona pregunta a borrar.	2	Sistema: Pide confirmación del usuario.	
3	Usuario: Confirma la decisión.	4	Sistema: Comprueba los datos proporcionados.	
5	Sistema: Elimina los recursos asociados.	6	Sistema: Muestra un mensaje de éxito.	
Cursos Alternos				
3a	Usuario: Cancela el proceso.			
3b	Fin del Caso de Uso.			
4a	Sistema: No encuentra la pregunta, no es del usuario o tiene exámenes o resultados asociados.			
4b	Sistema: Muestra un mensaje de error.			
4c	Fin del Caso de Uso.			
5a	Sistema: No encuentra recursos asociados a la pregunta.			
5b	Sistema: Muestra un mensaje de éxito.			
5c	Fin del Caso de Uso.			
Comentarios				

Tabla 19: CU-16: 'Eliminar Pregunta'.

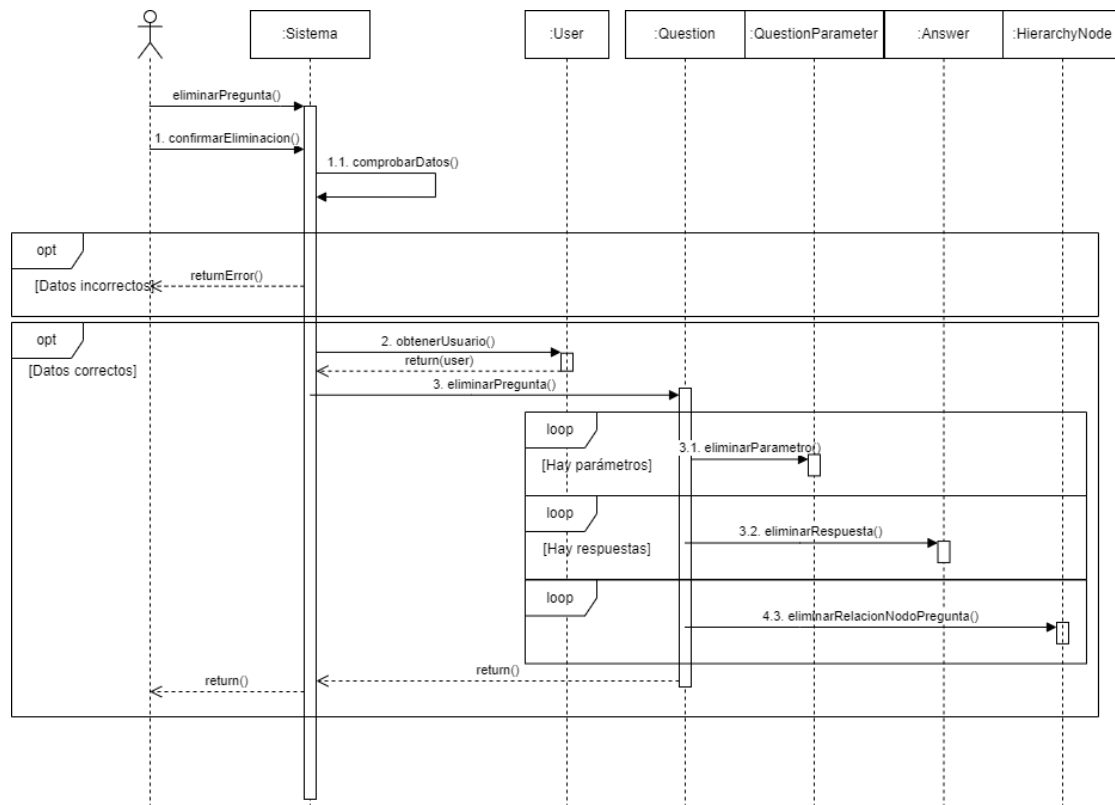


Figura 17: Diagrama de Interacción de CU-16: 'Eliminar Pregunta'.

Caso de Uso	Listar Preguntas			CU-17
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF17	CU-2, CU-4		
Precondición	El usuario debe de haber iniciado sesión y creado una asignatura.			
Postcondición	El usuario ve la lista de preguntas de una de sus asignaturas.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Acceder a la lista de preguntas de una de las asignaturas.				
Resumen				
Se piden las preguntas de una asignatura del usuario y se muestran.				
Curso Normal (Básico)				
1	Usuario: Proporciona una asignatura.	2	Sistema: Comprueba los datos dados.	
3	Sistema: Muestra las preguntas de una asignatura y muestra un mensaje de éxito.			
Cursos Alternos				
2a	Sistema: Detecta que la asignatura no existe o no es del usuario.			
2b	Sistema: Muestra un mensaje de error.			
2c	Fin del caso de uso.			
3a	Sistema: No encuentra preguntas propias del usuario.			
3b	Sistema: Muestra un mensaje de aviso.			
3c	Fin del caso de uso.			
Comentarios				

Tabla 20: CU-17: 'Listar Preguntas'.

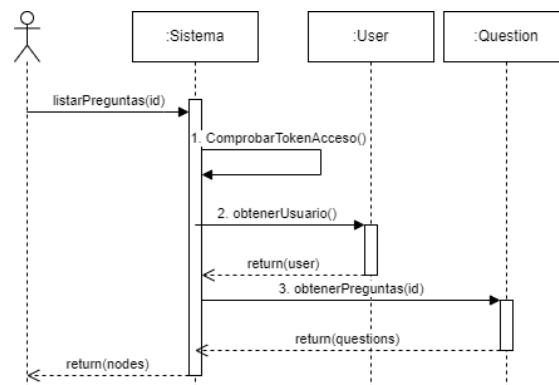


Figura 18: Diagrama de Interacción de CU-17: 'Listar Preguntas'.

Caso de Uso	Ver Pregunta		CU-18
Actor	Usuario		
Tipo	Secundario Real		
Referencias	RF18	CU-2, CU-4, CU-9, CU-14	
Precondición	El usuario debe haber iniciado sesión y creado una asignatura, un nodo y una pregunta.		
Postcondición	El usuario ve detalles de una pregunta de la asignatura.		
Autor	Rafael Delgado García-Valdecasas	Versión	1.0
Propósito			
Ver detalles de una pregunta concreta de una asignatura del usuario.			
Resumen			
El usuario pide ver una pregunta de su asignatura y se le muestra.			
Curso Normal (Básico)			
1	Usuario: Pide ver una pregunta del listado de preguntas.	2	Sistema: Comprueba los datos proporcionados.
3	Sistema: Muestra los datos de la pregunta.		
Cursos Alternos			
2a	Sistema: Detecta que la pregunta es inexistente o no es del usuario.		
2b	Sistema: Muestra un mensaje de error.		
2c	Fin del Caso de Uso.		
Comentarios			

Tabla 21: CU-18: 'Ver Pregunta'.

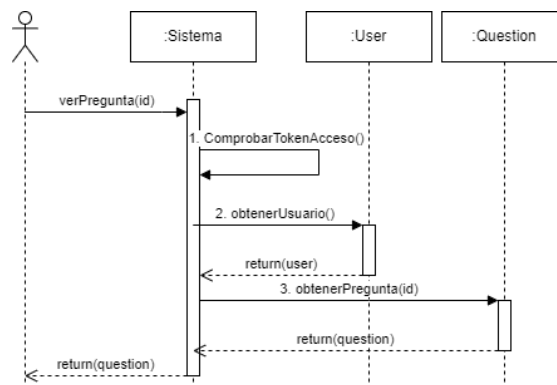


Figura 19: Diagrama de Interacción de CU-18: 'Ver Pregunta'.

Caso de Uso	Importar Banco de Preguntas			CU-19
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF19	CU-2, CU-4		
Precondición	El usuario debe haber iniciado sesión y creado una asignatura.			
Postcondición	Se importan preguntas asociadas a la asignatura.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Importar un banco de preguntas para dar más facilidad.				
Resumen				
Se pide un archivo de preguntas, se obtienen las preguntas del archivo y se añaden.				
Curso Normal (Básico)				
1	Usuario: Pide asignatura a la que importar preguntas.	2	Sistema: Comprueba existencia de la asignatura.	
3	Sistema: Pide archivo a importar.	4	Usuario: Proporciona archivo.	
5	Sistema: Comprueba datos del archivo, las añade y muestra mensaje de éxito.			
Cursos Alternos				
1a	Usuario: Proporciona una asignatura no existente o de otro usuario.			
1b	Sistema: Muestra mensaje de error.			
1c	Fin de caso de uso.			
4a	Usuario: Proporciona un archivo con un formato no válido.			
4b	Sistema: Muestra un mensaje de aviso y pide datos de nuevo.			
Comentarios				

Tabla 22: CU-19: 'Importar Banco de Preguntas'.

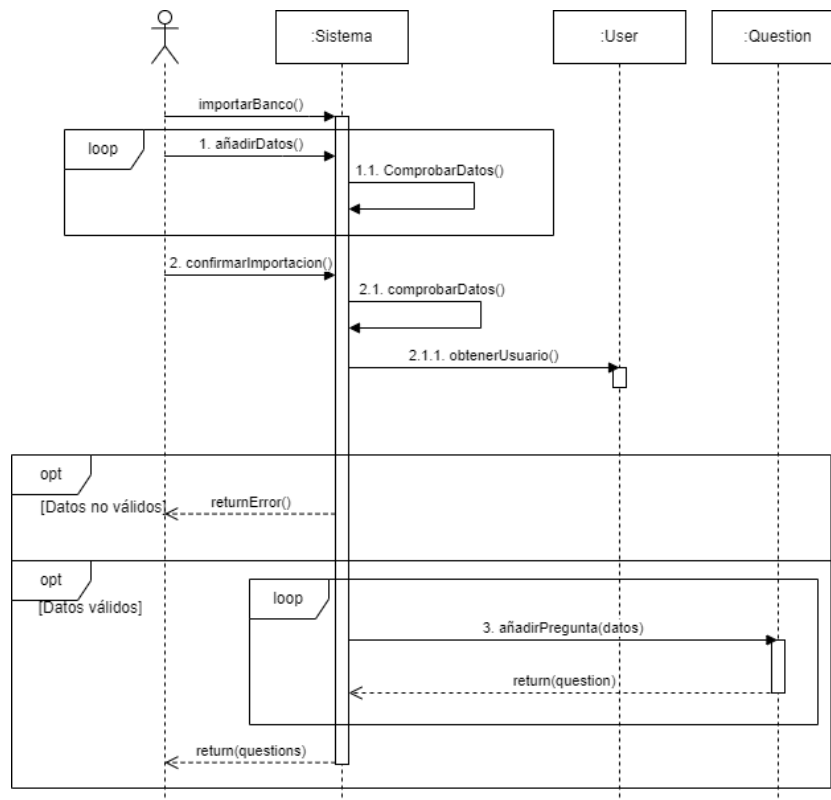


Figura 20: Diagrama de Interacción de CU-19: 'Importar Banco de preguntas'.



Caso de Uso		Desactivar pregunta		CU-20
Actor		Usuario		
Tipo		Secundario Real		
Referencias		RF20	CU-2, CU-4, CU-9, CU-14	
Precondición		El usuario debe haber iniciado sesión y creado una asignatura, un nodo y una pregunta sin recursos asociados.		
Postcondición		La pregunta es desactivada para ser usada en futuros exámenes.		
Autor		Rafael Delgado García-Valdecasas	Versión	1.0
Propósito				
Desactivar una pregunta para futuros usos.				
Resumen				
Se pide desactivar una pregunta y se desactiva.				
Curso Normal (Básico)				
1	Usuario: Proporciona pregunta a desactivar.		2	Sistema: Pide confirmación del usuario.
3	Usuario: Confirma la decisión.		4	Sistema: Comprueba los datos proporcionados.
5	Sistema: Muestra un mensaje de éxito.		6	
Cursos Alternos				
3a	Usuario: Cancela el proceso.			
3b	Fin del Caso de Uso.			
4a	Sistema: Detecta que la pregunta no existe, es de otro usuario o está ya desactivada.			
4b	Sistema: Muestra un mensaje de error.			
4c	Fin del Caso de Uso.			
Comentarios				

Tabla 23: CU-20: 'Desactivar Pregunta'.

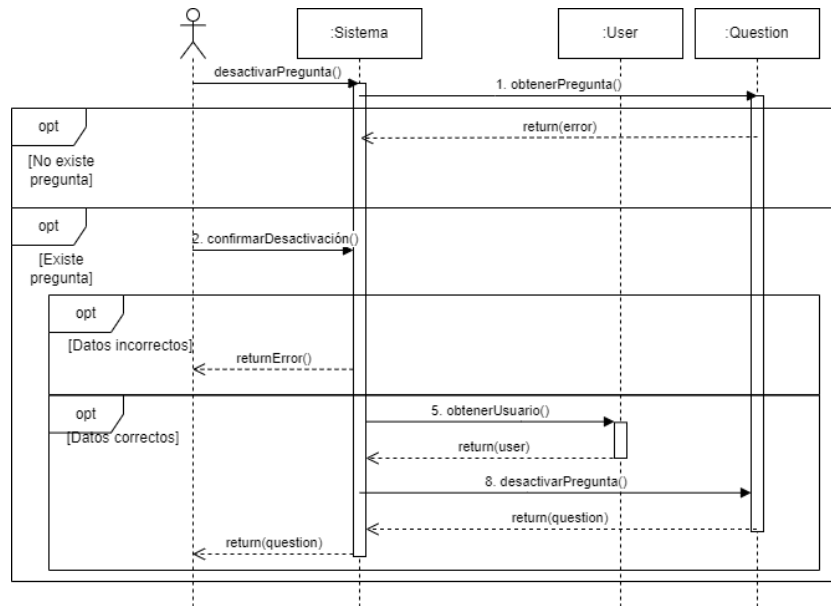


Figura 21: Diagrama de Interacción de CU-20: 'Desactivar Pregunta'.

Caso de Uso	Crear examen			CU-21
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF21	CU-2, CU-4, CU-9, CU-14		
Precondición	El usuario debe haber iniciado sesión y creado una asignatura, un nodo y una pregunta.			
Postcondición	La configuración de un examen se guarda en el sistema.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Guardar la configuración de un examen con los datos proporcionados.				
Resumen				
Se piden los datos de un examen y se crea.				
Curso Normal (Básico)				
1	Sistema: Pide datos para el examen.		2	Usuario: Proporciona datos.
3	Sistema: Comprueba datos proporcionados.		4	Sistema: Crea examen y muestra mensaje de éxito.
Cursos Alternos				
2a	Usuario: Proporciona datos no válidos.			
2b	Sistema: Muestra un mensaje de aviso de la no validez de los datos y pide datos de nuevo.			
3a	Sistema: detecta datos inexistentes o no válidos (por ejemplo, asociación a preguntas no existentes o que no son del usuario).			
3b	Sistema: Muestra mensaje de error.			
3c	Fin de Caso de Uso.			
Comentarios				

Tabla 24: CU-21: 'Crear Examen'.

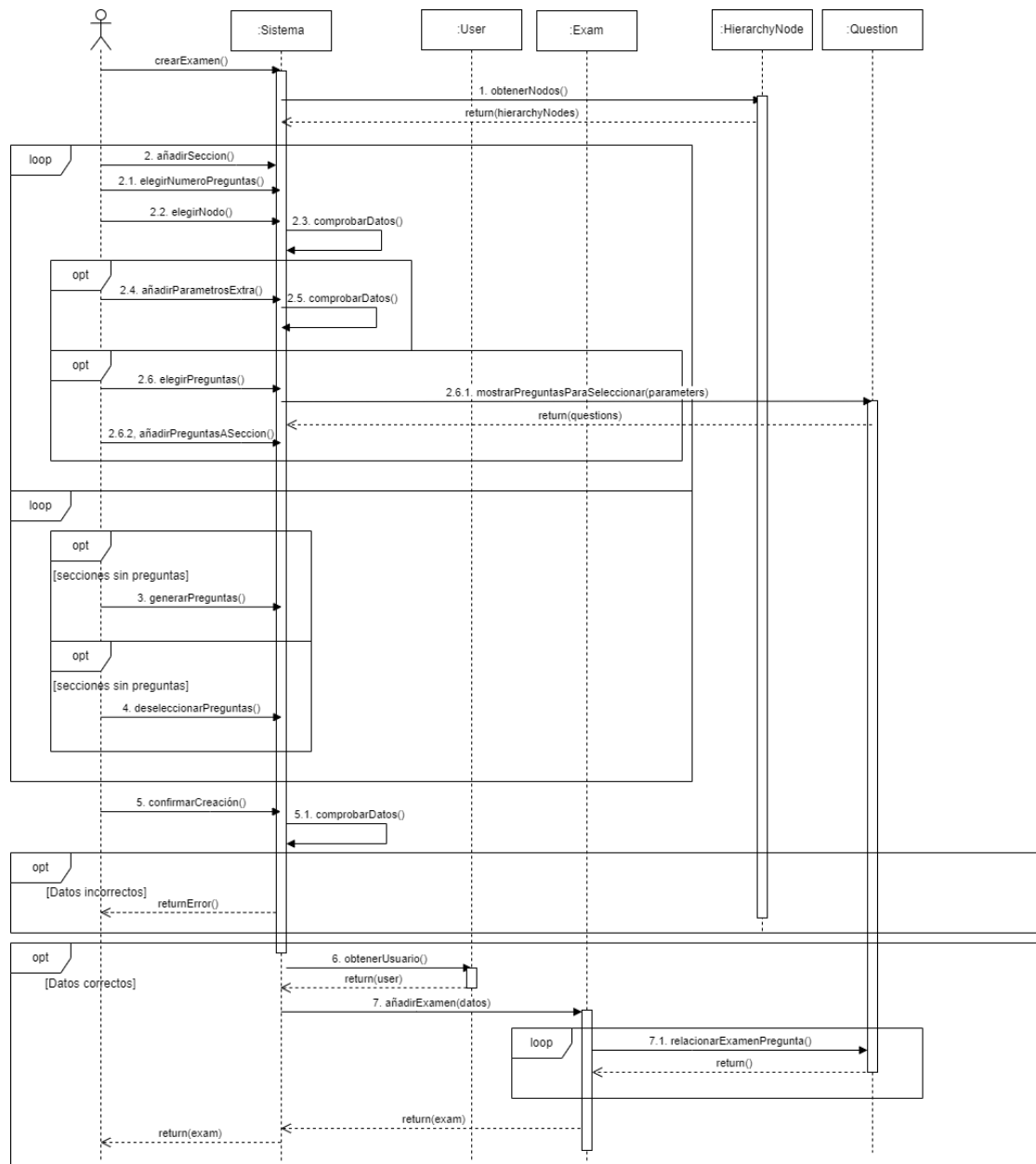


Figura 22: Diagrama de Interacción de CU-21: 'Crear Examen'.

Caso de Uso		Editar Examen		CU-22
Actor		Usuario		
Tipo		Secundario Real		
Referencias		RF22	CU-2, CU-4, CU-9, CU-14, CU-21	
Precondición		El usuario debe de haber iniciado sesión y haber creado una asignatura, un nodo, una pregunta y un examen sin resultados asociados.		
Postcondición		Los datos del examen son modificados.		
Autor		Rafael Delgado García-Valdecasas	Versión	1.0
Propósito				
Cambiar los datos de un examen sin resultados asociados.				
Resumen				
Se piden datos para actualizar el examen y se actualiza.				
Curso Normal (Básico)				
1	Usuario: Proporciona examen a editar.		2	Sistema: Comprueba existencia y conectividad del examen.
3	Sistema: Pide nuevos datos del examen.		4	Usuario: Proporciona nuevos datos del examen.
5	Sistema: Modifica el examen y muestra mensaje de éxito.			
Cursos Alternos				
1a	Usuario: Proporciona un examen no existente, de otro usuario o con resultados asociados.			
1b	Sistema: Muestra mensaje de error.			
1c	Fin de caso de uso.			
4a	Usuario: Proporciona datos no válidos.			
4b	Sistema: Muestra un mensaje de aviso y pide datos de nuevo.			
Comentarios				

Tabla 25: CU-22: 'Editar Examen'.

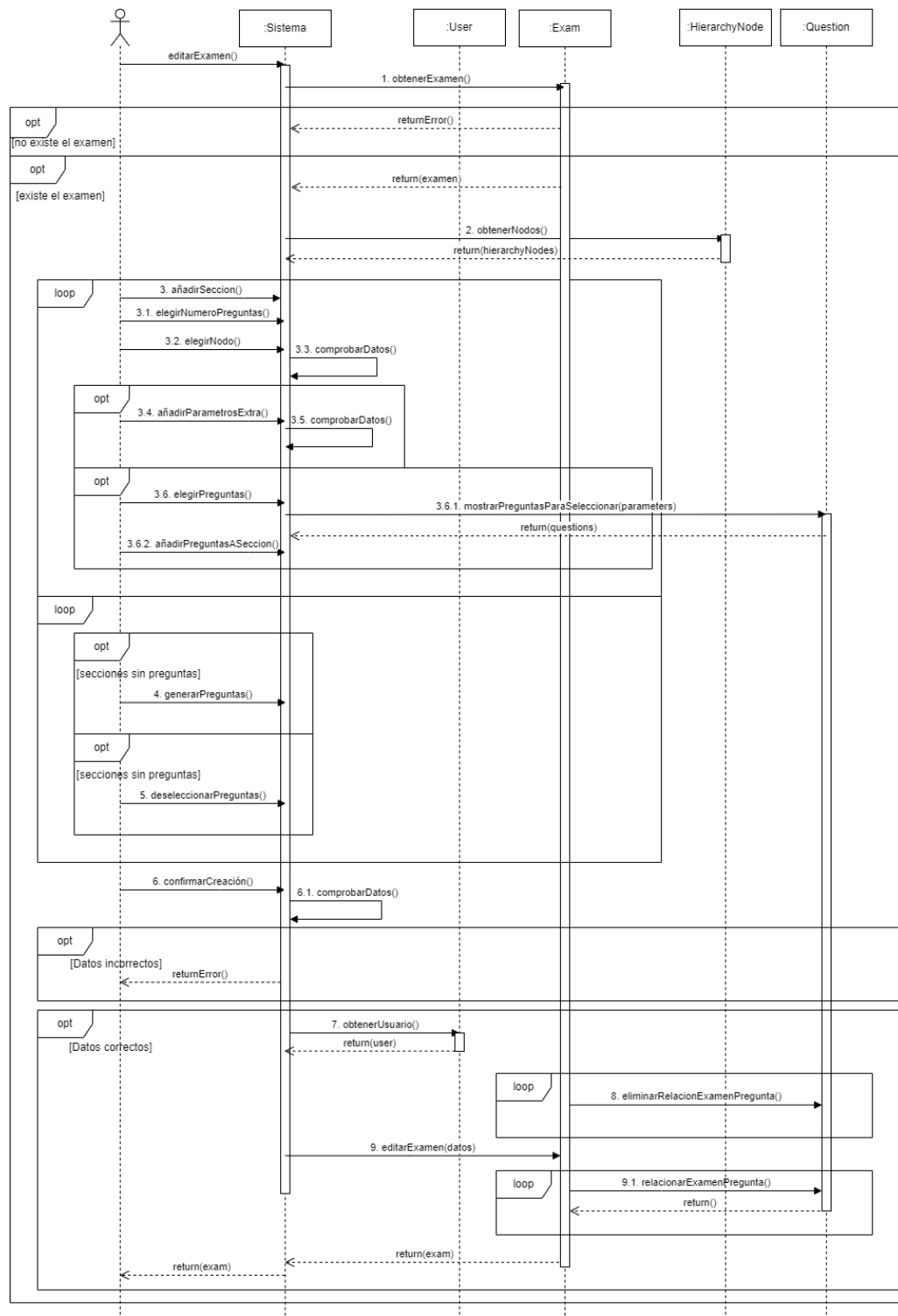


Figura 23: Diagrama de Interacción de CU-22: 'Editar Examen'.

Caso de Uso		Eliminar examen		CU-23	
Actor		Usuario			
Tipo		Secundario Real			
Referencias		RF23	CU-2, CU-4, CU-9, CU-14, CU-21		
Precondición		El usuario debe haber iniciado sesión y creado una asignatura, un nodo, una pregunta y un examen sin resultados asociados.			
Postcondición		El examen es eliminado del sistema.			
Autor		Rafael Delgado García-Valdecasas		Versión	1.0
Propósito					
Eliminar los datos de un examen de una asignatura.					
Resumen					
Se pide borrar un examen y se elimina.					
Curso Normal (Básico)					
1	Usuario: Proporciona examen a borrar.		2	Sistema: Pide confirmación del usuario.	
3	Usuario: Confirma la decisión.		4	Sistema: Comprueba los datos proporcionados.	
5	Sistema: Elimina el examen.		6	Sistema: Muestra mensaje de éxito.	
Cursos Alternos					
3a	Usuario: Cancela el proceso.				
3b	Fin del Caso de Uso.				
4a	Sistema: Detecta datos no válidos o el examen es inexistente, de otro usuario o con resultados asociados.				
4b	Sistema: Muestra un mensaje de error.				
4c	Fin del Caso de Uso.				
Comentarios					

Tabla 26: CU-23: 'Eliminar Examen'.

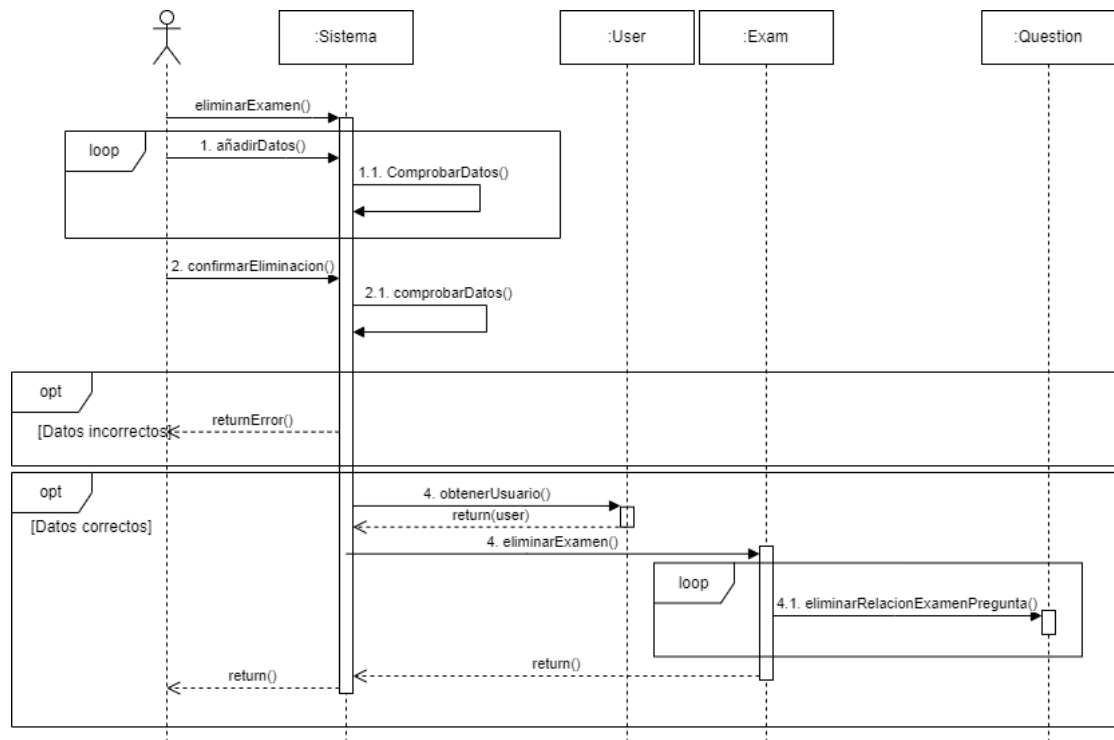


Figura 24: Diagrama de Interacción de CU-23: 'Eliminar Examen'.



Caso de Uso	Listar Exámenes			CU-24
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF24	CU-2, CU-4		
Precondición	El usuario debe de haber iniciado sesión y creado una asignatura.			
Postcondición	El usuario ve la lista de exámenes de una de sus asignaturas.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Acceder a la lista de exámenes de una de las asignaturas.				
Resumen				
Se piden los exámenes de una asignatura del usuario y se muestran.				
Curso Normal (Básico)				
1	Usuario: Proporciona una asignatura.	2	Sistema: Comprueba los datos dados.	
3	Sistema: Muestra los exámenes de una asignatura y muestra un mensaje de éxito.			
Cursos Alternos				
2a	Sistema: Detecta que la asignatura no existe o no es del usuario.			
2b	Sistema: Muestra un mensaje de error.			
2c	Fin del caso de uso.			
3a	Sistema: No encuentra exámenes propios del usuario.			
3b	Sistema: Muestra un mensaje de aviso.			
3c	Fin del caso de uso.			
Comentarios				

Tabla 27: CU-24: 'Listar Exámenes'.

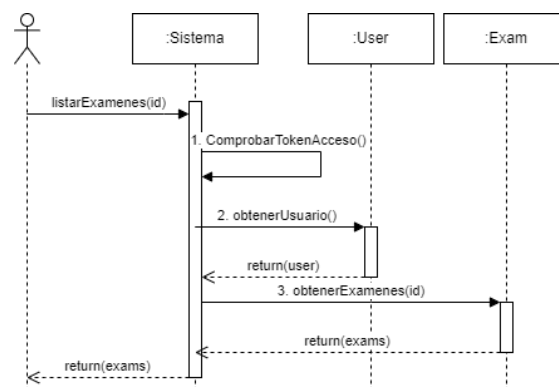


Figura 25: Diagrama de Interacción de CU-24: 'Listar Exámenes'.

Caso de Uso	Ver Examen	CU-25
Actor	Usuario	
Tipo	Secundario Real	
Referencias	RF25	CU-2, CU-4, CU-9, CU-14, CU-21
Precondición	El usuario debe haber iniciado sesión y creado una asignatura, un nodo, una pregunta y un examen.	
Postcondición	El usuario ve detalles de un examen de la asignatura.	
Autor	Rafael Delgado García-Valdecasas	Versión 1.0
Propósito		
Ver detalles de un examen concreto de una asignatura del usuario.		
Resumen		
El usuario pide ver un examen de su asignatura y se le muestra.		
Curso Normal (Básico)		
1	Usuario: Pide ver un examen del listado de exámenes.	2 Sistema: Comprueba los datos proporcionados.
3	Sistema: Muestra los datos del examen.	
Cursos Alternos		
2a	Sistema: Detecta que el examen es inexistente o de otro usuario.	
2b	Sistema: Muestra un mensaje de error.	
2c	Fin del Caso de Uso.	
Comentarios		

Tabla 28: CU-25: 'Ver Examen'.

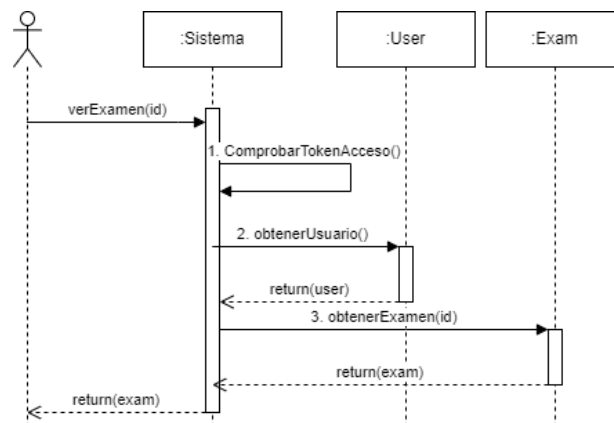


Figura 26: Diagrama de Interacción de CU-25: 'Ver Examen'.

Caso de Uso		Exportar examen		CU-26
Actor		Usuario		
Tipo		Secundario Real		
Referencias		RF26	CU-2, CU-4, CU-9, CU-14, CU-21	
Precondición		El usuario debe haber iniciado sesión y creado una asignatura, un nodo, una pregunta y un examen.		
Postcondición		Se descarga un archivo del examen.		
Autor		Rafael Delgado García-Valdecasas	Versión	1.0
Propósito				
Guardar la configuración de un examen con los datos proporcionados.				
Resumen				
Se piden los datos de un examen y se crea.				
Curso Normal (Básico)				
1	Usuario: Pide examen a exportar.	2	Sistema: Comprueba existencia del examen.	
3	Sistema: Pide formato de salida.	4	Usuario: Proporciona formato.	
5	Sistema: Genera archivo, lo añade y muestra mensaje de éxito.			
Cursos Alternos				
1a	Usuario: Proporciona un examen no existente o de otro usuario.			
1b	Sistema: Muestra mensaje de error.			
1c	Fin de caso de uso.			
4a	Usuario: Proporciona un formato que no se encuentra entre los válidos.			
4b	Sistema: Muestra un mensaje de aviso y pide datos de nuevo.			
Comentarios				

Tabla 29: CU-26: 'Exportar Examen'.

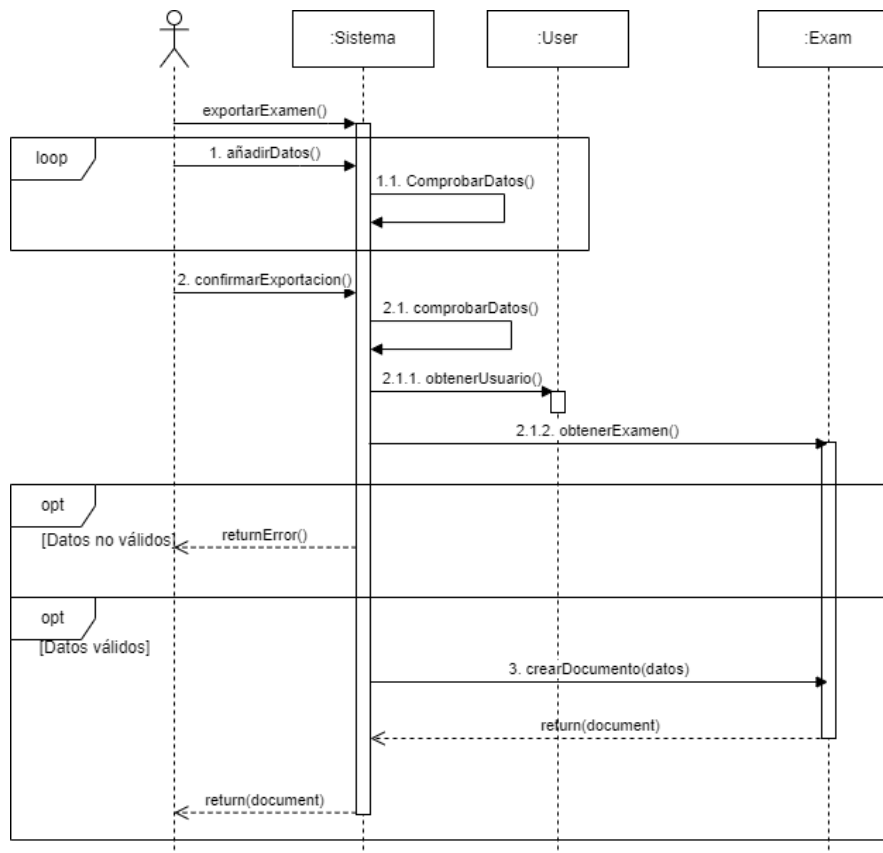


Figura 27: Diagrama de Interacción de CU-26: 'Exportar Examen'.

Caso de Uso		Importar resultados		CU-27	
Actor		Usuario			
Tipo		Secundario Real			
Referencias		RF28	CU-2, CU-4, CU-9, CU-14, CU-21		
Precondición		El usuario debe haber iniciado sesión y creado una asignatura, un nodo, una pregunta y un examen.			
Postcondición		Se importan los resultados asociados a un examen.			
Autor		Rafael Delgado García-Valdecasas		Versión	1.0
Propósito					
Guardar los resultados de un examen con los datos proporcionados.					
Resumen					
Se piden los datos de los resultados de un examen y se crea.					
Curso Normal (Básico)					
1	Usuario: Pide examen al que importar resultados.			2	Sistema: Comprueba existencia y conectividad del examen.
3	Sistema: Pide archivo a importar.			4	Usuario: Proporciona archivo.
5	Sistema: Comprueba datos del archivo, los añade y muestra mensaje de éxito.				
Cursos Alternos					
1a	Usuario: Proporciona datos un examen no existente, de otro usuario o con resultados asociados.				
1b	Sistema: Muestra mensaje de error.				
1c	Fin de caso de uso.				
4a	Usuario: Proporciona un archivo con un formato no válido.				
4b	Sistema: Muestra un mensaje de aviso y pide datos de nuevo.				
Comentarios					

Tabla 30: CU-27: 'Importar Resultados'.

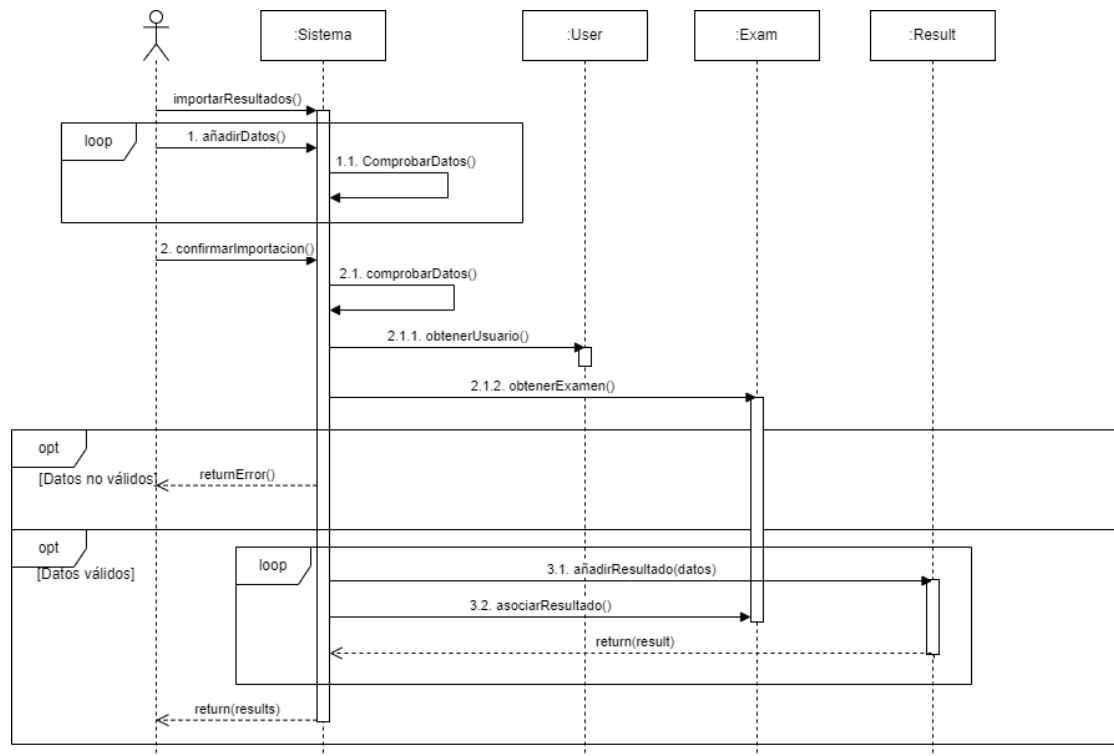


Figura 28: Diagrama de Interacción de CU-27: 'Importar Resultados'.

Caso de Uso	Eliminar resultados			CU-28
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF29	CU-2, CU-4, CU-9, CU-14, CU-21, CU-27		
Precondición	El usuario debe haber iniciado sesión y creado una asignatura, un nodo, una pregunta, un examen y haber asociado resultados.			
Postcondición	Los resultados asociados al examen son eliminados del sistema.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Eliminar los resultados de un examen de una asignatura.				
Resumen				
Se pide borrar los resultados asociados a un examen y se elimina.				
Curso Normal (Básico)				
1	Usuario: Proporciona examen al que borrar los resultados.	2	Sistema: Pide confirmación del usuario.	
3	Usuario: Confirma la decisión.	4	Sistema: Comprueba los datos proporcionados.	
5	Sistema: Elimina los resultados.	6	Sistema: Muestra mensaje de éxito.	
Cursos Alternos				
3a	Usuario: Cancela el proceso.			
3b	Fin del Caso de Uso.			
4a	Sistema: Detecta datos que el examen es inexistente, de otro usuario o sin resultados asociados.			
4b	Sistema: Muestra un mensaje de error.			
4c	Fin del Caso de Uso.			
Comentarios				

Tabla 31: CU-28: 'Eliminar Resultados'.

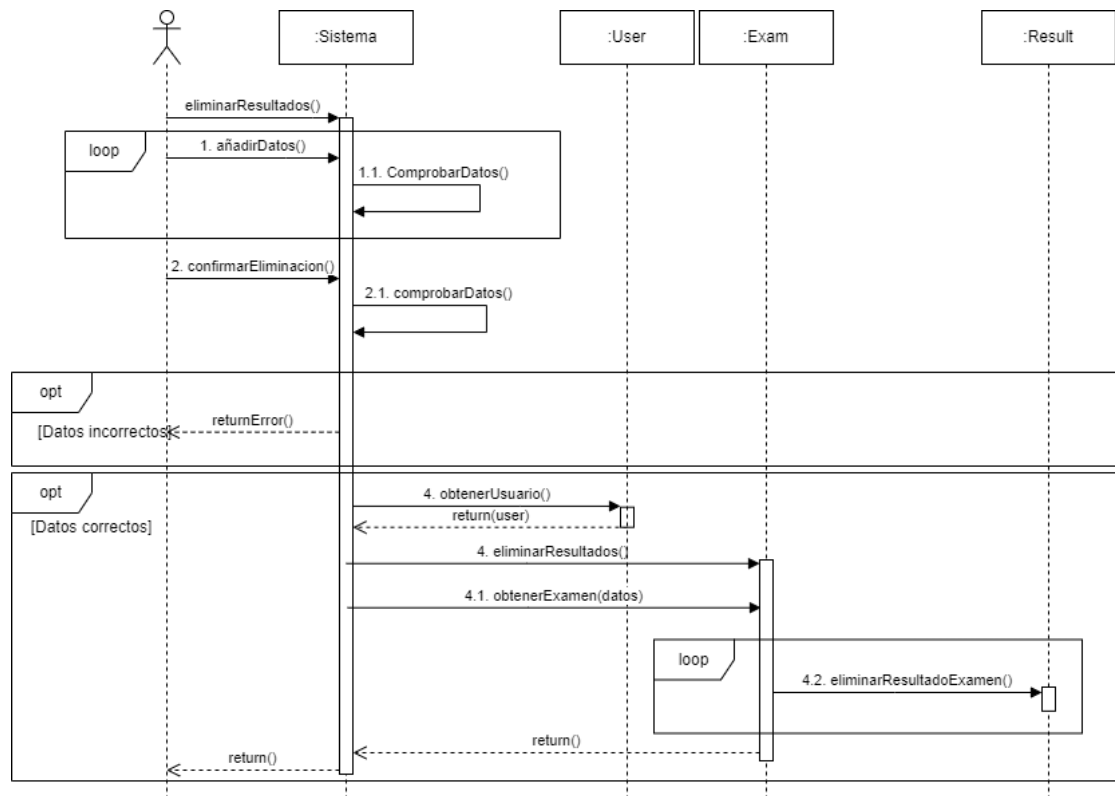


Figura 29: Diagrama de Interacción de CU-28: 'Eliminar Resultados'.



Caso de Uso	Listar Resultados			CU-29
Actor	Usuario			
Tipo	Secundario Real			
Referencias	RF30	CU-2, CU-4, CU-9, CU-14, CU-21, 24		
Precondición	El usuario debe de haber iniciado sesión y creado una asignatura, un nodo, una pregunta, un examen y haber importado resultados.			
Postcondición	El usuario ve la lista de resultados asociados a un examen de una de sus asignaturas.			
Autor	Rafael Delgado García-Valdecasas		Versión	1.0
Propósito				
Acceder a la lista de resultados asociados a un examen de una de las asignaturas.				
Resumen				
Se piden los resultados asociados a un examen del usuario y se muestran.				
Curso Normal (Básico)				
1	Usuario: Proporciona un examen.	2	Sistema: Comprueba los datos.	
3	Sistema: Muestra los resultados del examen y un mensaje de éxito.			
Cursos Alternos				
2a	Sistema: Detecta que el examen no existe o es de otro usuario.			
2b	Sistema: Muestra un mensaje de error.			
2c	Fin del caso de uso.			
3a	Sistema: No encuentra resultados asociados al examen.			
3b	Sistema: Muestra un mensaje de aviso.			
3c	Fin del caso de uso.			
Comentarios				

Tabla 32: CU-29: 'Listar Resultados'.

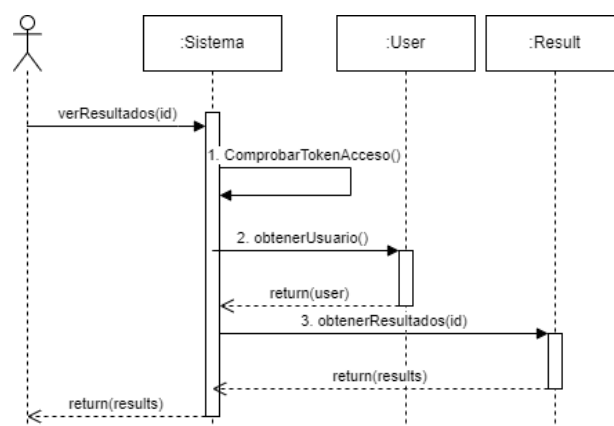


Figura 30: Diagrama de Interacción de CU-29: 'Listar Resultados'.

## 2.6. Implementación

En una primera aproximación, debido a la cercanía de esta plataforma con Moodle™, se planteó la idea de crear un módulo de Moodle™ que se pudiera instalar en una instancia de Moodle™ para dotarla de esta funcionalidad. Esto permitiría una integración muy sencilla con la plataforma, teniendo en cuenta que los usuarios de Moodle™ serían unos de los mayores beneficiarios de este proyecto. No obstante, la razón principal por la que no se realizó esta implementación fue debido a que no todo el público objetivo de este proyecto son usuarios de Moodle™, o no tienen una manera fácil de integrar un módulo de Moodle™ en la instancia que usan.

La alternativa de implementación llevada a cabo definitivamente fue una aplicación web, puesto que esto permite que los usuarios de la plataforma sean independientes a plataformas externas de aprendizaje, pero permitiendo a los usuarios que quieran complementar su plataforma externa de aprendizaje con este proyecto. Además, la implementación de una aplicación web ha sido elegida sobre alternativas como aplicaciones de escritorio o apps de móviles por varias razones, pero destacando principalmente la familiaridad de los usuarios con plataformas similares de creación de exámenes como Moodle™, así como la conexión más eficiente y sencilla con plataformas externas, llevada a cabo, como se explica en otros apartados, a través de archivos de distinto tipo tanto de entrada como de salida. Otra razón para decantarse por la alternativa de una aplicación web fue permitir un uso más cómodo en dispositivos de sobremesa, en los que se usará la gran mayoría del tiempo. A continuación, se detallan las tecnologías utilizadas y la arquitectura de la aplicación.

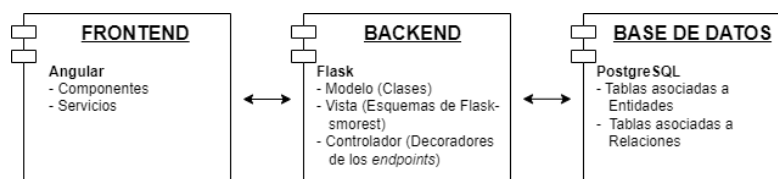


Figura 31: Diagrama de Componentes de la aplicación web.

Como se puede apreciar en el diagrama de la figura 31, la aplicación web está compuesta por las siguientes partes:

- **Backend.** El backend es una API de Python [2] creada con Flask [13]. Flask es un framework web escrito en Python. A diferencia de otros frameworks más completos, Flask es más ligero y minimalista para construir aplicaciones web sólo con los recursos esenciales. Esto hace que al desarrollar usando este framework se cuente con mayor control de las partes específicas, permitiendo mucha personalización del resto de componentes que se utilizan. Por la naturaleza de Flask, esto permite mucha simplicidad y flexibilidad, a la par que un desarrollo transparente y extensible. Las razones de uso de Flask son principalmente las mencionadas, destacando principalmente la transparencia y la simplicidad, debido a que el uso de la aplicación no va a necesitar de muchos recursos. Otra razón para utilizar Flask es la

familiaridad con el framework por proyectos anteriores. El framework de Flask se asocia al patrón de diseño de Modelo-Vista-Controlador, cuyos componentes son los siguientes:

- **Modelo:** Representa la capa de datos de la aplicación y su interacción con la base de datos. En el caso de esta aplicación, se lleva a cabo a través de la biblioteca ORM SQLAlchemy [9], que permite realizar operaciones en una base de datos SQL asegurando la persistencia de datos.
- **Vista:** Representa la capa de presentación en la que se muestran los datos. En el caso de la aplicación, se consigue a través de esquemas de Flask-smorest que formatean los datos necesarios.
- **Controlador:** Representa la capa de gestión de la lógica de la aplicación, recibiendo las entradas del usuario e interactuando con el modelo. En la aplicación, se consigue a través de los decoradores de Flask en los *endpoints*<sup>1</sup> de la aplicación, que permiten restringir el acceso a los recursos. Algunos decoradores utilizados son de autenticación, de entrada de datos y de salida de datos.

Como se puede suponer por el patrón de diseño, se pueden hacer las peticiones HTTP para obtener, crear, editar y borrar recursos. Para darle formato a las peticiones y las respuestas, se formatea el contenido en esquemas de Flask-Smorest para obligar a que los datos enviados y recibidos sean únicamente los que se piden. Para cada una de las entidades de la base de datos, se crea una clase, en la que se crean métodos estáticos para realizar las operaciones necesarias para cada clase, además de distintos esquemas de Flask-smorest en los que se formatea los datos necesarios de entrada o salida. Por último, para las entidades no débiles (usuarios, asignaturas, nodos jerárquicos, preguntas, exámenes y resultados) se añade un servicio que aloja todos los *endpoints* para realizar las operaciones. Algunos de los *endpoints* más destacados son los siguientes:

- **Crear objeto.** *Endpoints* para crear el objeto en la base de datos.
- **Listar objeto.** *Endpoints* para listar todos los objetos que cumplan una condición (normalmente, todos los objetos relacionados a un usuario o a una asignatura).
- **Ver objeto.** *Endpoints* para ver los detalles de un objeto concreto. Por ejemplo, en el caso de las preguntas, se recuperan los datos de las respuestas y de los parámetros de pregunta relacionados con la pregunta.
- **Editar objeto.** *Endpoints* para editar los detalles de un objeto concreto.
- **Borrar objeto.** *Endpoints* para borrar los detalles de un objeto concreto. Para todas las entidades, se obliga a que los objetos borrados no tengan recursos

---

<sup>1</sup>En el contexto de la aplicación, un '*endpoint*' es una petición HTTP válida que se espera en la aplicación, que permite al usuario introducir, recibir o manipular información.

asociados (por ejemplo, no se podrá borrar un examen con resultados asociados), exceptuando la asignatura, que borra todos los recursos asociados a la asignatura (nodos jerárquicos, preguntas, examen y resultados).

- **Importar banco de preguntas.** *Endpoints* para importar un banco de preguntas. En el caso de que las preguntas no estén en el formato requerido (CSV o Aiken), la importación es fallida. Además, en el caso del CSV, es necesario que el archivo cuente con las columnas ‘title’ y ‘type’, permitiendo contar con columnas ‘answerN’ y ‘pointsN’ para asociar respuestas. La decisión de los formatos de importación se tomó para permitir conexión con Moodle™ (en el caso de Aiken) y para permitir un formato más tradicional y polivalente (en el caso de CSV).
- **Obtener preguntas a seleccionar.** *Endpoint* para seleccionar preguntas en un examen. Selecciona las preguntas a añadir a una sección de un examen.
- **Obtener preguntas de exámenes anteriores.** *Endpoint* para obtener preguntas de exámenes anteriores. Esto permite que se puedan comparar con facilidad preguntas entre exámenes, para actuar en consecuencia de las repeticiones de preguntas.
- **Exportar examen.** *Endpoints* para exportar exámenes. En el caso de que no se seleccione uno de los formatos requeridos (PDF, ODT, Aiken, MoodleXML, GIFT), la exportación es fallida. Además, debido a la estructura del formato, al exportar un examen a Aiken solo se exportarán las preguntas de tipo test. La decisión de los formatos de exportación se tomó también para realizar conexión con Moodle™ (en el caso de Aiken, MoodleXML y GIFT), permitir un formato polivalente y apto para impresión (en el caso de PDF), y un formato apto para impresión que fuera editable en el caso de que el usuario quisiera editar el archivo (en el caso de ODT).
- **Importar resultados.** *Endpoint* para importar resultados. En el caso de que las preguntas no estén en CSV, la importación es fallida. La decisión de los formatos de importación se tomó para tener un formato más tradicional y polivalente, además de para mantener cierta cohesión con el resto de la aplicación.
- **Base de datos.** La API realiza peticiones a una base de datos creada en PostgreSQL® [17], que está alojado en un contenedor de Docker [14]. La idea de tener un contenedor separado del resto del backend es para contar con mayor modularización, para evitar que un fallo de la base de datos afectase a la API o viceversa. Cabe mencionar que se consideró el uso de un DBMS como Oracle®. Sin embargo, se acabó contando con PostgreSQL® debido a que este resulta ser de libre disposición y también por la familiaridad con el DBMS en proyectos anteriores.
- **Frontend.** El frontend es una aplicación de Angular [1], en la que se crean servicios para realizar peticiones a la API del backend y componentes que muestran el resultado de las peticiones hechas por los servicios. Angular es un framework

desarrollado principalmente por Google<sup>®</sup> para el desarrollo de aplicaciones web de front-end. Se basa en TypeScript, un conjunto de JavaScript que agrega algunas características para diferenciarse de este último. Angular cuenta con una arquitectura basada en componentes, dividiendo la aplicación en componentes independientes y reutilizables entre sí. A pesar de esto, Angular cuenta con una inyección de dependencias, en la que se gestionan las dependencias entre los distintos componentes y servicios. Por otra parte, también integra RxJS para gestionar eventos asíncronos, lo que permite en la aplicación realizar varias peticiones web que no tienen que esperarse entre sí. Las razones para contar finalmente con Angular, además de lo ya mencionado, son la fácil escalabilidad que suponen la independencia de componentes entre sí, la amplia comunidad que usa Angular y proporciona ayuda a través de tutoriales y documentación y por familiaridad con el framework en proyectos anteriores.

Asimismo, gran parte de los componentes están formados por componentes por defecto procedentes de Angular Material [11]. Los componentes son los siguientes:

- **Registro.** Componente para registrar a un usuario.
- **Iniciar sesión.** Componente para iniciar sesión con los datos de un usuario.
- **Crear objeto.** Componentes para crear objetos (asignatura, nodo, pregunta y examen).
- **Editar objeto.** Componentes para editar objetos (asignatura, nodo, pregunta y examen).
- **Listar objetos.** Componentes para listar objetos (asignatura, pregunta, examen y resultado).
- **Ver objetos.** Componentes para ver objetos (asignatura, nodo, pregunta y examen).
- **Borrar objetos.** Componentes para borrar objetos (asignatura, nodo, pregunta, examen y resultado).
- **Ver jerarquía.** Componente para listar la jerarquía de aprendizaje viendo los nodos y sus relaciones.
- **Crear pregunta.** Componente para crear pregunta. En este componente se añaden desde el enunciado, la dificultad y el nodo o nodos a los que se relaciona, hasta las respuestas (si se quieren añadir) y, en caso de ser una pregunta parametrizada, los parámetros.
- **Importar banco de preguntas.** Componente para añadir preguntas a partir de un archivo. Cabe destacar que independientemente del archivo utilizado para importar, las preguntas resultantes estarán asociadas al nodo raíz de la asignatura a la que se añadan, por lo que será necesario editar las preguntas a las que se les quiera asociar otros nodos.
- **Ver/Editar pregunta.** Componente para ver una pregunta, y editar la pregunta si se desea y es posible.

- **Configurar examen.** Componente para configurar un examen. Este se subdivide en secciones, que cuentan con una serie de parámetros obligatorios (número de preguntas y nodos de las preguntas) y otros opcionales (dificultad estimada, tiempo estimado por pregunta, tipo de preferencia y preferencia por preguntas parametrizadas). Estos parámetros son esgrimidos a la hora de seleccionar preguntas concretas (disponible para cada sección) o al generar las preguntas restantes para cada sección. Por otra parte, el configurador de examen también cuenta con un selector de exámenes anteriores con cuyas preguntas se quieren comparar las preguntas seleccionadas o generadas del examen a configurar.
- **Exportar examen.** Componente para exportar un examen a distintos formatos. Como se ha explicado anteriormente, se proponen distintos formatos para que el usuario elija dependiendo de sus necesidades.
- **Importar resultados.** Componente para añadir resultados asociados a un examen. Pese a que es la manera de crear un resultado, es importante destacar el componente sobre el resto, ya que se realiza únicamente a través de la importación de un archivo CSV que cuente con los datos necesarios.

### 2.6.1. Liberación del software

Por último, a continuación se van a desglosar los pasos que harían falta para llevar a cabo la liberación de este software según la Oficina de Software Libre (OSL) de la Universidad de Granada [16] :

1. Redactar un documento de aceptación de la liberación del Software que cuente con la firma de todas las partes que han contribuido, determinando que se está de acuerdo con la licencia de liberación que se pide.
2. Esperar a recibir la autorización de liberación para la publicación y liberación del código por parte de la Oficina de Transferencia de Resultados de Investigación (OTRI) de la Universidad de Granada. La OSL utiliza SafeCreative, una plataforma para registrar propiedad intelectual, para incluir la información de la liberación para su acceso público.
3. Cumplimentar un formulario de liberación disponible en la página oficial de la Oficina de Software Libre de la Universidad de Granada (<https://osl.ugr.es/liberacion-de-software/>), y remitir la documentación a la Oficina de Software Libre, tanto en formato digital (al correo electrónico [osl@ugr.es](mailto:osl@ugr.es)) como en formato físico (a la Oficina de Software Libre del Edificio Real, C/ Real de Cartuja 36-38 Granada). Estos pasarán a formar parte de un archivo propiedad de la OTRI.

Realizados los pasos necesarios, los datos de la liberación del software se publicarían tanto en la OSL como en la OTRI. Para permitir una accesibilidad más estandarizada, se publica también la liberación en SafeCreative, pudiendo acceder a la acreditación de la liberación a través del número de registro. Se ha de tener en cuenta que la licencia

concedida es la que especifique el autor de la liberación, siempre que sea compatible con el software y las librerías utilizadas. Cabe destacar que, en caso de que este proyecto sufriera cambios en el futuro, se deberían revisar los pasos a seguir para la liberación del software, ya que no se han mencionado pasos que no harían falta con la versión que se presenta (esperar a la resolución de una comisión de decisión de derechos convocada por la Universidad de Granada, y comprobar las licencias de otros recursos externos del proyecto, entre otros).

Es este caso, pediría la licencia de reconocimiento, ya que considero que el proyecto tiene mucho potencial, y contar con una licencia poco restrictiva puede atraer con más facilidad a otros desarrolladores que quieran implementar ampliaciones u otros cambios en la aplicación. Además, esto permitiría sin muchos problemas cambiar a una licencia más restrictiva en caso de que se viera conveniente.

### 2.6.2. Selección de preguntas de exámenes

En este apartado se van a explicar los detalles más relevantes de la función que se emplea para generar las preguntas para cada sección de un examen. Para ello, es preciso explicar previamente las ocasiones en las que se puede llamar a la función de selección de preguntas:

1. El usuario ha pedido seleccionar preguntas concretas en una sección durante la configuración o edición de un examen.
2. El usuario ha pedido completar una sección con las preguntas que falten durante la configuración o edición de un examen.

En realidad, en ambas ocasiones se realiza la misma operación, aunque con un pequeño añadido al final en el segundo caso.

En consecuencia, los datos de entrada son los siguientes:

- **IDs de Nodos** (conjunto de valores enteros positivos, obligatorio). ID del nodo de la jerarquía de aprendizaje al que deben de pertenecer las preguntas.
- **Número de preguntas** (valor entero positivo, opcional). Número de preguntas a añadir.
- **Dificultad estimada** (valor discreto en el intervalo  $[1,10]$ , opcional). Dificultad aproximada de cada pregunta.
- **IDs de preguntas no seleccionables** (conjunto de valores enteros positivos, opcional), IDs de las preguntas que no pueden ser seleccionadas.
- **Tiempo estimado** (valor entero positivo, opcional). Tiempo aproximado de cada pregunta.
- **Tipo de pregunta** (conjunto de valores literales, opcional). Tipos de pregunta que priorizar.

- **Prioridad de preguntas parametrizadas** (valor booleano, opcional). Prioridad mayor a preguntas que tengan parámetros de pregunta asociados sobre las que no los tengan.

La función tiene un funcionamiento básico, que se va refinando según se introducen más parámetros.

1. En primer lugar, se realiza una búsqueda de las preguntas activas que estén relacionadas con los IDs de los nodos especificados y que no tengan como ID uno de los de las preguntas no seleccionables.
2. A continuación, de la lista de preguntas recibida, se ordenan, siguiendo el siguiente listado de criterios. En caso de que dos o más preguntas se encuentren en igualdad de condiciones, se procederá al siguiente criterio de ordenación:
  - a) Si se ha especificado que se quieren preguntas parametrizadas, se devolverán primero todas las preguntas parametrizadas, aunque si no se especifica, se pasará al siguiente criterio de ordenación.
  - b) Si se han especificado tipos de pregunta, se devolverán primero las de los tipos especificados. Cabe destacar que si no se ha especificado esta opción, se atenderá a los demás criterios.
  - c) En caso de haber especificado un nivel de dificultad, por distancia al nivel de dificultad. Por ejemplo, al especificar un nivel de dificultad de 4, las preguntas de nivel de dificultad 3 se posicionarían antes que las de nivel 6.
  - d) En caso de haber especificado un tiempo estimado, por distancia al tiempo estimado. Por ejemplo, al especificar un tiempo estimado de 20, preguntas de tiempo estimado de 30 se posicionarían antes que las que tienen un tiempo estimado de 31.
  - e) En caso de no haber especificado ningún parámetro, se ordenarán por número de usos de menor a mayor. De esta manera, las preguntas que no hayan sido usadas en ningún examen se mostrarán antes que las que han sido usadas en exámenes anteriores.
  - f) En caso de que aún haya preguntas en igualdad de condiciones, se ordenarán las preguntas de forma aleatoria, para dotar de cierta variabilidad a la selección de preguntas.
3. Una vez ordenadas las preguntas, se devuelven al usuario. En caso de haber recibido un número concreto de preguntas, se devuelven únicamente las n mejores, puesto que debido a las circunstancias en las que se usa la selección de esta manera, estas preguntas están destinadas a completar automáticamente las preguntas restantes de una sección del examen.



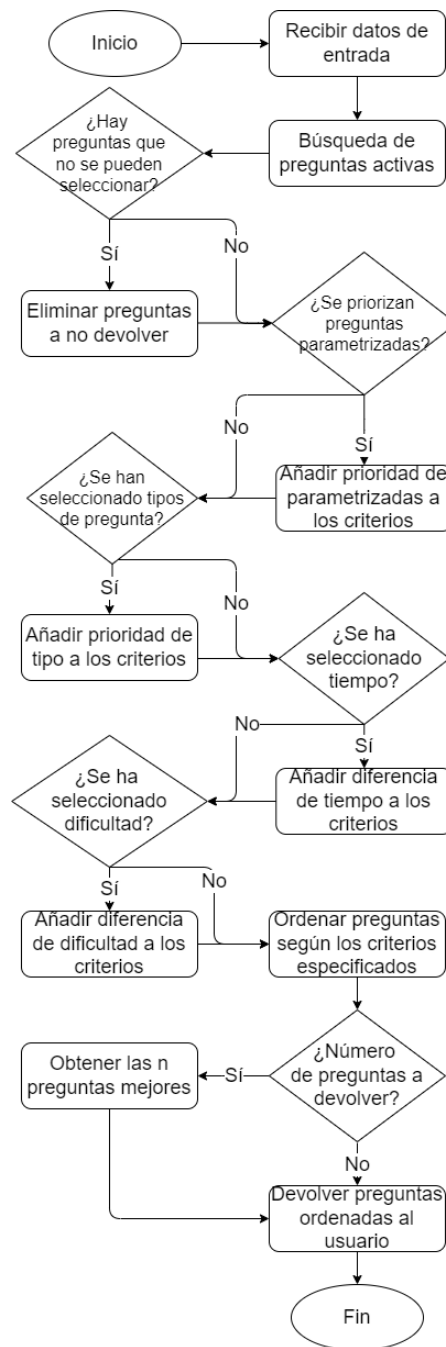


Figura 32: Diagrama de Flujo de la función de selección de preguntas.

Este proceso se iría repitiendo para cada una de las secciones que el usuario quiera añadir al examen, hasta que se hayan añadido todas las preguntas que se pidan. El usuario podrá descartar preguntas seleccionadas, cuyos IDs se tienen en cuenta junto

con los de preguntas seleccionadas para no reseleccionar preguntas en caso de realizar más de una petición de selección.

## 2.7. Estimación de Recursos

Para acercar este proyecto a la realización de un proyecto real en una empresa, este apartado se dedicará a realizar una estimación de los recursos para presentar el proyecto a una empresa.

En primer lugar, se han de estimar los recursos humanos, es decir, los miembros del equipo que llevarían a cabo el desarrollo del proyecto.

- **Full Stack Developer.** Para el desarrollo de la aplicación se contaría con un desarrollador tanto del backend como del frontend. Esto permitiría que los desarrolladores estuvieran familiarizados con el proyecto al completo y pudieran apoyarse entre ellos en caso de problemas.
- **Diseñador de UX/UI.** Para el diseño de la aplicación se contaría con un diseñador, que desarrollaría la experiencia e interfaz de usuario teniendo en cuenta una accesibilidad inclusiva y completa.
- **Testers.** Para realizar pruebas de la aplicación se contaría con un controlador de calidad, que se encargaría de realizar test para comprobar casos límite y pruebas de rendimiento y seguridad tanto de los componentes del frontend como de los componentes del backend.
- **Project Manager.** Aunque no lleve a cabo directamente ninguna tarea del desarrollo de la aplicación, el trabajo de un gestor del proyecto sería necesario para tomar decisiones con respecto a la coordinación del equipo, el seguimiento de las tareas en sus plazos correspondientes y la comunicación con otras partes interesadas, como los dueños del producto.

A continuación, es preciso estimar las necesidades de recursos materiales:

- **Herramientas de desarrollo.** A pesar de que se podrían utilizar herramientas de uso gratuito como Visual Studio Code, para ser más fieles al desarrollo real de la aplicación se optaría por licencias de JetBrains para un IDE en el que los desarrolladores, el diseñador y el controlador de calidad harían su trabajo con el código. De la misma manera, para contar con un control de versiones, cada uno de los mencionados debería contar con una cuenta en un servicio de control de versiones como GitHub.
- **Infraestructura.** Pese a que este no es un recurso que se use en la versión real de la aplicación, sería conveniente utilizar un servicio para alojar un servidor con la aplicación y la base de datos. Un ejemplo sería Amazon Web Services, que, a través de los diferentes servicios que ofrece, permitiría alojar la aplicación, incluso diferentes servidores para permitir, por ejemplo, un servidor de producción para

que la usara el público y otro de desarrollo para desarrollar ampliaciones u otros cambios.

- **Dispositivos de trabajo.** Para el desarrollo por parte de los trabajadores sería necesario un equipo adecuado para el desarrollo de la plataforma. Aproximadamente, necesitaría un equipo para cada trabajador de, al menos, 8 GB de RAM y un procesador i7, equivalente o mejor.

Por otra parte, el gestor de proyecto debería de realizar un cronograma de las distintas fases de desarrollo de la plataforma, temporalizando cada fase del proyecto e incluyendo en cada una de ellas las tareas correspondientes.

Finalmente, se deberían de estimar los costes de todo lo mencionado:

- Para cada miembro del equipo sería preciso estimar las horas que emplearía, y el coste salarial derivado del trabajo hecho. Teniendo en cuenta el tiempo estimado de realización de este proyecto, se necesitarían unas 300 horas para el desarrollo.
- En cuanto al equipo, la empresa debería estimar los gastos derivados de los dispositivos de trabajo.
- Por último, se deberían de tener en cuenta los gastos de tanto las licencias de desarrollo (JetBrains y GitHub) como del mantenimiento de la infraestructura mencionada en Amazon Web Services.

Esto es sólo una estimación que se podría hacer, pero dependiendo de las características de la empresa se podrían añadir más características que podrían beneficiar al desarrollo o incluso modificar los ejemplos mencionados para un desarrollo más ajustado al nivel de la empresa.

### 3. Manual de usuario

Una vez planteada la solución, es preciso explicar la instalación y despliegue de la aplicación web, así como el curso del uso básico de la misma (desde registrar un usuario hasta generar un examen y asociarle resultados). Este manual busca proporcionar una guía clara y concisa para que los usuarios puedan configurar e interactuar con la aplicación sin dificultades. Las funcionalidades más allá de lo que se va a explicar son fácilmente accesibles teniendo en cuenta la manera de acceder a las funcionalidades a exponer. No obstante, si existen dudas, la aplicación cuenta con elementos de ayuda que se muestran en algunos elementos al pasar el ratón para explicar el propósito o funcionamiento de estos elementos.

#### 3.1. Manual de Instalación y Despliegue en Linux

Previo a la instalación, es necesario contar con los siguientes recursos ya configurados:

- Aplicación de Docker. [14]
- Node.js.[18]
- Cuenta de GitHub.
- Navegador web.
- Terminal de Linux con git instalado.
- Python.

##### 3.1.1. Manual de Instalación en Linux

Por un lado, para instalar el backend:

1. Clonar el repositorio:

```
git clone https://github.com/rafadgvc/TFG-backend.git
```

2. En el directorio del proyecto, generar un entorno virtual (para evitar la instalación de dependencias en el equipo):

```
python -m venv venv
```

3. Activar el entorno virtual:

```
source venv/bin/activate
```

#### 4. Instalar los requerimientos:

```
pip install -r requirements.txt
```

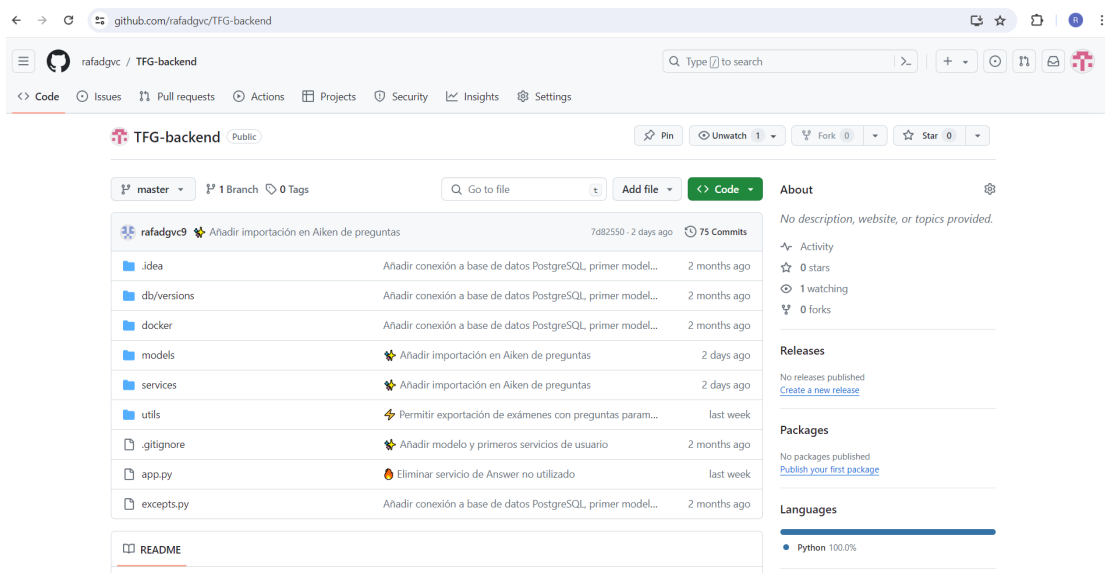


Figura 33: Repositorio de GitHub en el que se aloja el proyecto.

Por otro lado, para instalar el frontend:

#### 1. Clonar el repositorio:

```
git clone https://github.com/rafadgvc/TFG-frontend.git
```

#### 2. En el directorio del proyecto, instalar las dependencias:

```
npm install
```

### 3.1.2. Manual de Despliegue en Linux

Para realizar el despliegue de la aplicación:

1. Crear en el directorio del backend un archivo `'secret.py'` en el que almacenar las constantes `'PASSWORD_SALT'` (cadena de caracteres para realizar un hasheo de la contraseña del usuario) y `'JWT_SECRET_KEY'` (cadena de caracteres para generar el token de acceso con el que se comprueba el usuario que ha iniciado sesión). Un ejemplo del contenido se observa en la figura ??.

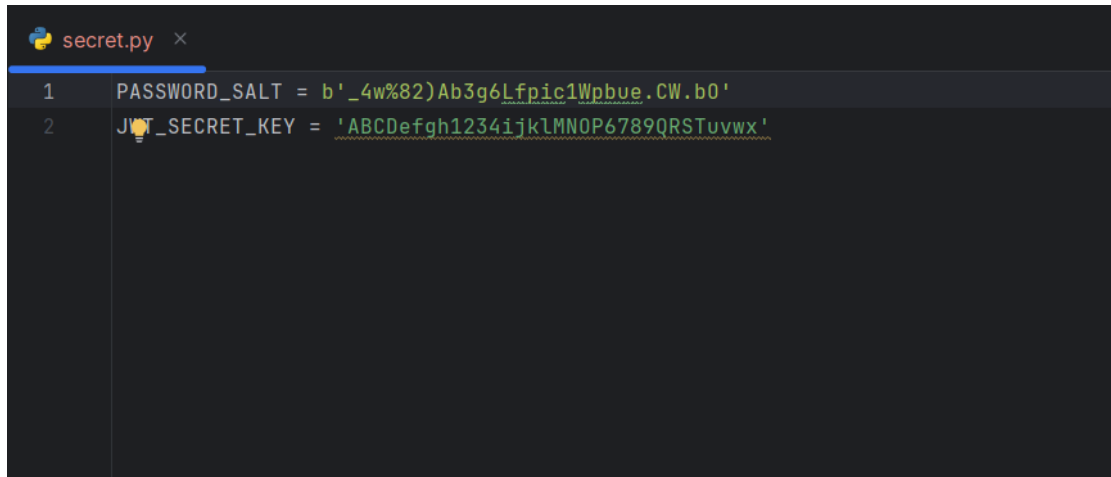


Figura 34: Ejemplo del contenido del archivo '*secret.py*'.

2. Iniciar el contenedor de Docker para iniciar la base de datos, ejecutando la composición del contenedor en el directorio del archivo '*docker-compose.yml*':

```
docker-compose up
```

3. Iniciar el backend desde su directorio:

```
flask run
```

4. Iniciar el frontend desde su directorio:

```
npm start
```

5. Abrir un navegador web y acceder a la dirección '*http://localhost:4200/*' (Ver figura 35).

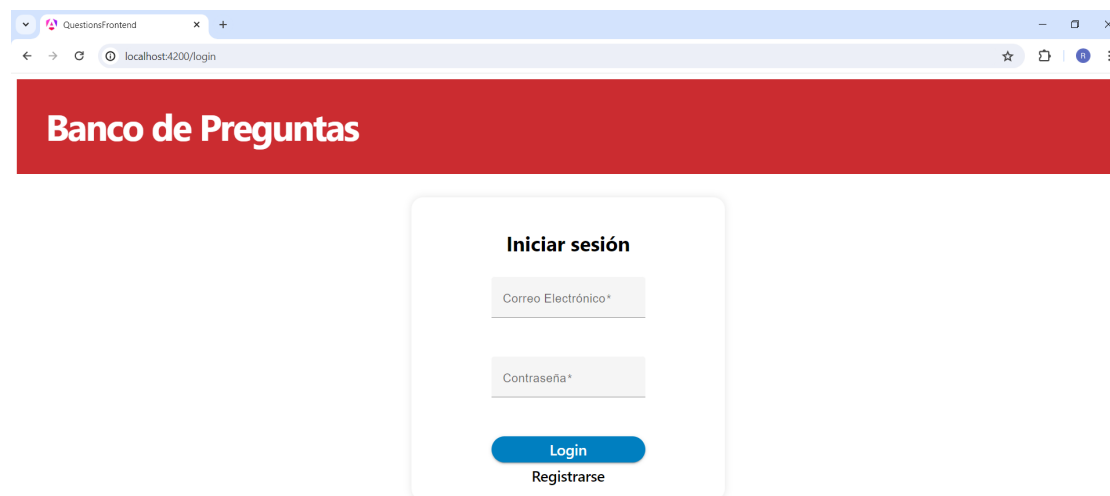


Figura 35: Captura de la pantalla de inicio de la aplicación.

### 3.2. Manual de Instalación y Despliegue en Windows

Previo a la instalación, es necesario contar con los siguientes recursos ya configurados:

- Aplicación de Docker. [14]
- Node.js.[18]
- Cuenta de GitHub.
- Navegador web.
- Símbolo del sistema de Windows (cmd) con git instalado.
- Python.

Además, es necesario contar con git y Python en el PATH del sistema.

#### 3.2.1. Manual de Instalación en Windows

Por un lado, para instalar el backend:

1. Clonar el repositorio:

```
git clone https://github.com/rafadgvc/TFG-backend.git
```

2. En el directorio del proyecto, generar un entorno virtual (para evitar la instalación de dependencias en el equipo):

```
python -m venv venv
```

3. Activar el entorno virtual:

```
venv\Scripts\activate
```

4. Instalar los requerimientos:

```
pip install -r requirements.txt
```

Por otro lado, para instalar el frontend:

1. Clonar el repositorio:

```
git clone https://github.com/rafadgvc/TFG-frontend.git
```

2. En el directorio del proyecto, instalar las dependencias:

```
npm install
```

### 3.2.2. Manual de Despliegue en Windows

Para realizar el despliegue de la aplicación:

1. Crear en el directorio del backend un archivo *secret.py* en el que almacenar las constantes *PASSWORD\_SALT* (cadena de caracteres para realizar un hash de la contraseña del usuario) y *JWT\_SECRET\_KEY* (cadena de caracteres para generar el token de acceso con el que se comprueba el usuario que ha iniciado sesión). Un ejemplo del contenido se observa en la figura ??.
2. Iniciar el contenedor de Docker para iniciar la base de datos, ejecutando la composición del contenedor en el directorio del archivo *docker-compose.yml*:

```
docker-compose up
```

3. Iniciar el backend desde su directorio:



```
flask run
```

4. Iniciar el frontend desde su directorio:

```
npm start
```

5. Abrir un navegador web y acceder a la dirección <http://localhost:4200/>. (Ver figura 35)

### 3.3. Manual de Uso

En primer lugar, es necesario registrar un usuario. Para ello, partiendo de la página de inicio mostrada en el apartado anterior, se accede a 'Registrarse' (o, en su defecto, se navega a '/signup') (Ver figura 36).

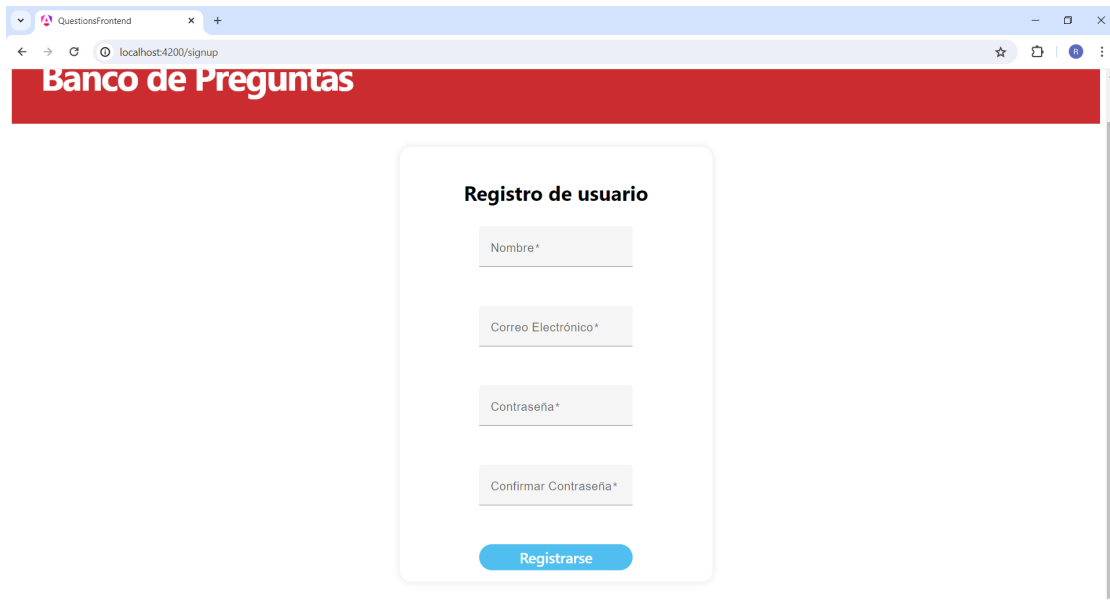
La imagen muestra una captura de pantalla de un navegador web. La barra de direcciones indica la URL 'localhost:4200/signup'. El encabezado de la página es un banner rojo con el texto 'Banco de Preguntas' en blanco. En el centro de la pantalla hay un formulario con el título 'Registro de usuario'. El formulario contiene cuatro campos de entrada de texto: 'Nombre\*', 'Correo Electrónico\*', 'Contraseña\*' y 'Confirmar Contraseña\*'. Debajo de estos campos hay un botón azul con el texto 'Registrarse'.

Figura 36: Captura de la pantalla de registro de usuario.

En esta pantalla, se registran los datos del usuario (nombre, email y contraseña). Al realizar el registro con éxito, ya se puede iniciar sesión. Para ello, se vuelve a '/login' y se inicia sesión con los credenciales (Ver figura 37).

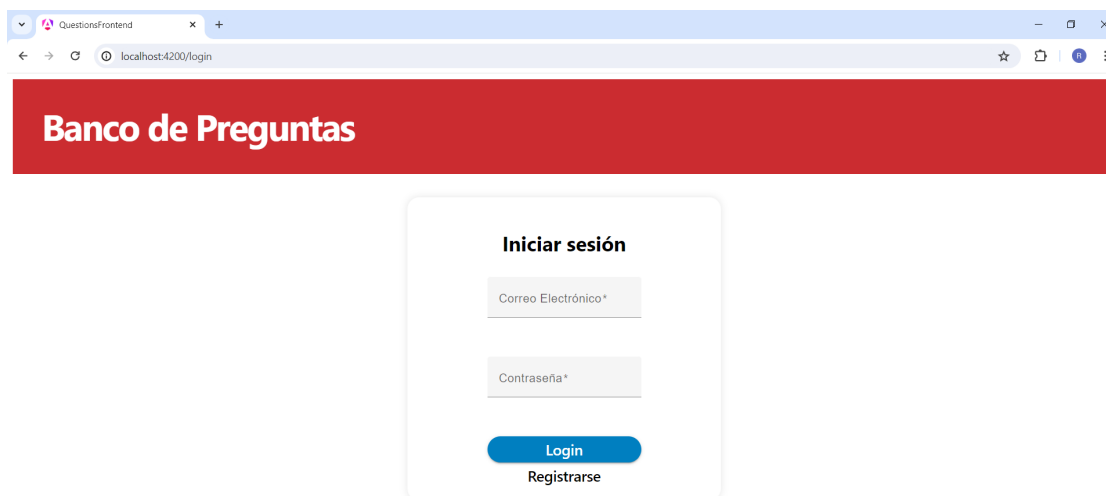


Figura 37: Captura de la pantalla de inicio de sesión.

Al iniciar sesión correctamente, se redirige a la página principal, '/home' (Ver figura 38), en la que se puede acceder a las asignaturas del usuario que ha iniciado sesión, además de poder crear nuevas asignaturas con el botón 'Crear' (Ver figura 39).

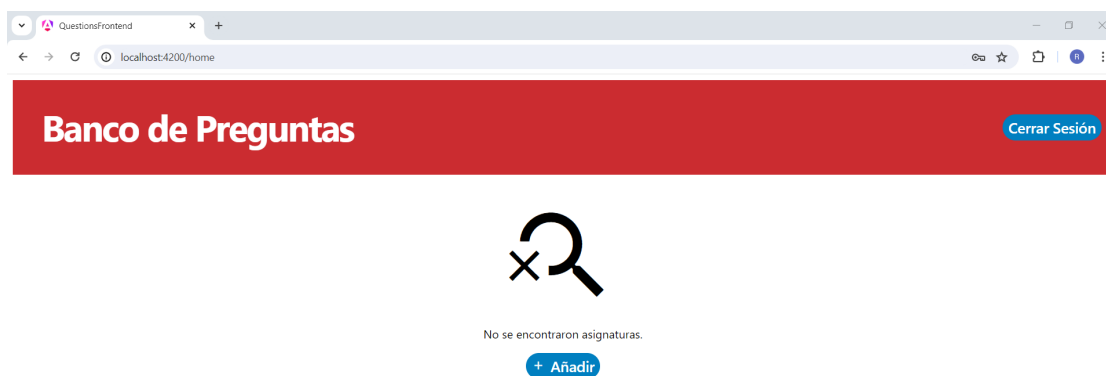


Figura 38: Captura de la pantalla principal.

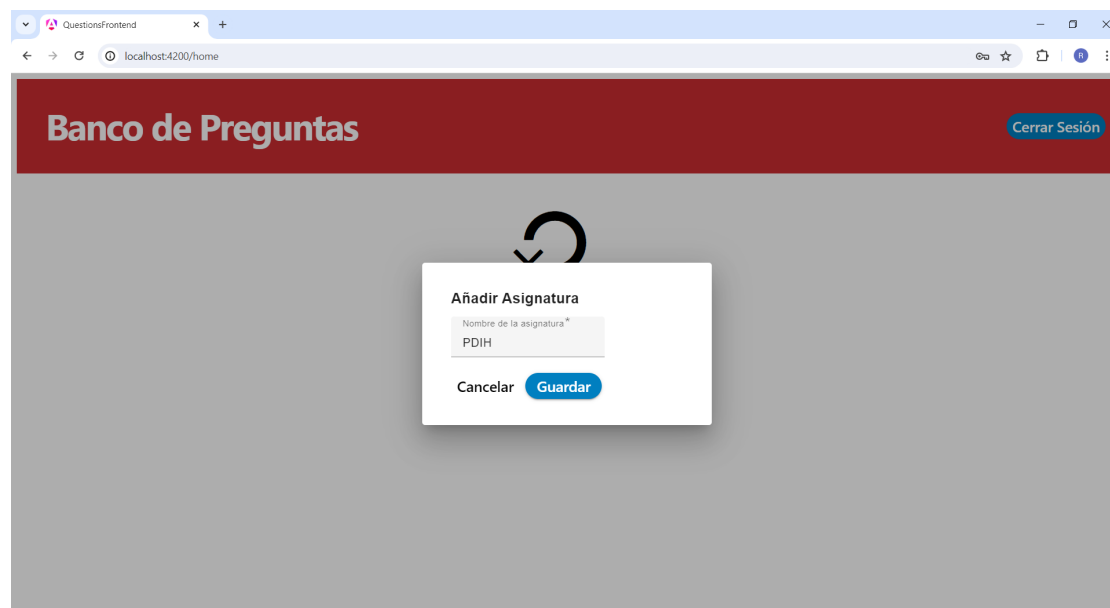


Figura 39: Captura de la pantalla de añadir asignatura.

Si fuera necesario, hay un botón de 'Cerrar sesión' en la parte superior de la página al haber iniciado sesión. Al pulsarlo, se cierra la sesión y se redirige a la pantalla de la imagen 37. En esta misma vista se pueden acceder a los detalles de la asignatura, desde los cuales se puede acceder a todos los recursos asociados a la misma (jerarquía de aprendizaje, banco de preguntas, exámenes y resultados) (Ver figura 40).



Figura 40: Captura de la pantalla de detalles de asignatura.

Por otra parte, en esta vista se pueden observar varios botones: Al pulsar el botón de

'Renombrar' se puede renombrar la asignatura (Ver figura 41) y al pulsar el botón de 'Borrar', se puede borrar la asignatura (Ver figura 42). Al pulsar el botón de 'Jerarquía', se accede al listado de nodos de la jerarquía de la asignatura, indentando e intensificando el color de cada nodo a medida que se accede a un nivel más profundo. En esta pantalla se pueden añadir nuevos nodos, especificando el nombre del nodo y el nombre del padre (Ver figura 43).

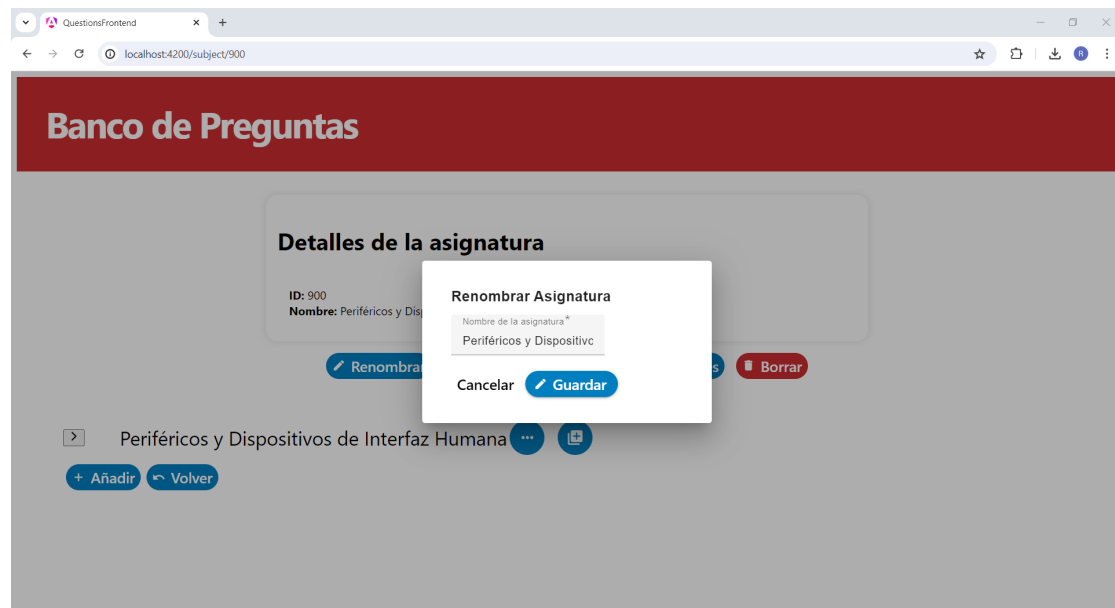


Figura 41: Ventana de editar asignatura.

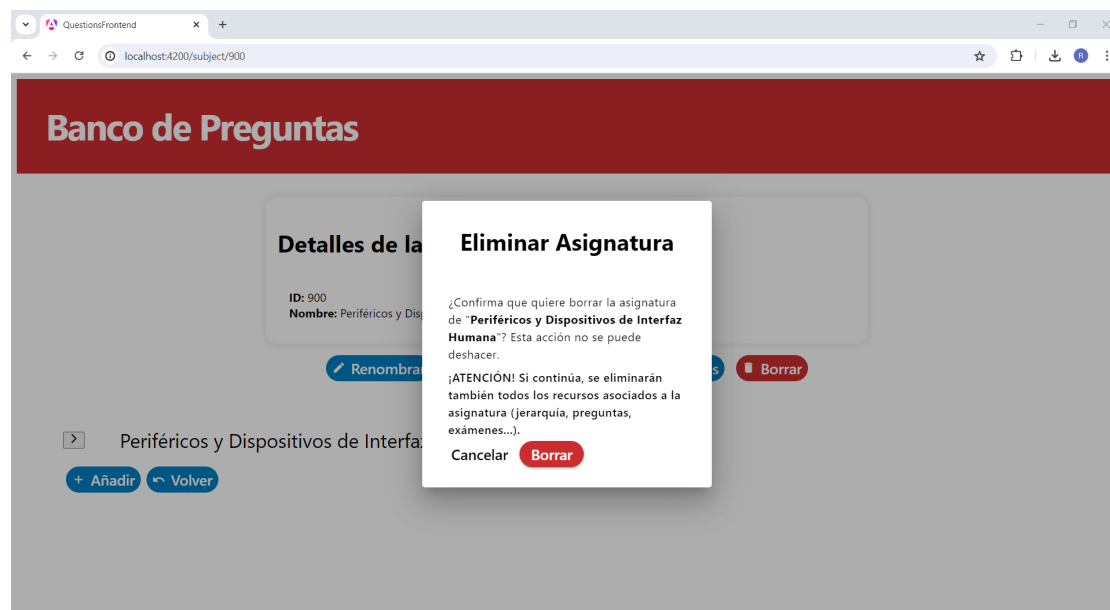


Figura 42: Captura de la ventana de eliminar asignatura.

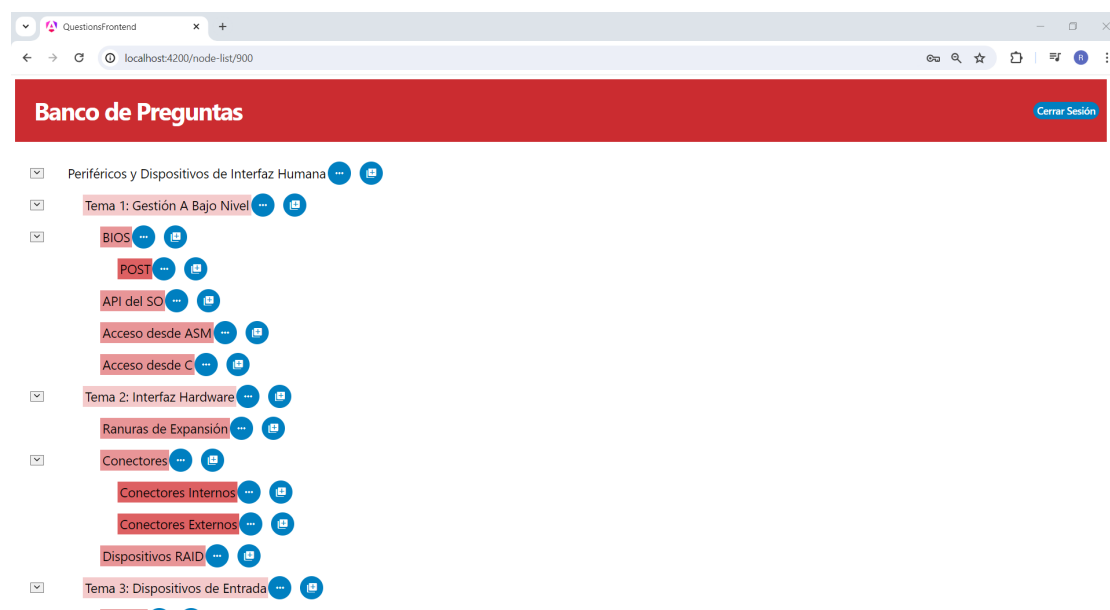


Figura 43: Captura de la pantalla de listado de nodos.

En esta vista se pueden observar los botones de 'Añadir' y los botones con el icono '+' a la derecha de cada nodo. Al pulsar cualquiera de estos, se abre una ventana para añadir un nodo, con la diferencia de que al pulsar el botón de un nodo, se preseleccionará ese nodo como padre del nuevo nodo (ver figura 44).

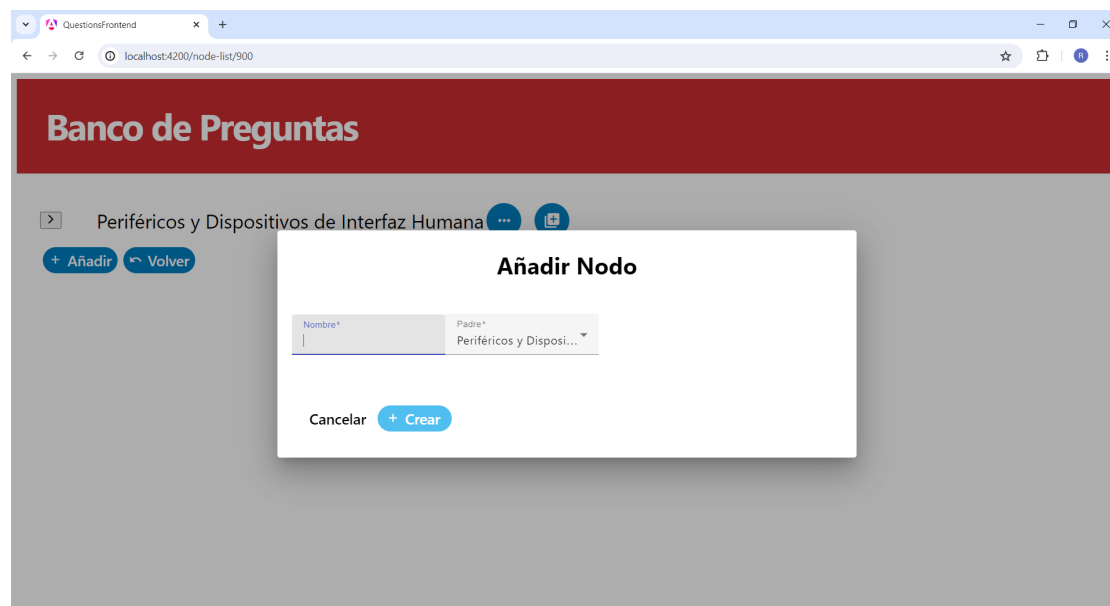


Figura 44: Ventana de añadir nodo.

Al pulsar en el icono de '...' a la derecha de cada nodo se puede acceder a una vista de los detalles del nodo (Ver figura 45). En esta pantalla se observan dos botones: 'Editar', para renombrar el nodo (ver figura 46), y 'Borrar', que estará desactivado si el nodo no se puede eliminar (ver figura 47).



Figura 45: Vista de detalles de nodo.

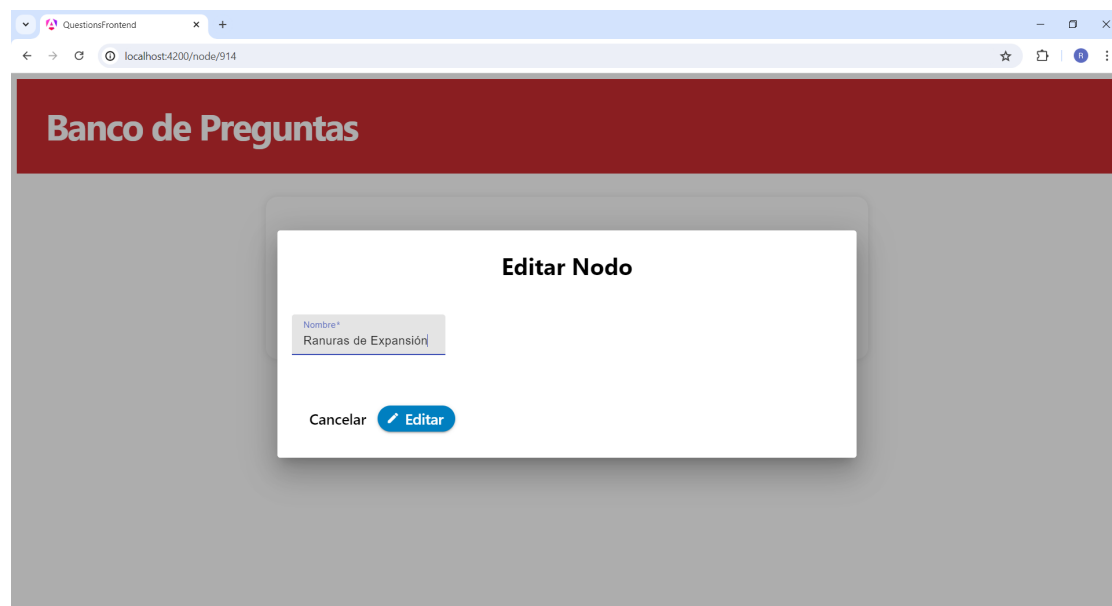


Figura 46: Ventana de editar nodo.

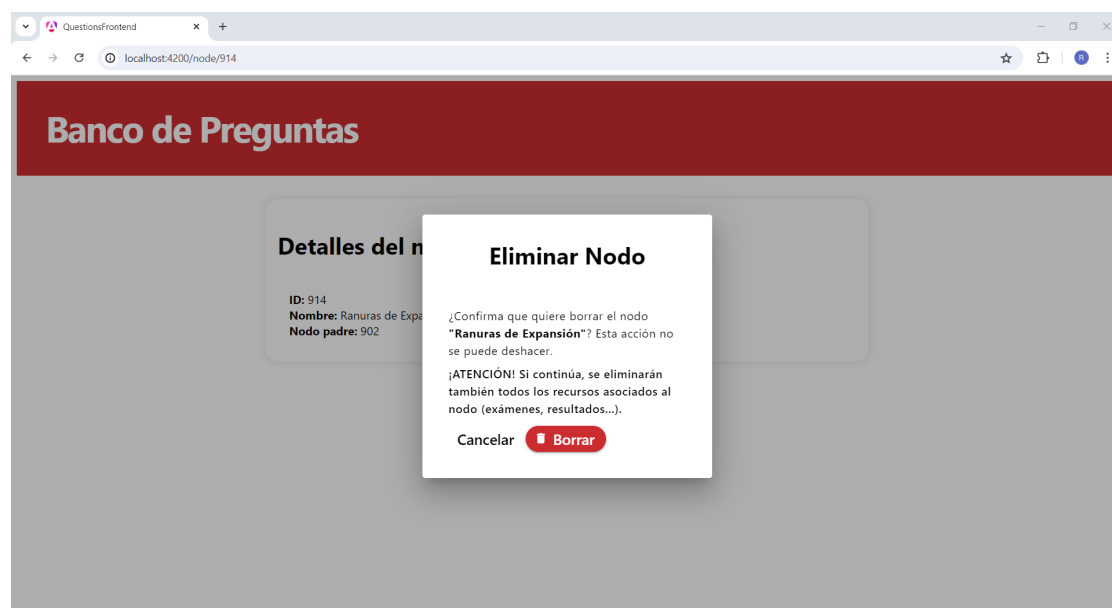


Figura 47: Ventana de eliminar nodo.

Si se vuelve a la pantalla de detalles de la asignatura, se puede observar el botón de 'Banco'. Al pulsarlo, se accede al listado de preguntas creadas para la asignatura (Ver figura 48).

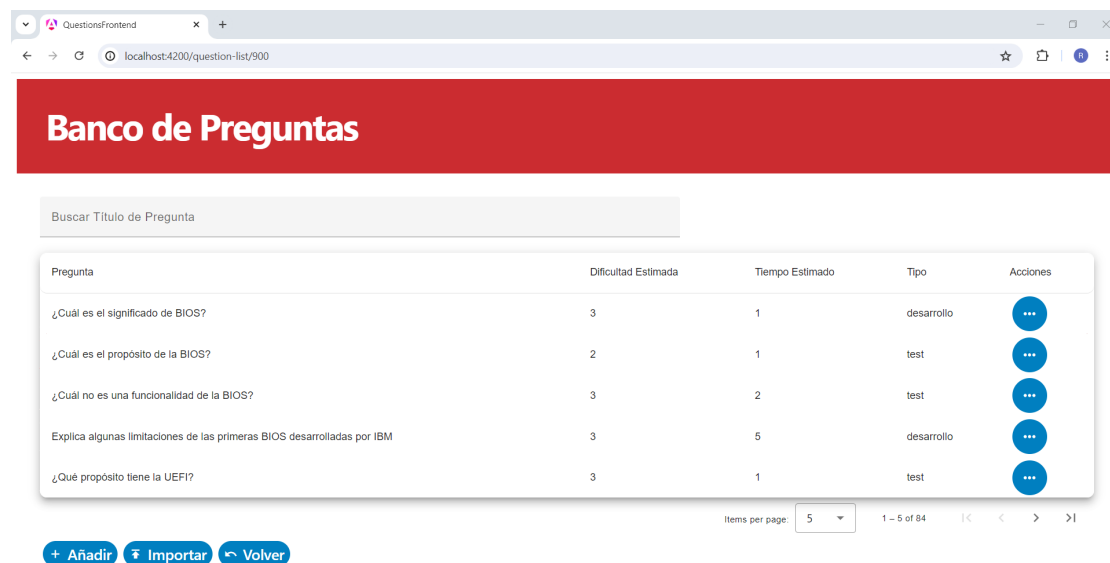


Figura 48: Captura de la pantalla de listado de preguntas.

En esta misma pantalla se puede acceder a la vista de creación de preguntas desde el botón 'Añadir'. En ella, se pueden especificar todos los datos relativos a una pregunta. El apartado de respuestas no se muestra hasta que no se ha seleccionado el tipo de pregunta, para evitar confusiones (por ejemplo, para evitar que el usuario piense que debe incluir puntuaciones en preguntas de desarrollo). De la misma manera, el apartado de los parámetros de pregunta se oculta hasta que se marca que la pregunta cuenta con parámetros (Ver figura 49).



**Añadir pregunta**

**Datos de la Pregunta**

Tipo de la pregunta\*    Nodos de la Jerarquía\*    Dificultad\*    Tiempo Estimado\*

**Enunciado**

Enunciado\*

☐ Parametrizada

+ Crear Pregunta

Figura 49: Captura de la vista de creación de preguntas.

Al crear la pregunta, se podrá ver en el listado de preguntas y se podrá utilizar al configurar un examen. También se pueden ver los detalles de la pregunta en la misma vista que la vista de edición (Ver figura 50). En esta última vista también se pueden desactivar preguntas si hace falta (Ver figura 51). Para añadir una pregunta a un examen, desde la vista de detalles de asignatura, al pulsar en el botón 'Exámenes', se puede ver un listado de los exámenes creados para la asignatura (Ver figura 52).

The screenshot shows a web browser window with the URL `localhost:4200/question/9001`. The page has a red header with the text **Banco de Preguntas**. Below the header, the title **Ver/Editar pregunta** is centered. The main content area is divided into two sections: **Datos de la Pregunta** and **Enunciado**. The **Datos de la Pregunta** section contains four input fields: **Tipo de la pregunta\*** (set to 'Desarrollo'), **Nodos de la Jerarquía\*** (set to 'Tema 1: Gestión A B...'), **Dificultad\*** (set to '3'), and **Tiempo Estimado\*** (set to '1'). The **Enunciado** section contains a text area with the text '¿Cuál es el significado de BIOS?'. The background of the form is a light pink color.

Figura 50: Captura de la pantalla de detalles/edición de una pregunta.

The screenshot shows the same web browser window, but now displaying the **Respuestas** section. A modal dialog titled **Desactivar Pregunta** is open in the center. The dialog contains the following text: '¿Confirma que quiere desactivar la pregunta **¿Cuál es el significado de BIOS?**? Esta acción no se puede deshacer. ¡ATENCIÓN! Si continúa, sus resultados actuales no cambiarán, pero esta pregunta no podrá usarse en futuros exámenes.' At the bottom of the dialog are two buttons: **Cancelar** and **Desactivar**. In the background, the **Respuestas** section is visible, showing a list of answers. The first answer is 'Resposta 1\*' with the text 'Basic Input Output System'. There are buttons for **+ Añadir**, **Guardar**, **Desactivar**, and **Borrar**. A toggle switch for 'Parametrizada' is also visible.

Figura 51: Captura de la ventana de desactivación de una pregunta.

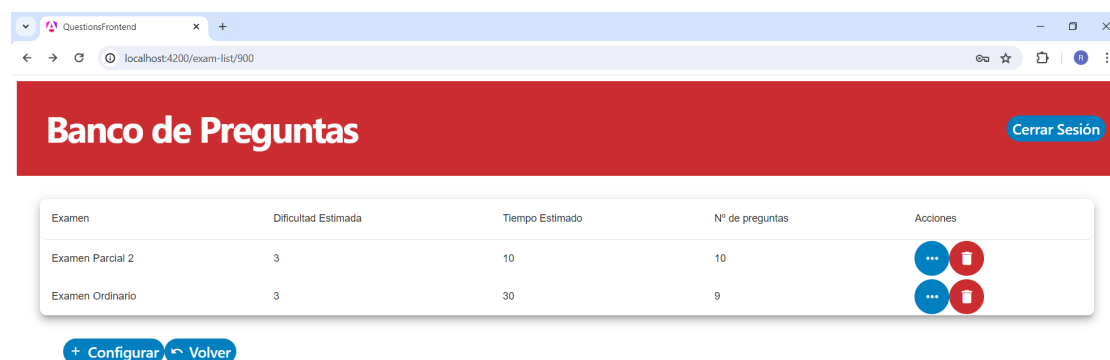


Figura 52: Captura de la vista del listado de exámenes.

En esta pantalla, al pulsar en el botón 'Configurar', se accede a la vista de configuración de examen (Ver figura 53). En ella, se pueden especificar todos los datos relativos a un examen, así como los datos de cada sección. Para cada sección, se puede observar un botón de 'Seleccionar', que mostrará las preguntas según los parámetros de la sección. Estas podrán ser seleccionadas para formar parte de la sección. Por otra parte, al final de la vista, se pueden apreciar los botones 'Generar preguntas', que genera para todas las secciones las preguntas que falten según los parámetros especificados, y 'Comparar'. Si se observa el inicio de la vista, se puede ver un campo 'Exámenes anteriores': al pulsarlo, se despliega la lista de exámenes creados de la asignatura, y el botón de 'Comparar' obtiene la lista de preguntas de los exámenes seleccionados. Para las preguntas que se seleccionen o se generen, se incluirá un icono de aviso (Ver figura 54).

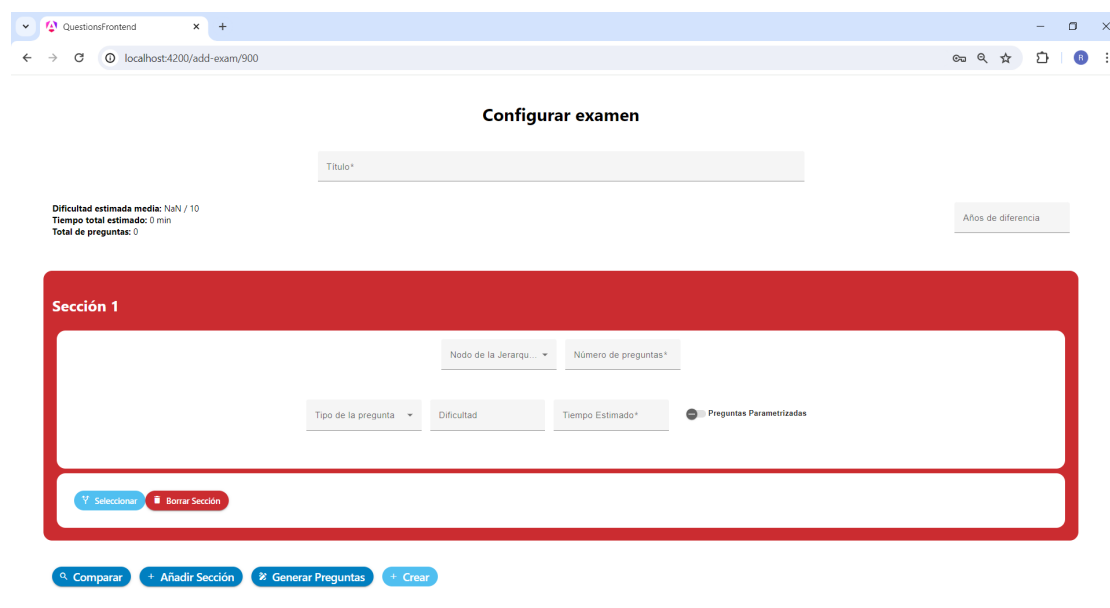


Figura 53: Captura de la vista de configuración de examen.

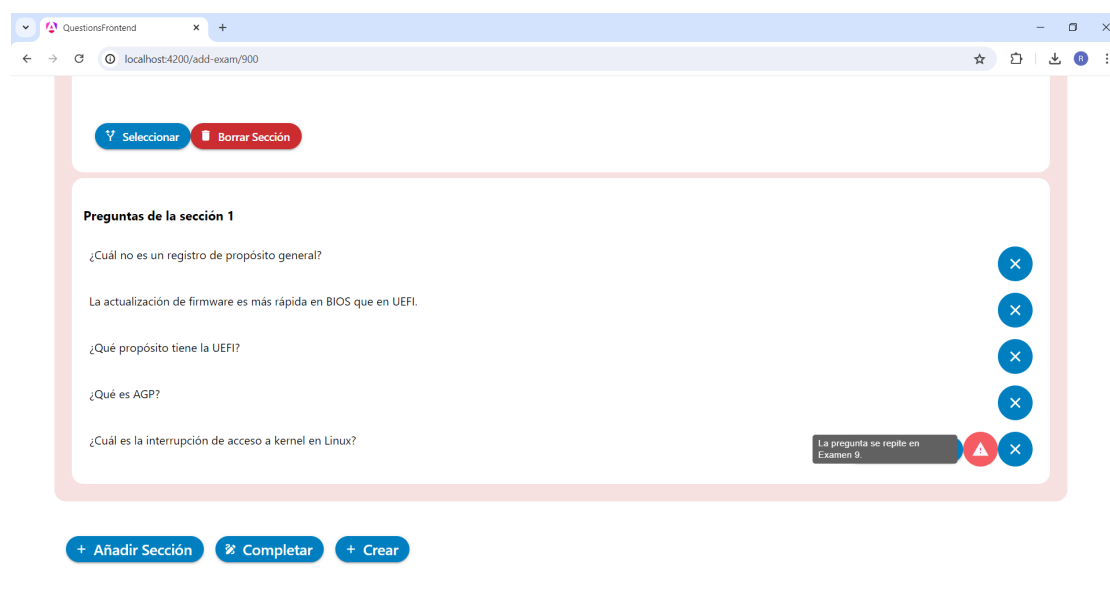


Figura 54: Captura de la vista de configuración de examen con un aviso de pregunta repetida.

Al crear el examen, se podrá ver en el listado de exámenes. En el listado de exámenes se puede borrar el examen al pulsar en el botón de la papelera (Ver figura 55). Al acceder a los detalles de un examen pulsando el botón '...' a la derecha del examen en el listado

de exámenes (Ver figura 56), se puede exportar pulsando el botón 'Exportar' (Ver figura 57) y asociar resultados pulsando el botón 'Importar Resultados' (Ver figura 58). En este último caso, se pedirá un archivo CSV con los datos de los resultados. Al importarlo correctamente, se podrán ver los resultados en los detalles del examen (Ver figura 59).

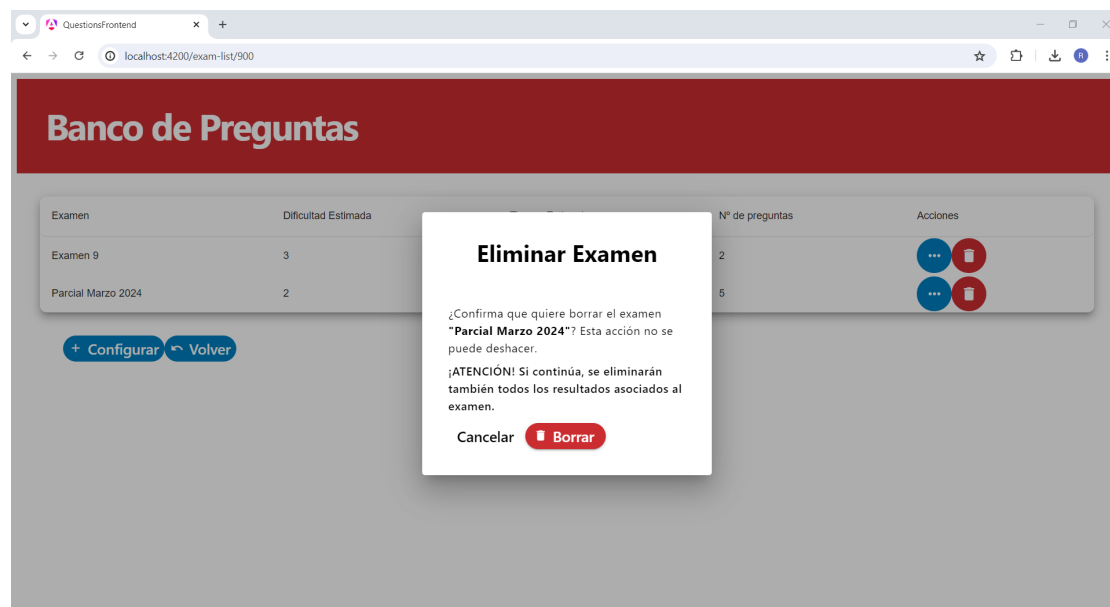


Figura 55: Captura de la ventana de eliminar examen.



Figura 56: Captura de la pantalla de detalles de un examen.

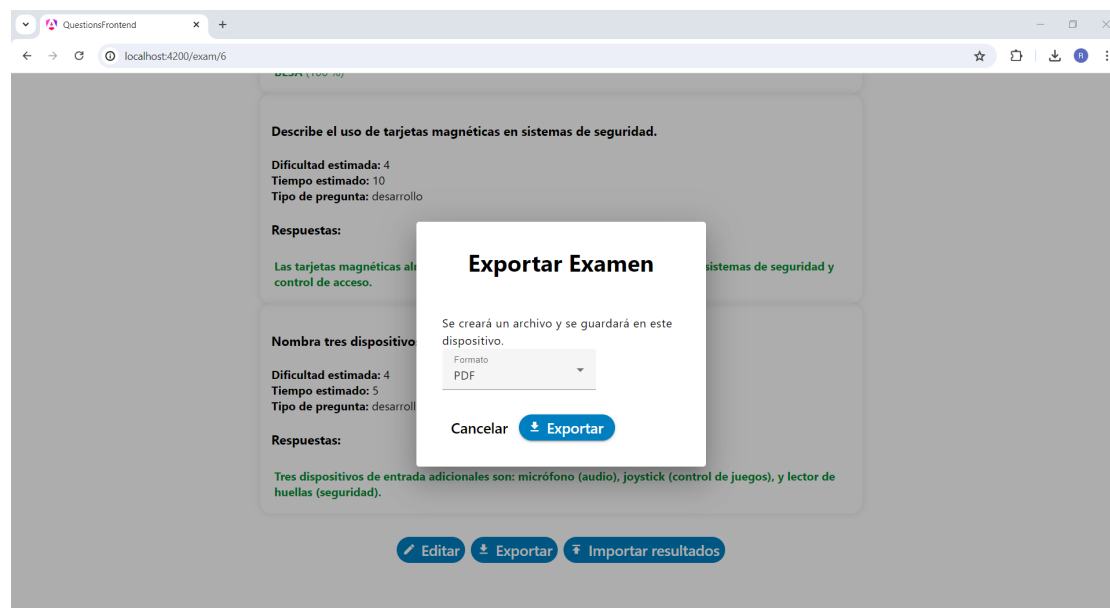
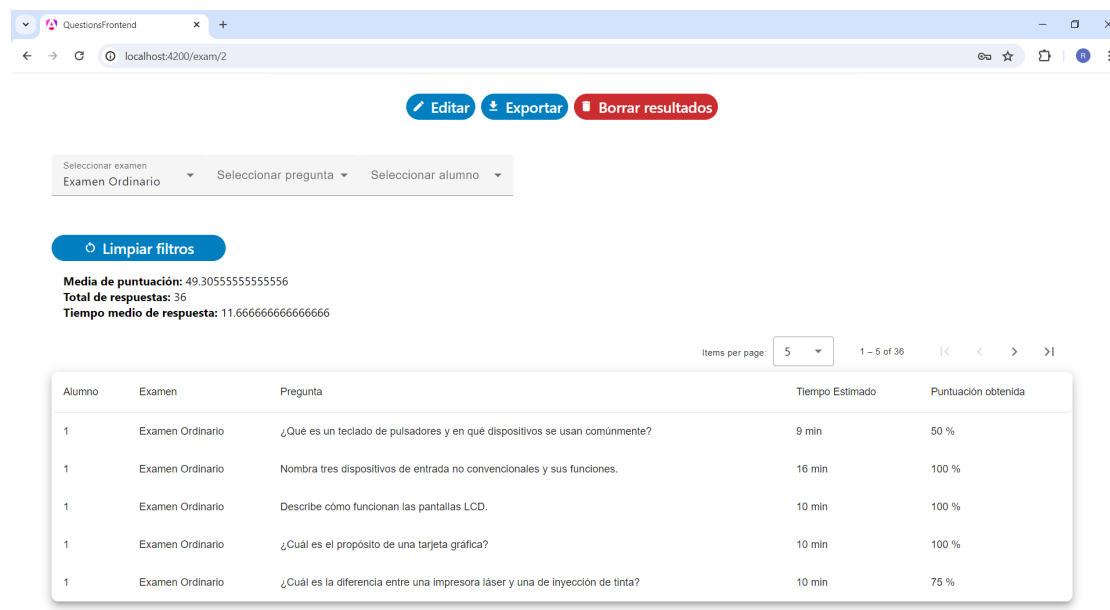


Figura 57: Captura de la ventana de exportación de examen.



Figura 58: Captura de la ventana de importación de resultados.



Alumno	Examen	Pregunta	Tiempo Estimado	Puntuación obtenida
1	Examen Ordinario	¿Qué es un teclado de pulsadores y en qué dispositivos se usan comúnmente?	9 min	50 %
1	Examen Ordinario	Nombra tres dispositivos de entrada no convencionales y sus funciones.	16 min	100 %
1	Examen Ordinario	Describe cómo funcionan las pantallas LCD.	10 min	100 %
1	Examen Ordinario	¿Cuál es el propósito de una tarjeta gráfica?	10 min	100 %
1	Examen Ordinario	¿Cuál es la diferencia entre una impresora láser y una de inyección de tinta?	10 min	75 %

Figura 59: Captura de la vista de listado de resultados.

Finalmente, para eliminar los resultados del examen, se accede al examen y se pulsa el botón 'Eliminar resultados' (Ver figura 60)

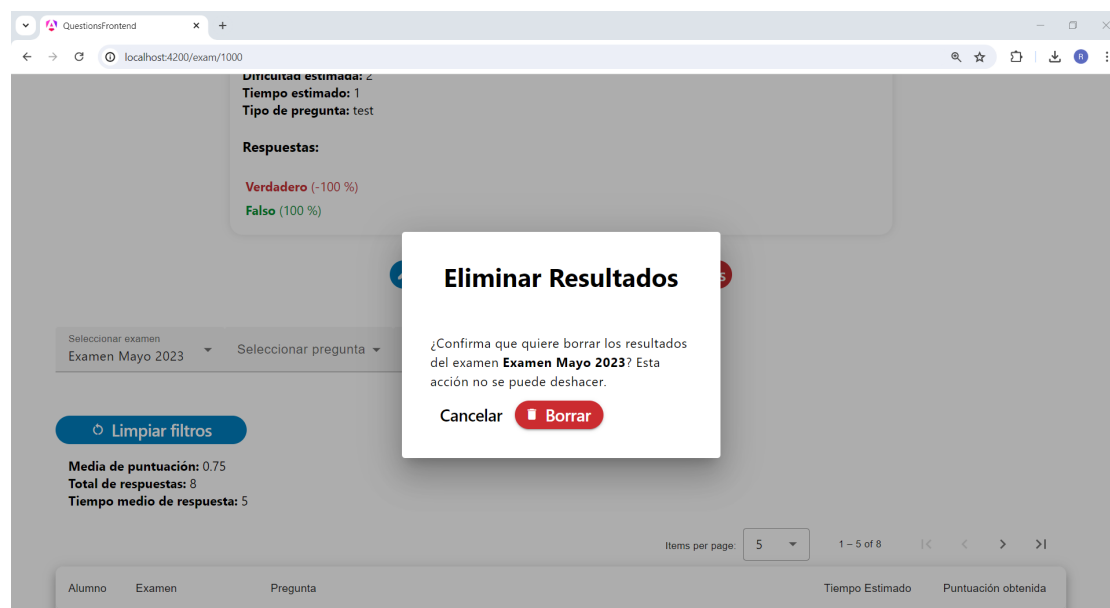


Figura 60: Captura de la ventana de eliminar resultados.

## 4. Conclusiones

El desarrollo del software ha resultado en una herramienta eficaz en el ámbito educativo. A lo largo de este proyecto se han alcanzado los objetivos planteados, y se han superado varios desafíos técnicos y metodológicos. Los principales logros que se han conseguido son los siguientes:

- **Gestión de Asignaturas de forma jerárquica:** Se ha implementado un sistema que permite la creación y organización de asignaturas en una estructura jerárquica. Esto facilita la gestión del contenido o competencias de la misma y la importancia de distintos ítems en la asignatura, además de la posibilidad de relacionar distintos ítems a través de antecesores comunes.
- **Gestión de Bancos de Preguntas:** Se ha implementado un sistema que permite la creación de preguntas de manera intuitiva. Esto facilita la asociación de preguntas a ítems específicos de la asignatura y niveles de dificultad, proporcionando una base sólida para la generación de exámenes personalizados. Asimismo, la asignación de preguntas a los distintos ítems de una asignatura es una característica novedosa y que, debido a la posibilidad de relaciones múltiples entre preguntas e ítems, cuenta con un método más para relacionar distintos ítems entre sí. Por otra parte, la inclusión de preguntas parametrizadas supone más posibilidades a la hora de gestionar preguntas de estructura similar.
- **Gestión de Exámenes:** El algoritmo desarrollado para la generación de exámenes permite especificar múltiples parámetros, garantizando así que las pruebas sean equilibradas y alineadas con los objetivos de aprendizaje y competencias deseadas. Al incluir criterios adicionales, como el historial de exámenes y la dificultad de las preguntas, se cuenta incluso con mayor precisión y control en la creación de exámenes. Asimismo, se cuenta con un sistema de exportación de exámenes en una variedad de formatos, que permiten un uso con otras plataformas más extendidas como Moodle<sup>™</sup> (a través de los formatos Aiken, GIFT y MoodleXML), a la par que un uso con formatos más estandarizados (en el caso de formatos como PDF, ODT y CSV), permitiendo incluso la exportación de exámenes en formatos editables para un examen más personalizable.
- **Evaluación y Análisis de Resultados:** El sistema no solo genera exámenes, sino que también permite la importación y análisis de resultados. Esta funcionalidad es crucial para identificar áreas de mejora tanto para los estudiantes como para los docentes, permitiendo detectar estas posibles mejoras más rápidamente. No obstante, a pesar del ligero componente de aleatoriedad en la generación de preguntas de los exámenes, este está controlado por el docente a través de los parámetros que se pueden incluir, así como la inclusión directa de las preguntas, por lo que no se ponen en compromiso la igualdad entre exámenes ni la evaluación de competencias.
- **Integración de Tecnologías:** El sistema integra tecnologías actuales, pero con fácil comprensión, como Flask, Angular y PostgreSQL. Esto permite una introducción



más sencilla al proyecto, en el caso de querer realizar modificaciones o ampliaciones en un futuro. En mi caso, como se ha mencionado en otros apartados, tenía cierto conocimiento de estas tecnologías debido a proyectos anteriores, aunque ha sido interesante y enriquecedor aprender el funcionamiento de tecnologías como Moodle™ así como los archivos que maneja esta plataforma, como Aiken, GIFT o MoodleXML. [13] [1] [17]

Es preciso destacar el aprendizaje en algunas asignaturas del Grado que han sido de gran utilidad para el desarrollo de este proyecto, tanto en lo que respecta al proyecto en sí como en lo que respecta a todo lo que rodea al mismo:

- **Fundamentos de Bases de Datos.** En esta asignatura se enseñan con detalle los conceptos relacionados con las bases de datos y el álgebra relacional. Estos conocimientos han sido indispensables para la configuración de la base de datos, entidades, relaciones y restricciones correctamente, así como para realizar operaciones de la base de datos siguiendo las bases del álgebra relacional.
- **Programación Web.** La programación web ha sido crucial para la concepción de gran parte del frontend, aunque cabe destacar que en esta asignatura ha sido de gran utilidad conocer los distintos patrones de diseño web para determinar cuál era el más cercano al proyecto.
- **Fundamentos de Ingeniería de Software.** En esta asignatura se enseña gran parte de la documentación que ha de incluir un proyecto: Diagramas de Interacción, Casos de Uso, Diagramas de Clases...
- **Inteligencia de Negocio.** En esta asignatura se enseña el funcionamiento de distintas técnicas de Inteligencia Artificial, así como las nociones básicas para la programación en Python, la cual ha sido muy importante para todo el trabajo del backend y el manejo de archivos.
- **Ingeniería de Sistemas de la Información.** En esta asignatura se muestra, especialmente en prácticas, el proceso de desarrollo de un proyecto, especialmente de cara a la interacción con el cliente: confección de presupuestos, diagramas que explican el funcionamiento de la plataforma desarrollada...
- **Diseño y Desarrollo de Sistemas de Información.** En esta asignatura se aprende a diseñar un Sistema de Información, lo que ha sido indispensable para el desarrollo de toda la aplicación web resultante.
- **Prácticas de Empresa.** Aunque se trata de una asignatura un tanto particular, a lo largo de las Prácticas de Empresa se aprenden mucho en situaciones reales con proyectos creados desde cero, como el trabajo en un equipo, el contacto con un cliente o la importancia de técnicas de metodología Ágil y de documentación extensa y concisa para un proyecto. Asimismo, en mi caso este periodo me introdujo a gran parte de las tecnologías utilizadas en el proyecto.

Personalmente, este proyecto me ha sido tremendamente útil para cumplimentar y demostrar la formación obtenida en el Grado, además de prepararme a un caso muy cercano a uno real. Agradezco la oportunidad de este proyecto con perspectivas al futuro, ya sea junto a este proyecto u otros nuevos horizontes.

#### 4.1. Posibilidades futuras

Al observar el proyecto, es aparente que la plataforma tiene algunas posibilidades de ampliación. A continuación se van a desglosar brevemente algunas de las posibles ampliaciones que se podrían hacer a este proyecto:

- **Admisión de varios docentes en la misma asignatura.** Esto permitiría que las preguntas, exámenes y resultados podrían ser gestionados por varios profesores. Los cambios a hacer serían:
  - Disponer de una aplicación común.
  - Permitir a los profesores compartir recursos (a través de, por ejemplo, un enlace de invitación).
  - Añadir como atributos de algunos modelos (nodos de la jerarquía, preguntas, exámenes...) los usuarios invitados a ciertos modelos y los permisos para los mismos (por ejemplo, creación, borrado, edición o posibilidad de compartir con otros usuarios).

Puesto que todos los usuarios utilizarían la misma API, la concurrencia de acceso a recursos sería gestionada por la propia API.

- **Integración con otras plataformas.** Pese a que anteriormente se ha mencionado que los usuarios que se esperan de esta aplicación son docentes sin importar su plataforma educativa, cabe destacar que una gran cantidad de comunidades educativas utilizan instancias de Moodle<sup>™</sup>, por lo que sería interesante que esta aplicación web tuviera una versión completamente integrada en Moodle<sup>™</sup> a través de un módulo como se pensó inicialmente para este proyecto. Por otra parte, aunque se ha añadido una amplia variedad de archivos de importación y exportación, sería interesante contar con aún más opciones de importación y exportación que fueran compatibles con otras plataformas que se ofrecen actualmente más allá de Moodle<sup>™</sup>.
- **Incorporación de técnicas de Inteligencia Artificial.** Pese a que el algoritmo es bastante útil a la hora de seleccionar preguntas para una sección, cabe la posibilidad de afinar aún más la selección de preguntas utilizando técnicas de Inteligencia Artificial que seleccionasen las mejores preguntas para cada caso, basándose en criterios más allá de los atributos básicos de la pregunta (por ejemplo, la puntuación obtenida de dicha pregunta en otros exámenes o la combinación de preguntas entre sí). Por otra parte, el auge en los últimos años de las Inteligencias Artificiales generativas es un hecho, por lo que sería interesante contar con una funcionalidad

que, a través de una Inteligencia Artificial generativa, pudiera generar preguntas para una asignatura en base a los datos que ya se han especificado de la asignatura, como la jerarquía de ítems u otras preguntas.

- **Incorporación de análisis de resultados más avanzado.** El componente de visualización de resultados es bastante útil para observar el rendimiento de un examen: sin embargo, los estadísticos que se generan podrían contar con algo más de variabilidad, especialmente a la hora de contar con muchos resultados de un mismo examen.
- **Interoperabilidad con otras plataformas.**

## 5. Bibliografía

### Referencias

- [1] Aristeidis Bampakos. *Angular Projects: Build modern web apps in Angular 16 with 10 different projects and cutting-edge technologies*. Packt Publishing, 2023. ISBN: 9781803239118.
- [2] Naomi Ceder. *The Quick Python Book*. Simon y Schuster, 2018. ISBN: 9781638353430.
- [3] *FlexiQuiz*. URL: <https://www.flexiquiz.com/> (visitado 21-10-2023).
- [4] *Formato Moodle Aiken*. URL: [https://docs.moodle.org/all/es/Formato\\_Aiken](https://docs.moodle.org/all/es/Formato_Aiken) (visitado 24-04-2024).
- [5] *Formato Moodle GIFT*. URL: [https://docs.moodle.org/all/es/Formato\\_GIFT](https://docs.moodle.org/all/es/Formato_GIFT) (visitado 24-04-2024).
- [6] *Formato Moodle XML*. URL: [https://docs.moodle.org/all/es/Formato\\_Moodle\\_XML](https://docs.moodle.org/all/es/Formato_Moodle_XML) (visitado 24-04-2024).
- [7] *Fyrexbox*. URL: <https://www.fyrexbox.com/> (visitado 21-10-2023).
- [8] *Generador de Exámenes*. URL: <https://javierfpanadero.github.io/generador-examenes/> (visitado 21-10-2023).
- [9] Richard D. Copeland Jason Myers Rick Copeland. *Essential SQLAlchemy*. O'Reilly Media, 2015. ISBN: 9781491916568.
- [10] *Kahoot!* URL: <https://kahoot.com/> (visitado 21-10-2023).
- [11] Venkata Keerti Kotaru. *Angular for Material Design: Leverage Angular Material and TypeScript to Build a Rich User Interface for Web Apps*. Apress, 2019. ISBN: 9781484254332.
- [12] *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*. URL: <https://www.boe.es/eli/es/lo/2018/12/05/3/con> (visitado 02-02-2024).
- [13] Charles Leifer Matt Copperwaite. *Learning Flask Framework*. Packt Publishing Ltd, 2018. ISBN: 9781783983377.
- [14] Russ McKendrick. *Mastering Docker: Enhance Your Containerization and DevOps Skills to Deliver Production-ready Applications*. Packt Publishing, 2020. ISBN: 9781839213519.
- [15] *Moodle*. URL: <https://moodle.org/> (visitado 28-10-2023).
- [16] *Oficina de software libre: Liberación de Software y Otro Tipo de Material*. URL: <https://osl.ugr.es/liberacion-de-software/> (visitado 15-05-2024).
- [17] Leo S.Hsu Regina O. Obe. *PostgreSQL: Up and Running*. 3rd. O'Reilly Media, 2017. ISBN: 978-1491963418.

- 
- [18] Sebastian Springer. *Node.js: The Comprehensive Guide (Rheinwerk Computing)*. Rheinwerk Computing, 2023. ISBN: 978-1493222929.
- [19] *Test Wizard*. URL: <https://www.testwizard.com/> (visitado 21-10-2023).