



Universidad de Oviedo  
Universidá d'Uviéu  
University of Oviedo



Escuela de  
Ingeniería  
Informática  
Universidad de Oviedo



Trabajo de  
Desarrollo

# *DISEÑO E IMPLANTACIÓN DE UN SISTEMA CUESTIONARIOS EN LÍNEA (AUDIENCE RESPONSE SYSTEM)*

**GRADO EN INGENIERÍA INFORMÁTICA DEL  
SOFTWARE  
TRABAJO DE FIN DE GRADO**

**AUTOR**

Daniel Ferreira Gómez

**TUTOR**

Miguel Riesco Albizu

**Julio 2023**



Yo, Daniel Ferreira Gómez, declaro expresamente que el presente proyecto y su documentación, basada en la plantilla elaborada por JOSÉ MANUEL REDONDO LÓPEZ [1], ha sido íntegramente realizado por mí, constituyendo así una obra original. Además, declaro que las fuentes de información utilizadas han sido citadas a lo largo del documento y residen en la bibliografía de este.

# Declaración Responsable

**El alumno:** Daniel Ferreira Gómez

**Con DNI:**

**Y UO:**

## **DECLARA**

Que esta obra es completamente original y se han citado debidamente las fuentes utilizadas durante la realización de esta.

Y para que conste, lo firma en Oviedo, a 16 de abril de 2023

**Firmado:**

# Índice de contenido

<b>Capítulo 1</b>	<b>¿Qué es este Trabajo?</b>	<b>15</b>
1.1	Resumen	15
1.2	Palabras clave	16
1.3	Abstract	17
1.4	Keywords	18
<b>Capítulo 2</b>	<b>Planificación del Sistema de Información</b>	<b>19</b>
2.1	PSI 1: Inicio del Plan de Sistemas de Información	20
2.1.1	PSI 1.1: Análisis de la Necesidad del PSI	20
2.2	PSI 2: Definición y Organización del PSI	20
2.2.1	PSI 2.1: Especificación del Ámbito y Alcance	20
2.3	PSI 3: Estudio de la Información Relevante	22
2.3.1	PSI 1.2: Identificación del Alcance del PSI	22
2.3.2	PSI 3.1: Selección y Análisis de Antecedentes	22
<b>Capítulo 3</b>	<b>PSI 7: Definición de la Arquitectura Tecnológica</b>	<b>25</b>
3.1	PSI 7.1: Identificación de las Necesidades de Infraestructura Tecnológica	26
3.1.1	API Rest (Backend)	26
3.1.2	Aplicación web (Frontend)	27
3.2	PSI 7.2: Selección de la Arquitectura Tecnológica	28
<b>Capítulo 4</b>	<b>Estudio de Viabilidad del Sistema</b>	<b>31</b>
4.1	EVS 4, 5 y 6: Estudio y Valoración de Alternativas de Solución y Selección de Alternativa Final	32
4.1.1	Aplicación (Cliente)	32
4.1.2	API Rest (Servidor)	35
<b>Capítulo 5</b>	<b>Planificación y Gestión del TFG</b>	<b>37</b>
5.1	Planificación del proyecto	38
5.1.1	Planificación Inicial	38
5.1.2	Riesgos	42
5.1.3	Presupuesto Inicial	42
<b>Capítulo 6</b>	<b>Análisis del Sistema de Información</b>	<b>45</b>

6.1	ASI 1: Definición del Sistema .....	46
6.1.1	Determinación del Alcance del Sistema .....	46
6.2	ASI 2: Establecimiento de Requisitos .....	47
6.2.1	Obtención de los Requisitos del Sistema .....	47
6.2.2	Identificación de Actores del Sistema.....	52
6.2.3	Especificación de Casos de Uso .....	53
6.3	ASI 3: Identificación de Subsistemas de Análisis.....	58
6.3.1	Descripción de los Subsistemas .....	58
6.3.2	Descripción de los Interfaces entre Subsistemas .....	59
6.4	ASI 4: Análisis de los Casos de Uso .....	61
6.4.1	Iniciar sesión .....	61
6.4.2	Crear usuario.....	62
6.4.3	Crear curso .....	63
6.4.4	Modificar usuario.....	64
6.4.5	Modificar curso .....	65
6.4.6	Eliminar usuario .....	66
6.4.7	Eliminar curso .....	67
6.4.8	Importar usuarios .....	67
6.4.9	Importar cursos.....	68
6.4.10	Crear pregunta .....	69
6.4.11	Crear cuestionario.....	70
6.4.12	Modificar pregunta .....	71
6.4.13	Modificar cuestionario.....	72
6.4.14	Eliminar pregunta .....	73
6.4.15	Eliminar cuestionario .....	74
6.4.16	Asignar cuestionario a un curso.....	75
6.4.17	Asignar pregunta a un curso .....	76
6.4.18	Asignar usuario a un curso.....	76
6.4.19	Lanzar cuestionario en línea .....	77
6.4.20	Consultar cuestionarios abiertos .....	79
6.4.21	Cambiar contraseña .....	80
6.4.22	Jugar cuestionario en línea .....	81

6.4.23	Consultar resultados .....	83
6.5	ASI 5: Análisis de Clases.....	84
6.5.1	Diagrama de Clases .....	84
6.5.2	Descripción de las Clases .....	86
6.6	ASI 8: Definición de Interfaces de Usuario .....	104
6.6.1	Descripción de la Interfaz .....	104
6.6.2	Descripción del Comportamiento de la Interfaz.....	120
6.6.3	Diagrama de Navegabilidad .....	121
6.7	ASI 10: Especificación del Plan de Pruebas .....	122
6.7.1	Pruebas unitarias .....	122
6.7.2	Pruebas de integración y de sistema .....	122
6.7.3	Pruebas de usabilidad .....	123
6.7.4	Pruebas de accesibilidad.....	123
<b>Capítulo 7</b>	<b>Diseño del Sistema de Información .....</b>	<b>125</b>
7.1	DSI 3: Diseño de Casos de Uso Reales.....	126
7.1.1	Creación de cuestionarios.....	126
7.1.2	Jugar cuestionario en línea .....	127
7.2	DSI 4: Diseño de Clases .....	128
7.2.1	Diagrama de Clases.....	128
7.3	DSI 5: Diseño de la Arquitectura de Módulos del Sistema .....	132
7.3.1	Diagrama de Paquetes .....	132
7.3.2	Diagrama de despliegue .....	134
7.4	DSI 6: Diseño Físico de Datos .....	134
7.4.1	Descripción del SGBD usado .....	134
7.4.2	Integración del SGBD en nuestro Sistema .....	135
7.4.3	E-R del modelo de dominio.....	136
7.5	DSI 10: Especificación Técnica del Plan de Pruebas.....	136
7.5.1	Pruebas Unitarias.....	136
7.5.2	Pruebas de Integración .....	160
7.5.3	Pruebas de Usabilidad .....	167
7.5.4	Pruebas de Accesibilidad .....	170
7.5.5	Pruebas de Rendimiento.....	171

<b>Capítulo 8</b>	<b>Construcción del Sistema de Información</b>	<b>173</b>
8.1	CSI 1: Preparación del Entorno de Generación y Construcción	174
8.1.1	Estándares y normas seguidos	174
8.1.2	Lenguajes de programación	174
8.1.3	Herramientas y programas usados para el desarrollo	176
8.2	CSI 2: Generación del Código de los Componentes y Procedimientos	177
8.2.1	Problemas encontrados	177
8.2.2	Descripción detallada de las clases	179
8.3	CSI 3: Ejecución de las Pruebas Unitarias	181
8.3.1	Herramientas utilizadas	181
8.3.2	Resultado de las pruebas	181
8.4	CSI 4: Ejecución de las Pruebas de Integración	196
8.4.1	Herramientas utilizadas	196
8.4.2	Resultado de las pruebas	196
8.5	CSI 5: Ejecución de las Pruebas del Sistema	205
8.5.1	Pruebas de Usabilidad	205
8.5.2	Pruebas de Accesibilidad	211
8.6	CSI 6: Elaboración de los Manuales de Usuario	224
8.6.1	Manual de Instalación	224
8.6.2	Manual de Ejecución y Despliegue	224
8.6.3	Manual de Usuario	226
<b>Capítulo 9</b>	<b>Conclusiones y Ampliaciones</b>	<b>237</b>
9.1	Conclusiones	238
9.2	Ampliaciones	239
9.2.1	Realización de exámenes tipo test	239
9.2.2	Aplicación para móvil	239
9.2.3	Pruebas automáticas para los cuestionarios en línea	239
9.2.4	Realizar cuestionarios en diferido	239
9.2.5	Poder filtrar los datos que se muestran	240
9.2.6	Autenticación utilizando credenciales de Uniovi	240
<b>Capítulo 10</b>	<b>Bibliografía</b>	<b>241</b>
<b>Apéndices</b>		<b>243</b>



Plan de gestión de riesgos.....244  
Contenido entregado .....249



# Índice de Tablas

TABLA 1 PLANIFICACIÓN TEMPORAL DEL PROYECTO .....	41
TABLA 2 PARTIDA 1 DESARROLLO DEL SISTEMA.....	42
TABLA 3 PARTIDA 2 MEMORIA .....	43
TABLA 4 COSTES DE LOS MEDIOS DE PRODUCCIÓN .....	43
TABLA 5 COSTES INDIRECTOS .....	43
TABLA 6 PRESUPUESTO DEL CLIENTE .....	44
TABLA 7 CASOS DE PRUEBA INICIO DE SESIÓN .....	160
TABLA 8 CASOS DE PRUEBA INICIO DE SESIÓN .....	196
TABLA 9 RESPUESTAS A LAS PREGUNTAS SOBRE LA APP DEL SUJETO 1 .....	208
TABLA 10 RESPUESTAS A LAS PREGUNTAS SOBRE LA APP DEL SUJETO 2 .....	209
TABLA 11 RESPUESTAS A LAS PREGUNTAS SOBRE LA APP DEL SUJETO 3 .....	210
TABLA 12 MATRIZ DE PROBABILIDAD E IMPACTO .....	246
TABLA 13 TABLA DE RIESGOS PRIORIZADOS.....	246
TABLA 14 ESTRATEGIA Y RESPUESTA ANTE LOS RIESGOS .....	248

# Índice de Ilustraciones

ILUSTRACIÓN 1 DIAGRAMA DE GANTT .....	38
ILUSTRACIÓN 2 DIAGRAMA DE CASOS DE USO 1 .....	53
ILUSTRACIÓN 3 DIAGRAMA DE CASOS DE USO 2 .....	53
ILUSTRACIÓN 4 DIAGRAMA DE CASOS DE USO 3 .....	55
ILUSTRACIÓN 5 DIAGRAMA DE CASOS DE USO 4 .....	57
ILUSTRACIÓN 6 TABLAS DE LA BASE DE DATOS.....	58
ILUSTRACIÓN 7 DIAGRAMA DE COMUNICACIONES ENTRE SUBSISTEMAS .....	60
ILUSTRACIÓN 8 DIAGRAMA DE ROBUSTEZ DE INICIO DE SESIÓN.....	61
ILUSTRACIÓN 9 DIAGRAMA DE ROBUSTEZ DE LA CREACIÓN DE UN USUARIO .....	62
ILUSTRACIÓN 10 DIAGRAMA DE ROBUSTEZ DE LA MODIFICACIÓN DE UN USUARIO.....	64
ILUSTRACIÓN 11 DIAGRAMA DE ROBUSTEZ DE LA ELIMINACIÓN DE UN USUARIO.....	66
ILUSTRACIÓN 12 DIAGRAMA DE ROBUSTEZ DEL IMPORTE DE USUARIOS .....	67
ILUSTRACIÓN 13 DIAGRAMA DE ROBUSTEZ DE CRACIÓN DE PREGUNTA.....	69
ILUSTRACIÓN 14 DIAGRAMA DE ROBUSTEZ DE LA MODIFICACIÓN DE PREGUNTA.....	71
ILUSTRACIÓN 15 DIAGRAMA DE ROBUSTEZ DE LA ELIMINACIÓN DE UNA PREGUNTA .....	73
ILUSTRACIÓN 16 DIAGRAMA DE ROBUSTEZ DE INSERCIÓN DE CUESTIONARIO EN UN CURSO.....	75
ILUSTRACIÓN 17 DIAGRAMA DE ROBUSTEZ DE LANZAMIENTO DE CUESTIONARIO EN LÍNEA .....	77
ILUSTRACIÓN 18 DIAGRAMA DE ROBUSTEZ PARA LA CONSULTA DE CUESTIONARIOS ABIERTOS .....	79
ILUSTRACIÓN 19 DIAGRAMA DE ROBUSTEZ DEL CAMBIO DE CONTRASEÑA .....	80
ILUSTRACIÓN 20 DIAGRAMA DE ROBUSTEZ PARA JUGAR UN CUESTIONARIO EN LÍNEA .....	81
ILUSTRACIÓN 21 DIAGRAMA DE ROBUSTEZ DE LA CONSULTA DE RESULTADOS .....	83
ILUSTRACIÓN 22 DIAGRAMA DE CLASES DEL SERVIDOR WEB.....	85
ILUSTRACIÓN 23 DIAGRAMA DE CLASES DE LA APLICACIÓN WEB .....	86
ILUSTRACIÓN 24 MOCKUP DE LA PANTALLA DE INICIO DE SESIÓN.....	104

ILUSTRACIÓN 25 MOCKUP DE LA PANTALLA DE LISTADO DE USUARIOS .....	105
ILUSTRACIÓN 26 MOCKUP DE LA PANTALLA DE CREACIÓN DE USUARIOS .....	106
ILUSTRACIÓN 27 MOCKUP DE LA PANTALLA DE BIBLIOTECA (CURSOS) .....	107
ILUSTRACIÓN 28 MOCKUP DE LA PANTALLA DE LA BIBLIOTECA (CUESTIONARIOS) .....	108
ILUSTRACIÓN 29 MOCKUP DE LA PANTALLA DE LA BIBLIOTECA (PREGUNTAS) .....	109
ILUSTRACIÓN 30 MOCKUP DE LA PANTALLA DE CREACIÓN DE PREGUNTAS .....	110
ILUSTRACIÓN 31 MOCKUP DE LA PANTALLA DE CREACIÓN DE CUESTIONARIO 1 .....	111
ILUSTRACIÓN 32 MOCKUP DE LA PANTALLA DE CREACIÓN DE CUESTIONARIO 2 .....	112
ILUSTRACIÓN 33 MOCKUP DE LA PANTALLA DE DETALLES DE UN CURSO (PROFESOR) .....	113
ILUSTRACIÓN 34 MOCKUP DE LA PANTALLA PRINCIPAL DEL ALUMNO .....	114
ILUSTRACIÓN 35 MOCKUP DE PANTALLA DE PERFIL .....	115
ILUSTRACIÓN 36 MOCKUP DE LA PANTALLA E DETALLES DE UN CURSO (ALUMNO).....	116
ILUSTRACIÓN 37 MOCKUP DE LA PANTALLA DE JUEGO (PREGUNTA).....	117
ILUSTRACIÓN 38 MOCKAUP DE LA PANTALLA DE JUEGO (RESPUESTA).....	118
ILUSTRACIÓN 39 MOCKUP DE LA PANTALLA DE JUEGO (PODIO).....	119
ILUSTRACIÓN 40 MOCKUP DE PANTALLA DE JUEGO (PUNTUACIÓN) .....	120
ILUSTRACIÓN 41 DIAGRAMA DE NAVEGABILIDAD .....	121
ILUSTRACIÓN 42 DIAGRAMA DE INTERACCIÓN DE LA CREACIÓN DE CUESTIONARIOS .....	126
ILUSTRACIÓN 43 DIAGRAMA DE INTERACCIÓN DEL DESARROLLO DE UN CUESTIONARIO EN LÍNEA .....	127
ILUSTRACIÓN 44 DIAGRAMA DE CLASES DE LOS COMPONENTES DEL ESTUDIANTE .....	129
ILUSTRACIÓN 45 DIAGRAMA DE CLASES DE LOS COMPONENTES DEL PROFESOR.....	129
ILUSTRACIÓN 46 DIAGRAMA DE CLASES DE LOS COMPONENTES DEL ADMINISTRADOR.....	129
ILUSTRACIÓN 47 DIAGRAMA DE CLASES DE LOS COMPONENTES COMPARTIDOS .....	130
ILUSTRACIÓN 48 DIAGRAMA DE CLASES DE LOS SERVICIOS .....	130
ILUSTRACIÓN 49 DIAGRAMA DE CLASES DE LOS CONTROLADORES .....	131
ILUSTRACIÓN 50 DIAGRAMA DE CLASES DE LOS MODELOS.....	131
ILUSTRACIÓN 51 DIAGRAMA DE CLASES DE LAS RUTAS.....	132
ILUSTRACIÓN 52 DIAGRAMA DE PAQUETES DEL SISTEMA.....	133
ILUSTRACIÓN 53 DIAGRAMA DE DESPLIEGUE DEL SISTEMA .....	134
ILUSTRACIÓN 54 LOGO DE POSTGRESQL.....	134
ILUSTRACIÓN 55 CONEXIÓN A LA BASE DE DATOS MEDIANTE PRISMA.....	135
ILUSTRACIÓN 56 MODELO ENTIDAD RELACIÓN .....	136
ILUSTRACIÓN 57 ROOMS EN SOCKET.IO.....	178
ILUSTRACIÓN 58 TOTAL DE TESTS UNITARIOS DE LA API REST .....	181
ILUSTRACIÓN 59 TESTS UNITARIOS DEL CONTROLADOR DE USUARIOS 1 .....	182
ILUSTRACIÓN 60 TESTS UNITARIOS DEL CONTROLADOR DE USUARIOS 2 .....	182
ILUSTRACIÓN 61 TESTS DEL CONTROLADOR DE USUARIOS 3 .....	183
ILUSTRACIÓN 62 TESTS DEL CONTROLADOR DE CURSOS 1.....	183
ILUSTRACIÓN 63 TESTS DEL CONTROLADOR DE CURSOS 2 .....	184
ILUSTRACIÓN 64 TESTS DEL CONTROLADOR DE CURSOS 3 .....	184
ILUSTRACIÓN 65 TESTS DEL CONTROLADOR DE PREGUNTAS 1 .....	185
ILUSTRACIÓN 66 TESTS DEL CONTROLADOR DE PREGUNTAS 2 .....	186
ILUSTRACIÓN 67 TESTS DEL CONTROLADOR DE PREGUNTAS 3 .....	186
ILUSTRACIÓN 68 TESTS DEL CONTROLADOR DE CUESTIONARIOS 1.....	187
ILUSTRACIÓN 69 TESTS DEL CONTROLADOR DE CUESTIONARIOS 2.....	187
ILUSTRACIÓN 70 TESTS DEL CONTROLADOR DE CUESTIONARIOS 3.....	188
ILUSTRACIÓN 71 TESTS DEL CONTROLADOR DE JUEGOS 1 .....	188
ILUSTRACIÓN 72 TESTS DEL CONTROLADOR DE JUEGOS 2 .....	189
ILUSTRACIÓN 73 TESTS DEL CONTROLADOR DE JUEGOS 3 .....	189

ILUSTRACIÓN 74 TESTS DEL CONTROLADOR DE JUEGOS 4 .....	190
ILUSTRACIÓN 75 TESTS UNITARIOS DE LA APLICACIÓN WEB 1 .....	191
ILUSTRACIÓN 76 TESTS UNITARIOS DE LA APLICACIÓN WEB 2 .....	192
ILUSTRACIÓN 77 TESTS UNITARIOS DE LA APLICACIÓN WEB 3 .....	192
ILUSTRACIÓN 78 TESTS UNITARIOS DE LA APLICACIÓN WEB 4 .....	193
ILUSTRACIÓN 79 TESTS UNITARIOS DE LA APLICACIÓN WEB 5 .....	194
ILUSTRACIÓN 80 TESTS UNITARIOS DE LA APLICACIÓN WEB 6 .....	195
ILUSTRACIÓN 81 TESTS UNITARIOS DE LA APLICACIÓN WEB 7 .....	195
ILUSTRACIÓN 82 CUESTIONARIO DE EVALUACIÓN DEL SUJETO 1.....	205
ILUSTRACIÓN 83 CUESTIONARIO DE EVALUACIÓN DEL SUJETO 2.....	206
ILUSTRACIÓN 84 CUESTIONARIO DE EVALUACIÓN DEL SUJETO 3.....	207
ILUSTRACIÓN 85 INFORME DE WAVE DEL LISTADO DE USUARIOS DEL ADMINISTRADOR .....	213
ILUSTRACIÓN 86 INFORME DE WAVE PARA LA PANTALLA DE CREACIÓN DE CURSOS .....	214
ILUSTRACIÓN 87 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE INICIO DE SESIÓN .....	214
ILUSTRACIÓN 88 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE LISTADO DE USUARIOS.....	215
ILUSTRACIÓN 89 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE CREACIÓN DE USUARIOS.....	215
ILUSTRACIÓN 90 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE INICIO DEL PROFESOR .....	215
ILUSTRACIÓN 91 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE CREACIÓN DE PREGUNTAS.....	215
ILUSTRACIÓN 92 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE CREACIÓN DE CUESTIONARIO .....	216
ILUSTRACIÓN 93 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE DETALLES DE CURSO DEL PROFESOR .....	216
ILUSTRACIÓN 94 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE JUEGO DEL PROFESOR.....	216
ILUSTRACIÓN 95 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE JUEGO DEL ALUMNO.....	216
ILUSTRACIÓN 96 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE INICIO DEL ALUMNO .....	217
ILUSTRACIÓN 97 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE DETALLES DE CURSO DEL ALUMNO .....	217
ILUSTRACIÓN 98 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE REVISIÓN DE CUESTIONARIO .....	217
ILUSTRACIÓN 99 INFORME DE LIGHTHOUSE PARA LA PANTALLA DE PERFIL.....	217
ILUSTRACIÓN 100 EJEMPLO DE .ENV.....	225
ILUSTRACIÓN 101 HTTPD-SSL.CONF .....	226
ILUSTRACIÓN 102 PANTALLA DE INICIO DE SESIÓN DEL MANUAL DE USUARIO.....	227
ILUSTRACIÓN 103 LISTA DE USUARIOS DEL ADMINISTRADOR.....	228
ILUSTRACIÓN 104 FORMULARIO DE CREACIÓN DE NUEVO USUARIO DEL ADMINISTRADOR .....	228
ILUSTRACIÓN 105 FORMULARIO DE ACTUALIZACIÓN DE UN USUARIO DEL ADMINISTRADOR.....	228
ILUSTRACIÓN 106 BOTÓN PARA CERRAR SESIÓN.....	229
ILUSTRACIÓN 107 PANTALLA DE LIBRERÍA DEL PROFESOR .....	229
ILUSTRACIÓN 108 FORMULARIO DE CREACIÓN DE PREGUNTAS .....	230
ILUSTRACIÓN 109 PREGUNTA DESPLEGADA .....	230
ILUSTRACIÓN 110 LISTA DE CUESTIONARIOS DE LA LIBRERÍA DEL PROFESOR.....	230
ILUSTRACIÓN 111 FORMULARIO DE CREACIÓN DE CUESTIONARIO DEL PROFESOR .....	231
ILUSTRACIÓN 112 LISTA DE CURSOS DE LA LIBRERÍA DEL PROFESOR .....	231
ILUSTRACIÓN 113 PANTALLA DE DETALLES DEL CURSO.....	232
ILUSTRACIÓN 114 DIÁLOGO DE CREACIÓN DE JUEGO DEL PROFESOR .....	232
ILUSTRACIÓN 115 PANTALLA DE LISTA DE ESPERA DE ALUMNOS .....	233
ILUSTRACIÓN 116 PANTALLA DE RESULTADO DE UN JUEGO COMO PROFESOR .....	233
ILUSTRACIÓN 117 PANTALLA DE INICIO DEL ALUMNO .....	233
ILUSTRACIÓN 118 SALA DE ESPERA DEL ALUMNO .....	234
ILUSTRACIÓN 119 PANTALLA DE PREGUNTA DEL ALUMNO.....	234
ILUSTRACIÓN 120 RESPUESTA CORRECTA.....	234
ILUSTRACIÓN 121 RESPUESTA INCORRECTA .....	234
ILUSTRACIÓN 122 PUNTUACIÓN DEL ALUMNO .....	235

ILUSTRACIÓN 123 PANTALLA DE DETALLES DE CURSO DEL ALUMNO .....	235
ILUSTRACIÓN 124 PANTALLA DE REVISIÓN DE CUESTIONARIO .....	235
ILUSTRACIÓN 125 PANTALLA DEL PERFIL .....	236
ILUSTRACIÓN 126 TABLA DEL CONTENIDO ENTREGADO .....	249

# Capítulo 1 ¿QUÉ ES ESTE TRABAJO?

## 1.1 RESUMEN

---

Este trabajo describe el diseño e implantación de un sistema de cuestionarios en línea, similar a otros ya existentes como Kahoot o Vevox, por ejemplo. Se trata de una aplicación que resuelva alguno de los problemas que los dos anteriores presentan, permitiendo la gestión y la publicación de cuestionarios por parte del profesor, así como la posibilidad de responderlos, por parte del alumnado. La clave principal de este sistema es la rapidez de puesta en juego, es decir, el profesor iniciará un cuestionario en línea y todos los alumnos podrán entrar inmediatamente.

## 1.2 PALABRAS CLAVE

---

Cuestionario, Kahoot, online, Angular, Nodejs, API, WebSocket, Socketio



## 1.3 ABSTRACT

---

This work describes the design and implementation of an online questionnaire system, like other existing ones such as Kahoot or Vevox, for example. It is an application that solves some of the problems that the two previous one's present, allowing the management and publication of questionnaires by the teacher, as well as the possibility of answering them, by the students. The main key to this system is the speed of implementation, that is, the teacher will start an online questionnaire and all the students will be able to enter immediately.

## 1.4 KEYWORDS

---

Survey, Kahoot, online, Angular, Nodejs, API, WebSocket, Socketio

# Capítulo 2 PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

**FASE DE PLANIFICACIÓN**

**PSI**



## 2.1 PSI 1: INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN

---

### 2.1.1 PSI 1.1: Análisis de la Necesidad del PSI

El director del trabajo ha solicitado el desarrollo de un sistema informático que sirva como plataforma para la realización de cuestionarios en línea en las clases impartidas en la escuela. Actualmente se está utilizando otro sistema, Kahoot, el cual se quiere sustituir por ser éste de pago y por tener ciertas restricciones. Para ello, el sistema gestionará las cuentas, tanto de los profesores como de los alumnos, y permitirá lanzar dichos cuestionarios en línea. El software debe permitir a los profesores gestionar cursos, pudiendo añadir a éstos, cuestionarios, preguntas y alumnos, haciendo que solo los alumnos puedan unirse a los cuestionarios lanzados en un curso determinado. Además, el sistema también tendrá un perfil de administrador, que permitirá la gestión de los cursos y usuarios.

## 2.2 PSI 2: DEFINICIÓN Y ORGANIZACIÓN DEL PSI

---

### 2.2.1 PSI 2.1: Especificación del Ámbito y Alcance

Con el fin de exponer más detalladamente los requisitos del sistema, se dividirá el proyecto en dos partes: la API (servidor) y la aplicación web (cliente). Para ambas partes se establecerán los objetivos que éstas deberán cumplir.

#### ***Aplicación (cliente)***

Se desarrollará una aplicación que permita gestionar diferentes tipos de usuarios, ya sea el usuario administrador, un profesor o un alumno. En el caso del administrador, podrá ver la lista entera de usuarios registrados en el sistema, pudiendo modificarlos, eliminarlos, crearlos e importarlos. También podrá gestionar los cursos, cuya funcionalidad es análoga a la de los usuarios. Por otro lado, el profesor podrá crear preguntas de diferentes tipos, ya sean con multi opción, verdadero o falso o de respuesta corta, modificarlas, eliminarlas y asignarlas varios cursos. Al igual que con las preguntas, el profesor puede crear, modificar y eliminar cuestionarios, compuestos por las preguntas creadas con anterioridad, pudiendo asignarlo también a varios cursos. El profesor tendrá una vista donde podrá llevar a cabo todas estas funciones, así como ver todos los cursos disponibles. Dentro de la vista de cada curso, el profesor podrá crear lanzar los cuestionarios online, asignar cuestionarios, preguntas y alumnos al curso. Finalmente, los alumnos tendrán una vista donde indica la lista de cursos a los que pertenece y van apareciendo los cuestionarios, a medida que los profesores los lanzan. Un alumno solo puede ver los cuestionarios en juego de los cursos a los que pertenece. Además, tendrán una sección donde cambiar la contraseña y ver los resultados de los cuestionarios jugados.

Los objetivos principales de la aplicación son:

- Gestión de usuarios por parte del administrador.



- Gestión de preguntas, cuestionarios y cursos por parte del profesor.
- Creación y posterior lanzamiento de cuestionarios en línea por parte del profesor.
- Que el alumno tenga la posibilidad de ver los cuestionarios en línea actuales y poder entrar a jugarlos.
- Registrar los resultados de los alumnos para su posterior visualización.

### **API Rest (servidor)**

Se desarrollará una API Rest que será la encargada de proveer todos los datos necesarios para la aplicación. Además de una API también se desarrollará en la parte del servidor la funcionalidad correspondiente con el juego de cuestionarios en línea que hará posible la sincronización online entre los jugadores (alumnos) y el profesor (host).

El objetivo principal que deberá cumplir es proveer una interfaz para:

- La gestión de los usuarios.
- La gestión de los cursos.
- La gestión de las preguntas.
- La gestión de los resultados.
- La gestión de los cuestionarios en línea.



## 2.3 PSI 3: ESTUDIO DE LA INFORMACIÓN RELEVANTE

---

### 2.3.1 PSI 1.2: Identificación del Alcance del PSI

Como se ha expuesto en el anterior apartado, el objetivo del trabajo es realizar un sistema similar a Kahoot pero que corrija algunas restricciones. Otro objetivo es basar el software a un entorno académico, pues está enfocado al uso por parte del profesorado y alumnado de la escuela. Concretamente, los requisitos a cumplir por el proyecto son los siguientes:

- El administrador del sistema podrá gestionar los dos tipos de usuarios: profesor y alumno. Esta gestión se corresponde con altas, bajas y modificaciones de cursos y usuarios. Asimismo, podrá asignar roles a los usuarios para asignarlos a uno de los dos tipos de usuarios mencionados, y tendrá la opción de importar tanto cursos como usuarios.
- El usuario de tipo profesor tendrá la capacidad de crear preguntas de distinto tipo: multi opción, verdadero o falso y de respuesta corta. Además, tendrá la opción de importar y exportar preguntas en formato csv y podrá incluir imágenes en el enunciado de la pregunta.
- Los profesores también podrán crear cuestionarios con las preguntas que hayan creado anteriormente, pudiendo añadir preguntas, eliminarlas y cambiarlas de orden.
- Los profesores, además, podrán asignar a los cursos las preguntas y cuestionarios que hayan creado, así como usuarios de tipo alumno.
- Los usuarios de tipo alumno, podrán ver los juegos lanzados actualmente y participar en ellos inmediatamente. Además, los alumnos podrán gestionar su perfil, así como ver los resultados de sus participaciones.

### 2.3.2 PSI 3.1: Selección y Análisis de Antecedentes

Con el propósito de realizar el trabajo es necesario el estudio de determinados conceptos teóricos y tecnologías, así como de otros sistemas ya existentes con una funcionalidad similar al proyecto que se quiere desarrollar.

#### 2.3.2.1 *Conceptos teóricos*

Cómo se ha comentado antes, es necesario realizar un estudio previo de ciertos aspectos teóricos necesarios para construir el sistema.

##### 2.3.2.1.1 SPA

Una *single-page application* (SPA), o aplicación de página única, es una aplicación web o un sitio web que cabe en una sola página con el propósito de dar una experiencia más fluida a los usuarios, como si fuera una aplicación de escritorio. En un SPA todos los códigos de HTML, JavaScript, y CSS se cargan una sola vez, los recursos necesarios se cargan dinámicamente cuando lo requiera la página,



normalmente como respuesta a las acciones del usuario. La página no tiene que cargarse de nuevo en ningún punto del proceso y tampoco es necesario transferir a otra página las peticiones [2].

Para realizar este tipo de aplicaciones, hoy en día existen varios *frameworks* de *Javascript* que lo permiten, por lo que para este proyecto se elegirá uno de ellos para construir la aplicación.

#### 2.3.2.1.2 Websocket

WebSockets es una tecnología avanzada que hace posible abrir una sesión de comunicación interactiva entre el navegador del usuario y un servidor. Con esta API, se puede enviar mensajes a un servidor y recibir respuestas controladas por eventos sin tener que consultar al servidor para una respuesta [3].

Gracias a esta tecnología va a ser posible realizar cuestionarios en línea, donde los profesores y los alumnos estén en una misma sesión y se pueda llevar a cabo el juego.

#### 2.3.2.1.3 REST

REST no es un protocolo ni un estándar, sino más bien un conjunto de límites de arquitectura. Los desarrolladores de las API pueden implementarlo de distintas maneras [4]. Es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web [5].

### 2.3.2.2 Sistemas similares

Es importante también estudiar sistemas ya existentes que tengan una funcionalidad similar al sistema que se quiere construir, pues puede ser un buen punto de partida del que tomar referencias. Concretamente se verán dos, Kahoot y Vevox, pues son los dos que hemos utilizado en la carrera.

#### 2.3.2.2.1 Kahoot

Kahoot es una plataforma de aprendizaje en línea basada en juegos, diseñada para proporcionar un enfoque interactivo y divertido para la educación, la capacitación y las evaluaciones. Permite a los usuarios crear y participar en cuestionarios, encuestas y discusiones en tiempo real a través de dispositivos móviles u ordenadores. La principal ventaja que tiene es la cantidad de funcionalidades que ofrece, así como la personalización, tanto de los cuestionarios como de las preguntas. Este proyecto está basado en gran parte en este, salvando la gran distancia de funcionalidades que tiene Kahoot, puesto que este proyecto es más modesto. Por otro lado, Kahoot tiene una serie de inconvenientes que se intentarán resolver en este proyecto. Estas son la limitación de caracteres, tanto en el enunciado de la pregunta, como en las respuestas, la limitación del número de participantes y la necesidad de identificarse cada vez que se entra en un juego. Todo esto, sumado a que muchas funcionalidades son de pago, hacen que Kahoot pierda mucho valor en el caso de que escojas el plan gratuito.



#### 2.3.2.2.2 Vevox

Vevox es otra plataforma de participación y votación en tiempo real que se utiliza en entornos educativos, de negocios y eventos en vivo. Es muy parecida a Kahoot, y la funcionalidad principal es la que se quiere replicar en este proyecto. Sus principales ventajas son que ofrece la opción de participar de forma anónima y permite integrarse con herramientas de presentaciones, como PowerPoint, lo que permite a los presentadores incrustar encuestas y preguntas en sus diapositivas para una experiencia más fluida. Por poner algún inconveniente, podríamos decir que el costo de para desbloquear algunas de las funcionalidades avanzadas puede suponer una barrera para algunos usuarios.



# Capítulo 3 PSI 7: DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

**FASE DE PLANIFICACIÓN**

**PSI**



## 3.1 PSI 7.1: IDENTIFICACIÓN DE LAS NECESIDADES DE INFRAESTRUCTURA TECNOLÓGICA

---

Como se ha especificado en el alcance del sistema, es necesaria la construcción de dos elementos, una API Rest (Backend), para proveer al cliente de los datos necesarios, y una aplicación web (Frontend), con la cual llevar a cabo el propósito del trabajo. La decisión de ser una aplicación web y no ningún otro tipo se verá en el siguiente capítulo. A continuación, se mostrarán las alternativas estudiadas para ambos.

### 3.1.1 API Rest (Backend)

Existen varias opciones populares para construir una API Rest, ya sea construyéndola nosotros mismos o haciendo uso de plataformas de Backend. Algún ejemplo de estas últimas es:

- **Firebase:** Plataforma de desarrollo de ampliaciones en la nube que proporciona una base de datos en tiempo real y herramientas para construir APIs.
- **AWS API Gateway:** Amazon API Gateway es un servicio completamente administrado que facilita a los desarrolladores la creación, la publicación, el mantenimiento, el monitoreo y la protección de API a cualquier escala [6].

Por otro lado, si nos centramos en opciones para construirla desde cero, tenemos:

- **Express.js (Node.js):** Un framework minimalista y flexible para construir APIs Rest. Puede ser utilizado tanto con JavaScript como con TypeScript.
- **Django (Python):** Un framework web de alto nivel y basado en Python que incluye soporte para construir APIs Rest.
- **Ruby on Rails:** Un framework web completo y fácil de usar que incluye funcionalidades para construir APIs Rest.
- **Spring Boot:** es un framework de desarrollo de aplicaciones basado en Spring Framework, que simplifica la configuración y el desarrollo de aplicaciones Java. Proporciona una manera rápida y eficiente de construir APIs Rest con Java

Para construir una API Rest también es necesario elegir una base de datos donde almacenar toda la información. Para este propósito, también están disponibles las plataformas como Firebase antes mencionadas, que pueden ofrecer una base de datos en la nube. Sin embargo, también hay que abordar otras alternativas como:

- **MySQL:** Una base de datos relacional de código abierto ampliamente utilizada, con buena velocidad y escalabilidad.
- **PostgreSQL:** Otra base de datos relacional de código abierto con un enfoque en la confiabilidad y la conformidad con los estándares.



- **MongoDB:** Una base de datos NoSQL orientada a documentos, que proporciona flexibilidad en el esquema y escalabilidad horizontal.

Finalmente, aunque opcional, es estudiar el uso de un Mapeador de Objetos Relacional (ORM). El uso de uno de ellos puede hacer del desarrollo de la API mucho más fluida y sencilla. Algunos para las opciones anteriores son:

- **Hibernate:** Es parte integral del framework Spring y se utiliza ampliamente en aplicaciones Java.
- **Sequelize:** Es un ORM de Node.js que admite bases de datos relacionales como MySQL, PostgreSQL y SQLite. Proporciona funciones para definir modelos, realizar consultas y administrar las relaciones entre las entidades.
- **Django ORM:** este ORM viene incluido en el framework Django para Python y es una buena opción si se quiere desarrollar con este framework.
- **SQLAlchemy:** Es un ORM de Python que proporciona una capa de abstracción sobre diferentes bases de datos relacionales.

### 3.1.2 Aplicación web (Frontend)

Para construir una aplicación web, lo más popular es utilizar un framework de JavaScript, por lo que será lo que se utilice para este proyecto. Estos frameworks son:

- **React:** React es una biblioteca de JavaScript desarrollada por Facebook. Se utiliza para construir interfaces de usuario interactivas y reactivas. React utiliza un enfoque basado en componentes y ofrece un rendimiento eficiente a través de la manipulación virtual del DOM.
- **Angular:** Angular es un framework de JavaScript desarrollado por Google. Es utilizado para construir aplicaciones web de una sola página (SPA) y aplicaciones móviles. Angular utiliza el lenguaje TypeScript y ofrece una amplia gama de características, incluyendo enlace de datos bidireccional, inyección de dependencias y enrutamiento.
- **Vue.js:** Vue.js es un framework JavaScript progresivo y de código abierto que permite construir interfaces de usuario interactivas. Al igual que React, Vue.js se basa en componentes y ofrece una curva de aprendizaje suave. Es conocido por su flexibilidad y su capacidad para integrarse fácilmente en proyectos existentes.

Por último, quiero apuntar que todas las tecnologías expuestas son totalmente válidas para la construcción del sistema. Es cuestión del propio desarrollador elegir aquellas que le gusten más o con las que se encuentre más cómodo.



## 3.2 PSI 7.2: SELECCIÓN DE LA ARQUITECTURA TECNOLÓGICA

---

Una vez realizado el estudio de alternativas para la arquitectura tecnológica de la aplicación, se procede a seleccionar aquellas, que, por los motivos explicados a continuación, se han decidido usar.

### ***Framework para la API Rest***

Las dos principales elecciones para realizar el Backend son SpringBoot y Nodejs con el framework de Express, pues son las dos vistas en la carrera. En este caso, he decidido utilizar Nodejs principalmente porque, además de ser un entorno conocido, tengo más experiencia usándolo y me conozco mejor la arquitectura de una API realizada con Nodejs y Express. Aun así, estudié la otra alternativa para ver qué aspectos me ofrecía el uso de Springboot frente a Nodejs. Una vez realizado el estudio, concluyó con que Nodejs es más liviano que Springboot, tiene una utilización de memoria baja y es ideal para tareas de Entrada/Salida, lo que es perfecto para el propósito de este trabajo.

### ***Base de Datos***

La primera decisión que es necesaria tomar respecto a la base de datos, es el tipo de esta. En primera instancia, se decidió hacer uso de una base de datos relacional, frente a las no relacionales, debido a la robustez que una base de datos relacional puede llegar a presentar y a lo definidas que pueden estar las relaciones entre las entidades del sistema. Una vez decidido el tipo de base de datos, se procedió a determinar qué base de datos relacional usar, la cual fue PostgreSQL. La motivación principal de esta elección, frente a MySQL, viene por el aprendizaje de un Sistema Gestor de Bases de Datos (SGBD) que no hubiera utilizado anteriormente.

Otra decisión necesaria es si se va a usar un Mapeador de Objetos Relacional (ORM). Desde un primer momento estaba contemplado su uso, pues las facilidades y las ventajas que proporciona son inmensas. Investigando qué tecnologías había disponibles, me encontré con dos disponibles para Nodejs: Prisma y Sequelize. He elegido la primera, Prisma, principalmente porque es un nuevo tipo de ORM que resuelve alguno de los problemas que tienen otros, como por ejemplo la falta de seguridad de tipos, y permite tratar con la base de datos mediante un modelo definido de manera declarativa. Además, todo cambio realizado en dicho modelo se verá reflejado en la base de datos, por lo que realizar cualquier cambio en esta será realmente sencillo, lo que disminuirá considerablemente el tiempo de desarrollo.

### ***Framework de Javascript para la aplicación web***

Para el caso de la aplicación web, había tres claros candidatos para construir el sistema, y todo serían totalmente válidos. En este caso no he tomado la vía fácil de utilizar uno con el que ya estuviera familiarizado (React), y he optado por estudiar y aprender uno nuevo, pues esto supone adquirir un conocimiento extra que me puede venir bien en mi futuro laboral. Una vez descartado React, me quedaba decidir entre Angular y Vue, por lo que me puse a investigar ambas opciones, viendo el funcionamiento y arquitectura de estas dos. Finalmente opté por el uso de Angular, puesto que está asegurado el mantenimiento del framework por parte de Google, además que se centra en el uso de



los patrones de diseño inyección de dependencias y el observer, lo cual me llama la atención, debido a que hacen del framework una opción muy fácil de aprender.

### ***Sincronización para los cuestionarios en línea***

Anteriormente se comentó que se va a hacer uso del protocolo de comunicación WebSockets, sin embargo, no se van a utilizar de manera nativa, si no que se utilizará una librería llamada **Socket.io**. Dicha librería no es una implementación del protocolo WebSockets, si no que está construida sobre este, además de implementar una serie de características extra que, de no utilizar esta librería, eventualmente necesitaríamos implementar la mayoría de estas, por lo que puede reducir considerablemente el tiempo de desarrollo [7]. Desde el primer momento sabía que iba hacer uso de esta librería, pues no hay muchas más opciones y esta es la más popular.

### ***Despliegue***

Por último, queda la decisión de cómo se va a llevar a cabo el despliegue de la aplicación. El despliegue se realizará usando un servidor con el sistema operativo Windows Server 2019 proporcionado por el tutor del trabajo. Por otro lado, para lanzar la aplicación web, se utilizará un servidor Apache configurado con un certificado ssl autofirmado para poder servir la página vía https.



# Capítulo 4 ESTUDIO DE VIABILIDAD DEL SISTEMA

**FASE DE DESARROLLO**

**EVS**



## 4.1 EVS 4, 5 Y 6: ESTUDIO Y VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN Y SELECCIÓN DE ALTERNATIVA FINAL

---

En este apartado se verán varias alternativas que han sido estudiadas para afrontar la solución al problema planteado. Como se ha dicho anteriormente, será necesaria una aplicación y un servidor, por lo que se describirán a continuación varias alternativas tenidas en cuenta y, finalmente, el enfoque seleccionado.

### 4.1.1 Aplicación (Cliente)

Para realizar la parte del cliente, existen varias aproximaciones, las cuales se describirán a continuación y se indicarán también, las ventajas y desventajas de cada una.

#### 4.1.1.1 Aplicación de escritorio

Este tipo de aplicaciones se ejecutan en sistemas operativos de escritorio como Windows, macOS o Linux. Pueden estar escritas en una variedad de lenguajes de programación, como C++, Java o Python, y ofrecen una experiencia más robusta y potente en comparación con las aplicaciones web o móviles que veremos más adelante.

##### 4.1.1.1.1 Ventajas

- **Rendimiento:** Las aplicaciones de escritorio suelen tener un rendimiento más rápido y eficiente en comparación con las aplicaciones web o móviles, debido a que se ejecutan directamente en el sistema operativo del dispositivo.
- **Uso sin conexión:** A diferencia de las aplicaciones web, muchas aplicaciones de escritorio pueden funcionar sin conexión a internet

##### 4.1.1.1.2 Desventajas

- **Instalación y actualizaciones:** Las aplicaciones de escritorio requieren ser instaladas en el dispositivo del usuario, lo que puede ser un proceso más complicado en comparación con simplemente acceder a una aplicación web en un navegador.
- **Plataforma y compatibilidad:** Las aplicaciones de escritorio están vinculadas a un sistema operativo específico. Esto significa que una aplicación desarrollada para Windows no funcionará en macOS o Linux sin modificaciones adicionales. Esto puede limitar el alcance de la aplicación y requerir más esfuerzo de desarrollo para admitir múltiples plataformas.



#### 4.1.1.2 Aplicación móvil nativa

Estas aplicaciones se desarrollan específicamente para una plataforma móvil particular, como iOS o Android. Se escriben en los lenguajes de programación nativos de esas plataformas, como Objective-C o Swift para iOS, y Java o Kotlin para Android. Las aplicaciones móviles nativas suelen ofrecer un alto rendimiento y acceso a las características y funcionalidades específicas de cada plataforma.

##### 4.1.1.2.1 Ventajas

- **Rendimiento óptimo:** Las aplicaciones móviles nativas se desarrollan específicamente para una plataforma móvil particular, como iOS o Android. Esto permite aprovechar al máximo las capacidades y recursos del dispositivo, lo que resulta en un rendimiento óptimo y una experiencia fluida para el usuario.
- **Disponibilidad en tiendas de aplicaciones:** Las aplicaciones móviles nativas se distribuyen a través de las tiendas de aplicaciones oficiales, como App Store de iOS y Google Play Store de Android. Esto brinda a los desarrolladores la oportunidad de llegar a una gran base de usuarios y aprovechar los mecanismos de descubrimiento y promoción integrados en las tiendas de aplicaciones.

##### 4.1.1.2.2 Desventajas

- **Costos y tiempo de desarrollo:** El desarrollo de aplicaciones móviles nativas puede llevar más tiempo y requerir más recursos en comparación con otras alternativas. Es necesario desarrollar y mantener dos versiones separadas de la aplicación, una para iOS y otra para Android, lo que implica un mayor costo y esfuerzo.
- **Limitaciones de plataforma:** Las aplicaciones móviles nativas están vinculadas a una plataforma específica. Esto significa que una aplicación desarrollada para iOS no funcionará en dispositivos Android y viceversa. Para llegar a usuarios en múltiples plataformas, se necesita desarrollar y mantener versiones separadas de la aplicación.

#### 4.1.1.3 Aplicación móvil híbrida

Las aplicaciones móviles híbridas se desarrollan utilizando tecnologías web como HTML, CSS y JavaScript, y luego se envuelven en un contenedor nativo que permite ejecutarlas como una aplicación nativa en el dispositivo móvil. Algunos ejemplos populares de frameworks para el desarrollo de aplicaciones móviles híbridas son Ionic, React Native y Xamarin.

##### 4.1.1.3.1 Ventajas

- **Desarrollo más rápido:** Las aplicaciones móviles híbridas se desarrollan utilizando tecnologías web como HTML, CSS y JavaScript, lo que permite utilizar un solo código base para múltiples plataformas. Esto puede reducir significativamente el tiempo y el esfuerzo de desarrollo en comparación con el desarrollo de aplicaciones nativas separadas para iOS y Android.



- **Menor costo de desarrollo:** Al tener un único código base compartido entre plataformas, las aplicaciones móviles híbridas pueden ser más rentables en términos de desarrollo. No es necesario contratar equipos de desarrollo separados para cada plataforma, lo que puede ayudar a reducir los costos asociados con el desarrollo de aplicaciones móviles.

#### 4.1.1.3.2 Desventajas

- **Rendimiento limitado:** Las aplicaciones móviles híbridas generalmente no tienen el mismo rendimiento que las aplicaciones nativas, ya que se ejecutan en un contenedor nativo que a menudo es más lento que la ejecución directa en el sistema operativo. Las aplicaciones híbridas pueden experimentar tiempos de carga más lentos y una menor capacidad de respuesta en comparación con las aplicaciones nativas.
- **Dependencia de frameworks y herramientas:** Las aplicaciones móviles híbridas dependen de frameworks y herramientas específicas, como Ionic, React Native o Xamarin. Esto significa que los desarrolladores pueden estar limitados por las características y limitaciones de estas herramientas y pueden requerir una curva de aprendizaje adicional.

#### 4.1.1.4 Aplicación web

Las aplicaciones web son aquellas que se ejecutan en un navegador web y están escritas en lenguajes como HTML, CSS y JavaScript. Permiten a los usuarios acceder a la aplicación a través de un navegador sin necesidad de descargar o instalar software adicional.

##### 4.1.1.4.1 Ventajas

- **Accesibilidad:** Las aplicaciones web son accesibles a través de un navegador web en diferentes dispositivos y sistemas operativos, lo que las hace ampliamente disponibles para usuarios de todo tipo, sin necesidad de descargar o instalar software adicional.
- **Actualizaciones sencillas:** Al ser una aplicación centralizada en un servidor, las actualizaciones y mejoras se pueden implementar de manera rápida y eficiente en el lado del servidor. Los usuarios simplemente necesitan actualizar su navegador para obtener las últimas versiones de la aplicación.

##### 4.1.1.4.2 Desventajas

- **Dependencia de la conexión a internet:** Las aplicaciones web requieren una conexión a internet para funcionar correctamente. Si los usuarios no tienen una conexión estable o están en un área sin acceso a internet, no podrán acceder a la aplicación o experimentarán un rendimiento deficiente.
- **Rendimiento limitado:** Las aplicaciones web pueden tener un rendimiento ligeramente inferior en comparación con las aplicaciones nativas, especialmente en términos de tiempo de carga y respuesta interactiva. Esto se debe a la necesidad de comunicarse constantemente con el servidor y procesar la información en el navegador.



#### 4.1.1.5 Opción escogida

Después de analizar todas las opciones, la que más encaja con lo propuesto, es la aplicación web. Esto se debe a varios factores decisivos:

- El autor está más familiarizado con la creación de aplicaciones web que con el resto de las opciones.
- Con una aplicación web, se logra tener soporte para todos los sistemas existentes, ya sean sistemas operativos de escritorio, como sistemas operativos de dispositivos móviles, que hacen uso del navegador para acceder a ella.

Otra opción a tener en cuenta serían las aplicaciones móviles híbridas, que pueden hacer uso de código de la aplicación web para generar una aplicación móvil para Android como para iOS. Finalmente se descartó esta opción, puesto que agregaba demasiada complejidad al proyecto.

#### 4.1.2 API Rest (Servidor)

A diferencia de la parte del cliente, para la parte del servidor es necesaria una API. Hoy en día se utiliza la arquitectura REST para construirlas, por lo que no hay más opciones para realizarla. Las únicas opciones que pueden ser estudiadas son las tecnologías con las cuales construir la API. Estas son:

- SpringBoot con Java.
- Nodejs con Javascript o Typescript.

Finalmente, se ha decidido utilizar nodejs por la facilidad que brinda para construir APIs Rest junto con el framework de Express. La facilidad viene dada también por la experiencia que he obtenido utilizando Nodejs en otros proyectos, por lo que es un candidato perfecto para este proyecto.



# Capítulo 5 PLANIFICACIÓN Y GESTIÓN DEL TFG

**FASE DE DESARROLLO**

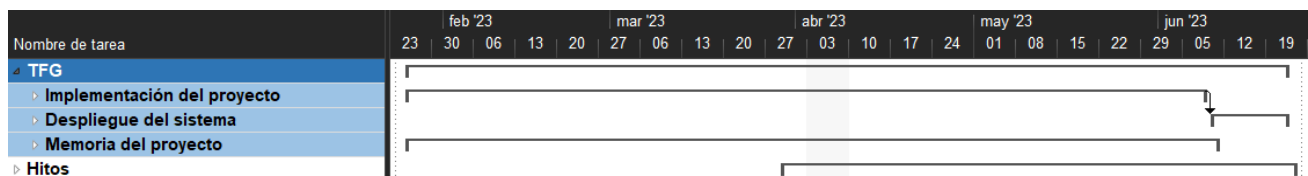
## 5.1 PLANIFICACIÓN DEL PROYECTO

### 5.1.1 Planificación Inicial

El proyecto está estimado para empezar el día 26 de enero de 2023 y finalizarse el 25 de junio de 2023, planeando una totalidad de 115 días. Para llevar a cabo la planificación, se hace uso de la aplicación Microsoft Project. El calendario que se ha utilizado para realizar la planificación es:

- Días de trabajo:
  - Lunes de 17:00 a 19:00.
  - Jueves de 17:00 a 19:00.
  - Viernes de 15:00 a 16:00.
  - Sábado de 10:00 a 12:00 y de 16:00 a 17:00.
  - Domingo de 10:00 a 12:00 y de 16:00 a 18:00.
- Excepciones:
  - Un descanso del 25/03/2023 al 26/03/2023.
  - Semana Santa del 01/04/2023 al 09/04/2023.

A continuación, se muestran el diagrama de Gantt que resumen la planificación temporal. Como se puede apreciar, el trabajo consta de tres tareas principales, la Memoria y la Implementación del proyecto y el Despliegue del sistema:



*Ilustración 1 Diagrama de Gantt*

Las tareas en detalle se pueden ver en la siguiente tabla:

EDT	Nombre de tarea	Duración	Comienzo	Fin
<b>1</b>	<b>TFG</b>	<b>113,5 días</b>	<b>jue 26/01/23</b>	<b>jue 22/06/23</b>
<b>1.1</b>	<b>Implementación del proyecto</b>	<b>102,5 días</b>	<b>jue 26/01/23</b>	<b>vie 09/06/23</b>
1.1.1	Creación de la base de datos	3 hrs	jue 26/01/23	vie 27/01/23
<b>1.1.2</b>	<b>Backend</b>	<b>45 días</b>	<b>sáb 28/01/23</b>	<b>dom 19/03/23</b>
<b>1.1.2.1</b>	<b>Creación del proyecto y de la arquitectura</b>	<b>1,5 días</b>	<b>sáb 28/01/23</b>	<b>sáb 28/01/23</b>
1.1.2.1.1	Crear el proyecto y estructurarlo	1 hr	sáb 28/01/23	sáb 28/01/23
1.1.2.1.2	Integrar con la base de datos	2 hrs	sáb 28/01/23	sáb 28/01/23
<b>1.1.2.2</b>	<b>Creación de las funcionalidades</b>	<b>31 días</b>	<b>dom 29/01/23</b>	<b>dom 05/03/23</b>
1.1.2.2.1	CRUD de usuarios	3 hrs	dom 29/01/23	dom 29/01/23
1.1.2.2.2	Login de usuarios	6 hrs	dom 29/01/23	vie 03/02/23



1.1.2.2.3	CRUD de cursos	3 hrs	sáb 04/02/23	sáb 04/02/23
1.1.2.2.4	CRUD de preguntas	15 hrs	dom 05/02/23	dom 12/02/23
1.1.2.2.5	CRUD de cuestionarios	10 hrs	dom 12/02/23	dom 19/02/23
1.1.2.2.6	CRUD de juegos	5 hrs	dom 19/02/23	lun 20/02/23
1.1.2.2.7	CRUD de resultados	5 hrs	jue 23/02/23	sáb 25/02/23
1.1.2.2.8	Creación del servicio de sockets	15 hrs	sáb 25/02/23	dom 05/03/23
1.1.2.3	Pruebas unitarias	25 hrs	dom 05/03/23	dom 19/03/23
<b>1.1.3</b>	<b>Frontend</b>	<b>54 días</b>	<b>dom 19/03/23</b>	<b>lun 05/06/23</b>
1.1.3.1	Creación del proyecto	2 hrs	dom 19/03/23	lun 20/03/23
<b>1.1.3.2</b>	<b>Creación de las funcionalidades</b>	<b>33 días</b>	<b>lun 20/03/23</b>	<b>dom 14/05/23</b>
1.1.3.2.1	Funcionalidad de login	3 hrs	lun 20/03/23	jue 23/03/23
<b>1.1.3.2.2</b>	<b>Funcionalidades del administrador</b>	<b>3 días</b>	<b>vie 24/03/23</b>	<b>vie 31/03/23</b>
1.1.3.2.2.1	Gestión de usuarios	4 hrs	vie 24/03/23	jue 30/03/23
1.1.3.2.2.2	Gestión de cursos	2 hrs	jue 30/03/23	vie 31/03/23
<b>1.1.3.2.3</b>	<b>Funcionalidades del profesor</b>	<b>24,5 días</b>	<b>lun 10/04/23</b>	<b>lun 08/05/23</b>
1.1.3.2.3.1	Gestión de cursos	8 hrs	lun 10/04/23	sáb 15/04/23
1.1.3.2.3.2	Gestión de preguntas	6 hrs	dom 16/04/23	lun 17/04/23
1.1.3.2.3.3	Gestión de cuestionarios	5 hrs	jue 20/04/23	sáb 22/04/23
1.1.3.2.3.4	Lanzamiento y juego de cuestionarios	30 hrs	sáb 22/04/23	lun 08/05/23
<b>1.1.3.2.4</b>	<b>Funcionalidades del alumno</b>	<b>4 días</b>	<b>lun 08/05/23</b>	<b>dom 14/05/23</b>
1.1.3.2.4.1	Jugar cuestionarios	30 hrst	lun 08/05/23	mié 10/05/23
1.1.3.2.4.2	Consultar resultados	6 hrs	jue 11/05/23	sáb 13/05/23
1.1.3.2.4.3	Cambio de contraseña	1 hr	dom 14/05/23	dom 14/05/23
1.1.3.3	Pruebas unitarias	30 hrs	dom 14/05/23	jue 01/06/23
<b>1.1.3.4</b>	<b>Pruebas de usabilidad</b>	<b>4 días</b>	<b>jue 01/06/23</b>	<b>dom 04/06/23</b>
1.1.3.4.1	Realizar las pruebas	2 hrs	jue 01/06/23	vie 02/06/23
1.1.3.4.2	Analizar los resultados	1 hr	sáb 03/06/23	sáb 03/06/23
1.1.3.4.3	Corregir observaciones	5 hrs	sáb 03/06/23	dom 04/06/23
1.1.3.5	Pruebas de accesibilidad	2 hrs	dom 04/06/23	lun 05/06/23
1.1.4	Integración continua	4 hrs	lun 05/06/23	vie 09/06/23
<b>1.2</b>	<b>Despliegue del sistema</b>	<b>11 días</b>	<b>sáb 10/06/23</b>	<b>jue 22/06/23</b>
1.2.1	Configuración del servidor	10 hrs	sáb 10/06/23	jue 15/06/23
1.2.2	Despliegue de la API Rest	1 hr	jue 15/06/23	jue 15/06/23
1.2.3	Despliegue de la aplicación web	1 hr	vie 16/06/23	vie 16/06/23
1.2.4	Pruebas de carga	10 hrs	sáb 17/06/23	jue 22/06/23
<b>1.3</b>	<b>Memoria del proyecto</b>	<b>105 días</b>	<b>jue 26/01/23</b>	<b>dom 11/06/23</b>
<b>1.3.1</b>	<b>Inicio del proyecto</b>	<b>1 día</b>	<b>jue 26/01/23</b>	<b>jue 26/01/23</b>
1.3.1.1	Resumen	1 hr	jue 26/01/23	jue 26/01/23
1.3.1.2	Palabras clave	1 hr	jue 26/01/23	jue 26/01/23
<b>1.3.2</b>	<b>Planificación del Sistema de Información</b>	<b>9,75 días</b>	<b>vie 27/01/23</b>	<b>dom 05/02/23</b>
<b>1.3.2.1</b>	<b>Inicio del Plan de Sistema de Información</b>	<b>2 días</b>	<b>vie 27/01/23</b>	<b>sáb 28/01/23</b>
1.3.2.1.1	Análisis de la necesidad del PSI	2 hrs	vie 27/01/23	sáb 28/01/23
1.3.2.1.2	Identificación del alcance del PSI	2 hrs	sáb 28/01/23	sáb 28/01/23



1.3.2.2	Definición y Organización del PSI	4 hrs	dom 29/01/23	dom 29/01/23
1.3.2.3	Especificación del ámbito y alcance	2 hrs	lun 30/01/23	lun 30/01/23
1.3.2.4	Estudio de la información relevante	2 hrs	jue 02/02/23	jue 02/02/23
1.3.2.5	Selección y análisis de antecedentes	1,5 hrs	vie 03/02/23	sáb 04/02/23
1.3.2.6	Conceptos teóricos	4 hrs	sáb 04/02/23	dom 05/02/23
1.3.2.7	Sistemas similares	2 hrs	dom 05/02/23	dom 05/02/23
<b>1.3.3</b>	<b>Definición de la arquitectura tecnológica</b>	<b>3,5 días</b>	<b>dom 05/02/23</b>	<b>sáb 11/02/23</b>
1.3.3.1	Identificación de las necesidades de infraestructura tecnológica	3 hrs	dom 05/02/23	jue 09/02/23
1.3.3.2	Selección de la arquitectura tecnológica	4 hrs	jue 09/02/23	sáb 11/02/23
1.3.4	Estudio de viabilidad del sistema	2 hrs	sáb 11/02/23	dom 12/02/23
<b>1.3.5</b>	<b>Planificación y gestión del TFG</b>	<b>8,5 días</b>	<b>dom 12/02/23</b>	<b>lun 20/02/23</b>
1.3.5.1	Planificación del proyecto	10 hrs	dom 12/02/23	sáb 18/02/23
1.3.5.2	Ejecución del proyecto	4 hrs	sáb 18/02/23	dom 19/02/23
1.3.5.3	Cierre del proyecto	3 hrs	dom 19/02/23	lun 20/02/23
<b>1.3.6</b>	<b>Análisis del sistema de información</b>	<b>28 días</b>	<b>lun 20/02/23</b>	<b>vie 31/03/23</b>
1.3.6.1	Definición del sistema	3 hrs	lun 20/02/23	vie 24/02/23
<b>1.3.6.2</b>	<b>Establecimiento de requisitos</b>	<b>8,5 días</b>	<b>vie 24/02/23</b>	<b>dom 05/03/23</b>
1.3.6.2.1	Obtención de los requisitos del sistema	4 hrs	vie 24/02/23	dom 26/02/23
1.3.6.2.2	Identificación de los actores del sistema	1 hr	dom 26/02/23	dom 26/02/23
1.3.6.2.3	Especificación de los casos de uso	12 hrs	dom 26/02/23	dom 05/03/23
<b>1.3.6.3</b>	<b>Identificación de subsistemas de análisis</b>	<b>1,75 días</b>	<b>dom 05/03/23</b>	<b>lun 06/03/23</b>
1.3.6.3.1	Descripción de los subsistemas	2 hrs	dom 05/03/23	dom 05/03/23
1.3.6.3.2	Descripción de las interfaces entre subsistemas	1,5 hrs	dom 05/03/23	lun 06/03/23
1.3.6.4	Análisis de los casos de uso	20 hrs	lun 06/03/23	dom 19/03/23
<b>1.3.6.5</b>	<b>Análisis de clases</b>	<b>4 días</b>	<b>dom 19/03/23</b>	<b>vie 24/03/23</b>
1.3.6.5.1	Diagrama de clases	1 hr	dom 19/03/23	dom 19/03/23
1.3.6.5.2	Descripción de las clases	7 hrs	dom 19/03/23	vie 24/03/23
<b>1.3.6.6</b>	<b>Definición de interfaces de usuario</b>	<b>1,75 días</b>	<b>lun 27/03/23</b>	<b>jue 30/03/23</b>
1.3.6.6.1	Descripción de la interfaz	1 hr	lun 27/03/23	lun 27/03/23
1.3.6.6.2	Descripción del comportamiento de la interfaz	2 hrs	lun 27/03/23	jue 30/03/23
1.3.6.6.3	Diagrama de navegabilidad	0,5 hrs	jue 30/03/23	jue 30/03/23
1.3.6.7	Especificación del plan de pruebas	1 hr	jue 30/03/23	vie 31/03/23
<b>1.3.7</b>	<b>Diseño del sistema de información</b>	<b>24,75 días</b>	<b>vie 31/03/23</b>	<b>lun 08/05/23</b>
1.3.7.1	Diseño de casos de uso reales	2 hrs	vie 31/03/23	lun 10/04/23
1.3.7.2	Diseño de clases	4 hrs	lun 10/04/23	sáb 15/04/23
<b>1.3.7.3</b>	<b>Diseño de la Arquitectura de módulos del sistema</b>	<b>6,5 días</b>	<b>sáb 15/04/23</b>	<b>sáb 22/04/23</b>
1.3.7.3.1	Diagrama de paquetes	3 hrs	sáb 15/04/23	dom 16/04/23
1.3.7.3.2	Diagrama de despliegue	3 hrs	dom 16/04/23	dom 16/04/23
1.3.7.3.3	Diseño de módulos del sistema	2 hrs	dom 16/04/23	lun 17/04/23
1.3.7.3.4	Diseño de comunicaciones entre módulos	2 hrs	lun 17/04/23	jue 20/04/23
1.3.7.3.5	Revisión de la interfaz de usuario	3 hrs	jue 20/04/23	sáb 22/04/23
<b>1.3.7.4</b>	<b>Diseño físico de datos</b>	<b>3,5 días</b>	<b>sáb 22/04/23</b>	<b>lun 24/04/23</b>
1.3.7.4.1	Descripción del SGBD usado	1 hr	sáb 22/04/23	sáb 22/04/23





1.3.7.4.2	Integración del SGBD	2 hrs	sáb 22/04/23	dom 23/04/23
1.3.7.4.3	E-R del modelo de dominio	4 hrs	dom 23/04/23	lun 24/04/23
<b>1.3.7.5</b>	<b>Especificación técnica del plan de pruebas</b>	<b>11,75 días</b>	<b>lun 24/04/23</b>	<b>lun 08/05/23</b>
1.3.7.5.1	Pruebas unitarias	6 hrs	lun 24/04/23	sáb 29/04/23
1.3.7.5.2	Pruebas de integración y de sistema	6 hrs	sáb 29/04/23	lun 01/05/23
<b>1.3.7.5.3</b>	<b>Pruebas de usabilidad</b>	<b>1,25 días</b>	<b>lun 01/05/23</b>	<b>jue 04/05/23</b>
1.3.7.5.3.1	Diseño del cuestionario de evaluación	1,5 hrs	lun 01/05/23	jue 04/05/23
1.3.7.5.3.2	Identificación de actividades	1 hr	jue 04/05/23	jue 04/05/23
1.3.7.5.4	Pruebas de accesibilidad	5 hrs	vie 05/05/23	dom 07/05/23
1.3.7.5.5	Pruebas de rendimiento	10 hrs	jue 04/05/23	lun 08/05/23
<b>1.3.8</b>	<b>Construcción del sistema de información</b>	<b>22,5 días</b>	<b>lun 08/05/23</b>	<b>dom 04/06/23</b>
<b>1.3.8.1</b>	<b>Preparación del entorno de generación y construcción</b>	<b>1,5 días</b>	<b>lun 08/05/23</b>	<b>jue 11/05/23</b>
1.3.8.1.1	Estándares y normas seguidos	1 hr	lun 08/05/23	lun 08/05/23
1.3.8.1.2	Lenguajes de programación	1 hr	jue 11/05/23	jue 11/05/23
1.3.8.1.3	Herramientas y programas usados para el desarrollo	1 hr	jue 11/05/23	jue 11/05/23
1.3.8.2	Generación del código de los componentes y procedimientos	5 hrs	vie 12/05/23	dom 14/05/23
1.3.8.3	Ejecución de las pruebas unitarias	10 hrs	dom 14/05/23	sáb 20/05/23
1.3.8.4	Ejecución de las pruebas de Integración	10 hrs	sáb 20/05/23	vie 26/05/23
<b>1.3.8.5</b>	<b>Ejecución de las pruebas del sistema</b>	<b>4,5 días</b>	<b>sáb 27/05/23</b>	<b>lun 29/05/23</b>
1.3.8.5.1	Pruebas de usabilidad	6 hrs	sáb 27/05/23	dom 28/05/23
1.3.8.5.2	Pruebas de accesibilidad	3 hrs	dom 28/05/23	lun 29/05/23
<b>1.3.8.6</b>	<b>Elaboración de los manuales de usuario</b>	<b>4 días</b>	<b>jue 01/06/23</b>	<b>dom 04/06/23</b>
1.3.8.6.1	Manual de instalación	2 hrs	jue 01/06/23	jue 01/06/23
1.3.8.6.2	Manual de ejecución	2 hrs	vie 02/06/23	sáb 03/06/23
1.3.8.6.3	Manual de usuario	2 hrs	sáb 03/06/23	sáb 03/06/23
1.3.8.6.4	Manual del programador	2 hrs	dom 04/06/23	dom 04/06/23
<b>1.3.9</b>	<b>Conclusiones y ampliaciones</b>	<b>1 día</b>	<b>dom 04/06/23</b>	<b>dom 04/06/23</b>
1.3.9.1	Conclusiones	1 hr	dom 04/06/23	dom 04/06/23
1.3.9.2	Ampliaciones	1 hr	dom 04/06/23	dom 04/06/23
<b>1.3.10</b>	<b>Apéndices</b>	<b>5 días</b>	<b>lun 05/06/23</b>	<b>dom 11/06/23</b>
1.3.10.1	Plan de gestión de riesgos	6 hrs	lun 05/06/23	sáb 10/06/23
1.3.10.2	Referencias bibliográficas	4 hrs	sáb 10/06/23	dom 11/06/23
<b>2</b>	<b>Hitos</b>	<b>65,5 días</b>	<b>jue 30/03/23</b>	<b>sáb 24/06/23</b>
2.1	Versión estable de la API Rest	0 días	jue 30/03/23	jue 30/03/23
2.2	Versión estable de la aplicación web	0 días	jue 08/06/23	jue 08/06/23
2.3	Sistema desplegado en el servidor	0 días	sáb 24/06/23	sáb 24/06/23

**Tabla 1 Planificación temporal del proyecto**

## 5.1.2 Riesgos

### 5.1.2.1 Plan de Gestión de Riesgos

El plan de gestión de riesgos se encuentra en el anexo **Plan de gestión de riesgos**.

## 5.1.3 Presupuesto Inicial

### 5.1.3.1 Presupuesto de Costes

En este apartado se realiza el presupuesto de costes inicial a partir de las tareas detalladas en el primer apartado de esta sección, basado en lo aprendido en la Guía de aprendizaje de la asignatura de Dirección y Planificación de Proyectos Informáticos [8].

Para justificar los costes de precio/hora, es necesario definir el modelo de la empresa y, como cabe esperar, solo existe un trabajador, el autor del proyecto, cuyo precio/hora son unos 15<sup>1</sup> €.

Concretamente, el presupuesto se divide en tres partidas: Desarrollo del sistema, Memoria, Costes indirectos y Costes de los medios de producción. Con este presupuesto, solo se busca estimar el coste total del proyecto, sin contar ningún margen de beneficio, puesto que no es para un cliente real.

La primera partida es la del Desarrollo del sistema, que muestra los costes de la implementación del sistema, es decir, construir la API Rest y la aplicación web, y el despliegue del sistema.

Partida 1: Desarrollo del sistema								
n1	n2	n3	Nombre de tarea	Horas	Precio/hora	Subtotal(n3)	Subtotal(n2)	Subtotal(n1)
<b>1</b>			<b>Implementación del proyecto</b>	<b>205</b>				<b>2.896 €</b>
	1		Creación de la base de datos	3	15 €		<b>45 €</b>	
	<b>2</b>		<b>Backend</b>	<b>90</b>			<b>1.171 €</b>	
		1	Creación del proyecto y de la arquitectura	3	15 €	52 €		
		2	Creación de las funcionalidades	62	15 €	744 €		
		3	Pruebas unitarias	25	15 €	375 €		
	<b>3</b>		<b>Frontend</b>	<b>108</b>			<b>1.620 €</b>	
		1	Creación del proyecto	2	15 €	30 €		
		2	Creación de las funcionalidades	66	15 €	990 €		
		3	Pruebas unitarias	30	15 €	450 €		
		4	Pruebas de usabilidad	8	15 €	120 €		
		5	Pruebas de accesibilidad	2	15 €	30 €		
	4		Integración continua	4	15 €		<b>60 €</b>	
<b>2</b>			<b>Despliegue del sistema</b>	<b>22</b>				<b>330 €</b>
	1		Configuración del servidor	10	15 €		<b>150 €</b>	
	2		Despliegue de la API Rest	1	15 €		<b>15</b>	
	3		Despliegue de la aplicación web	1	15 €		<b>15</b>	
	4		Pruebas de carga	10	15 €		<b>150</b>	
<b>TOTAL</b>								<b>3.226 €</b>

**Tabla 2 Partida 1 Desarrollo del sistema**

<sup>1</sup> El precio/hora fue sacado de una página web que indica cuánto gana de media un programador en España [22]

Partida 2: Memoria						
n1	n2	Nombre de tarea	Horas	Precio/Hora	Subtotal(n2)	Subtotal(n1)
<b>1</b>		<b>Memoria del proyecto</b>	210			<b>3150 €</b>
	1	Inicio del proyecto	2	15 €	30 €	
	2	Planificación del Sistema de Información	19,5	15 €	292,5 €	
	3	Definición de la arquitectura tecnológica	7	15 €	105 €	
	4	Estudio de viabilidad del sistema	2	15 €	30 €	
	5	Planificación y gestión del TFG	17	15 €	255 €	
	6	Análisis del sistema de información	56	15 €	840 €	
	7	Diseño del sistema de información	49,5	15 €	742,5 €	
	8	Construcción del sistema de información	45	15 €	675 €	
	9	Conclusiones y ampliaciones	2	15 €	30 €	
	10	Apéndices	10	15 €	150 €	

**Tabla 3 Partida 2 Memoria**

Existen una serie de costes correspondientes a los medios de producción, ya sean físicos o licencias de software utilizado para el proyecto. A continuación, se muestran estos costes:

Costes de los medios de producción				
Descripción	Unidades	Precio	Coste total	Tipo
Ordenador para el desarrollo	1	1.000 €	1.000 €	Amortización
Monitor HD 24'	2	200 €	400 €	Amortización
Equipo portátil para el desarrollo	1	900 €	900 €	Amortización
Tarifa internet	1	30 € / mes	150 €	Alquiler
<b>TOTAL</b>			<b>2450 €</b>	

**Tabla 4 Costes de los medios de producción**

Finalmente, hay que tener en cuenta también otros costes indirectos que afectan al proyecto. Concretamente, se trata del gasto energético de los equipos involucrados en el desarrollo del proyecto.

Costes indirectos			
Descripción	Precio	Coste total	Tipo
Consumo energético	50 € / mes	250 €	Amortización
<b>TOTAL</b>		<b>250 €</b>	

**Tabla 5 Costes indirectos**



### 5.1.3.2 Presupuesto de Cliente

Presupuesto del cliente		
Código	Partida	Coste total
1	Desarrollo del sistema	3.226 €
2	Memoria	3.150 €
3	Costes de los medios de producción	2.450 €
4	Costes indirectos	250 €
	<b>TOTAL</b>	<b>9.076 €</b>

**Tabla 6 Presupuesto del cliente**

# Capítulo 6 ANÁLISIS DEL SISTEMA DE INFORMACIÓN

**FASE DE DESARROLLO**

**ASI**



## 6.1 ASI 1: DEFINICIÓN DEL SISTEMA

---

### 6.1.1 Determinación del Alcance del Sistema

El alcance ya ha sido descrito en anteriores apartados, por lo que en este se detallarán aquellos aspectos, que por falta de tiempo o por otras limitaciones no se va a realizar. En primer lugar, la idea del proyecto era construir una aplicación web, tanto para los profesores como para los alumnos, y una aplicación móvil para que los alumnos pudieran contestar los cuestionarios, teniendo las mismas funcionalidades que la parte de alumnos de la aplicación web. Esta opción se ha descartado por falta de tiempo. Por otra parte, hay muchas funcionalidades posibles, en un sistema de este tipo, que no serán desarrolladas, puesto que no es viable hacerlas. Algún ejemplo puede ser:

- La posibilidad de realizar cuestionarios offline, similar a una encuesta.
- Realizar exámenes de tipo test con el banco de preguntas creadas por el profesor y el posterior análisis de resultados de éstos.

Por tanto, y, para resumir, el objetivo principal del sistema es la realización de cuestionarios en línea con una puesta en marcha lo más rápida posible y la obtención de resultados de los mismos.



## 6.2 ASI 2: ESTABLECIMIENTO DE REQUISITOS

---

### 6.2.1 Obtención de los Requisitos del Sistema

#### 6.2.1.1 *Requisitos funcionales*

##### 6.2.1.1.1 Gestión de Usuarios

**RGU.1** El sistema deberá mostrar al usuario administrador diferentes opciones sobre la gestión de usuarios.

**RGU.1.1** El sistema deberá permitir crear nuevos usuarios. Serán necesarios los siguientes campos:

**RG.1.1.1** Nombre (Obligatorio)

**RG.1.1.2** Rol (Obligatorio)

**RGU.1.2** Si el usuario no rellena todos los campos obligatorios (ver **RGU.1.1**) se mostrará un mensaje de error indicando el motivo.

**RGU.1.3** El sistema deberá asignar una contraseña aleatoria cuando se creen nuevos usuarios.

**RGU.1.4** El sistema enviará un correo al nuevo usuario con las credenciales de acceso.

**RGU.1.5** El sistema deberá permitir asignar el rol de profesor o alumno al usuario creado.

**RGU.1.6** El sistema deberá permitir modificar los usuarios. Podrá modificar los campos indicados en **RGU.1.1**.

**RGU.1.7** El sistema deberá permitir dar de baja a los usuarios.

**RGU.1.8** El sistema deberá permitir importar usuarios mediante un fichero de tipo csv.

**RGU.2** El sistema deberá permitir a los usuarios profesor añadir a usuarios alumnos a los cursos.

**RGU.3** El sistema deberá permitir a los usuarios alumnos modificar su contraseña.

##### 6.2.1.1.2 Gestión de cursos

**RGC.1** El sistema deberá mostrar al usuario administrador diferentes opciones sobre la gestión de cursos.

**RGC.1.1** El sistema deberá permitir crear nuevos cursos. Serán necesarios los siguientes campos:

**RGC.1.1.1** Nombre (Obligatorio)

**RGC.1.1.2** Descripción (Obligatorio)



### **RGC.1.1.3 Imagen (Opcional)**

**RGC.1.2** Si no se pasan las validaciones de todos los campos (ver **RGC.1.1**) el sistema mostrará un mensaje de error indicando el motivo.

**RGC.1.3** El sistema deberá permitir modificar los cursos. Podrá modificar los campos indicados en RGC.1.1

**RGC.1.4** El sistema deberá permitir eliminar los cursos.

**RGC.1.5** El sistema deberá permitir importar cursos mediante un fichero de tipo csv.

**RGC.2** El sistema deberá mostrar al usuario profesor diferentes opciones sobre la gestión de cursos.

**RGC.2.1** El sistema deberá permitir asignar cuestionarios creados por el usuario profesor, como se expone en **RGC.1.1**.

**RGC.2.2** El sistema deberá permitir asignar preguntas creadas por el usuario profesor, como se expone en **RGP.1.1**.

**RGC.2.3** El sistema deberá permitir asignar usuarios alumnos al curso.

**RGC.2.4** El sistema deberá permitir eliminar las asociaciones de los cuestionarios a los cursos.

**RGC.2.5** El sistema deberá permitir eliminar las asociaciones de las preguntas a los cursos.

**RGC.2.6** El sistema deberá permitir eliminar usuarios que hayan sido asignados al curso.

#### 6.2.1.1.3 Gestión de preguntas

**RGP.1** El sistema deberá mostrar al usuario profesor diferentes opciones sobre la gestión de preguntas.

**RGP.1.1** El sistema deberá permitir la creación de preguntas. Para crearlas, el usuario deberá indicar los siguientes parámetros:

**RGP.1.1.1** Enunciado (Obligatorio)

**RGP.1.1.2** Tipo de pregunta (Obligatorio). Estas pueden ser:

**RGP.1.1.2.1** Multiopción, es decir, la pregunta tendrá cuatro respuestas posibles, todas obligatorias y solo una puede de ser correcta.

**RGP.1.1.2.2** Verdadero o falso. La pregunta tendrá dos posibles respuestas, ambas obligatorias y solo una puede ser correcta obligatoriamente.

**RGP.1.1.2.3** Respuesta corta. La pregunta tendrá hasta 4 posibles respuestas correctas, pero solo una es obligatoria.

**RGP.1.1.3** Tiempo para responder (Obligatorio).

**RGP.1.1.3.1** El tiempo deberá de ser mayor que 5.



#### **RGP.1.1.4** Imagen (Opcional).

**RGP.1.2** Si no se pasan las validaciones de todos los campos (ver **RGP.1.1**) el sistema mostrará un mensaje de error indicando el motivo.

**RGP.1.3** El sistema deberá permitir la modificación de preguntas creadas por el propio usuario. El usuario profesor podrá modificar los parámetros mencionados en **RGP.1.1** exceptuando:

**RGP.1.3.1** Tipo de pregunta.

**RGP.1.3.2** En el caso de las preguntas de tipo multi opción y verdadero o falso, la respuesta correcta.

**RGP.1.4** El sistema deberá permitir la eliminación de preguntas creadas por el propio usuario.

**RGP.1.4.1** El sistema solo podrá llevar a cabo esta acción si la pregunta no está asociada a un cuestionario.

**RGP.1.5** El sistema deberá permitir la asignación de preguntas a cursos, como está expuesto en **RGC.2.2**.

**RGP.1.6** El sistema deberá permitir eliminar la asociación de las preguntas a los cursos, como está expuesto en **RGC.2.5**.

**RGP.1.7** El sistema deberá permitir importar preguntas en un fichero con formato csv.

**RGP.1.8** El sistema deberá permitir exportar las preguntas creadas por el propio usuario en formato csv.

#### 6.2.1.1.4 Gestión de cuestionarios

**RGC.1** El sistema deberá proporcionar al usuario profesor diferentes opciones sobre la gestión de cuestionarios.

**RGC.1.1** El sistema debe permitir la creación de cuestionarios.

**RGC.1.1.1** El usuario profesor deberá añadir obligatoriamente un título al cuestionario.

**RGC.1.1.2** El usuario profesor podrá añadir tantas preguntas como desea y, al menos, debe añadir una.

**RGC.1.1.3** Las preguntas que podrá añadir son solo las creadas (**RGP.1.1**) por el mismo usuario.

**RGC.1.1.4** El usuario podrá modificar el orden en el que aparecen las preguntas dentro del cuestionario.

**RGC.1.1.5** El usuario podrá añadir, opcionalmente, una imagen al cuestionario.



**RGC.1.2** Si no se pasan las validaciones de todos los campos (ver **RGC.1.1**) el sistema deberá mostrar un mensaje de error indicando el motivo.

**RGC.1.2** El sistema debe permitir la modificación de los cuestionarios creados por el propio usuario. El usuario profesor podrá modificar los parámetros mencionados en **RGC.1.1** exceptuando:

**RGC.1.2.1** Nuevas preguntas.

**RGC.1.3** El sistema debe permitir la eliminación de los cuestionarios creados por el propio usuario.

**RGC.1.3.1** El sistema solo podrá llevar a cabo esta acción si el cuestionario no ha sido jugado en ninguna ocasión.

**RGC.1.4** El sistema deberá permitir la asignación de cuestionarios a cursos, como está expuesto en **RGC.2.1**.

**RGC.1.5** El sistema deberá permitir eliminar la asociación de los cuestionarios a los cursos, como está expuesto **RGC.2.4**

#### 6.2.1.1.5 Lanzamiento y respuesta de los cuestionarios

**RLRC.1** El sistema deberá permitir a los usuarios profesores las siguientes funcionalidades.

**RLRC.1.1** El usuario podrá crear un juego con cualquier cuestionario asignado al curso (ver **RGC.2.1**). El usuario deberá seleccionar dos parámetros:

**RLRC.1.1.1** Si el usuario alumno verá o no el enunciado de la pregunta (Obligatorio).

**RLRC.1.1.2** Tipo de puntuación (Obligatorio). Puede ser:

**RLRC.1.1.2.1** Normal, es decir, se aplica la puntuación establecida en **RLRC.2.4**.

**RLRC.1.1.2.2** Doble, es decir, la puntuación es el doble a la establecida en **RLRC.2.4**.

**RLRC.1.1.2.3** Sin puntuación.

**RLRC.1.2** Una vez lanzado el juego, el sistema mostrará al usuario profesor el listado de alumnos que se van uniendo al juego.

**RLRC.1.3** El sistema permitirá empezar el juego si se ha unido, al menos, un jugador alumno.

**RLRC.1.4** El sistema mostrará las preguntas del cuestionario y mostrará el tiempo restante, definido en **RGP.1.1.3**, hasta llegar a cero.

**RLRC.1.5** Después de mostrar la pregunta, el sistema mostrará la respuesta correcta.

**RLRC.1.6** Después de mostrar la respuesta correcta, mostrará los tres usuarios con mayor puntuación, siendo el primero el que haya obtenido mayor puntuación hasta el momento.



**RLRC.1.7** Cuando se hayan terminado las preguntas del cuestionario, el sistema permitirá al usuario profesor finalizar el juego.

**RLRC.1.7.1** Al finalizar el juego, el sistema deberá persistir las puntuaciones de todos y cada uno de los usuarios que hayan participado en el juego.

**RLRC.2** El sistema deberá permitir a los usuarios alumnos las siguientes funcionalidades respecto a los juegos:

**RLRC.2.1** El usuario alumno mostrará los juegos abiertos de los cursos a los que haya sido asignado.

**RLRC.2.2** El usuario podrá pasar a participar en los juegos mostrados en el anterior requisito.

**RLRC.2.3** El sistema permitirá seleccionar la respuesta correcta para cada pregunta, y cuando acabe el tiempo de la pregunta, el sistema mostrará si ha acertado o no.

**RLRC.2.3.1** En el caso de que la pregunta sea de tipo “Respuesta corta”, el sistema permitirá al usuario introducir la respuesta.

**RLRC.2.4** Los puntos otorgados al usuario por responder correctamente seguirán la siguiente fórmula:  $(1 - ((\text{tiempo transcurrido} / \text{tiempo total}) / 2)) * \text{PUNTOS\_BASE}$ .

**RLRC.2.4.1** Los PUNTOS\_BASE serán 1000.

**RLRC.2.4.2** Los PUNTOS\_BASE podrán ser modificados por el administrador del sistema.

**RLRC.2.5** Una vez se muestre si el usuario ha acertado o no, el sistema mostrará la puntuación actual del usuario.

**RLRC.2.6** Una vez terminadas todas las preguntas, el usuario alumno podrá salir del juego.

#### 6.2.1.1.6 Login

**RLogin.1** El sistema debe permitir la autenticación a un usuario no identificado.

**RLogin.2** El sistema no podrá ser utilizado por ningún usuario que no haya sido identificado previamente.

#### 6.2.1.1.7 Logout

**RLogout.1** El sistema debe permitir cerrar sesión a cualquier usuario identificado previamente en el sistema.

### 6.2.1.2 *Requisitos no funcionales*

#### 6.2.1.2.1 Requisitos de usuario

N/A



#### 6.2.1.2.2 Requisitos tecnológicos

**RNFTecnológicos.1** La web del sistema deberá ser compatible con los principales navegadores web (Chrome, Edge, Firefox, Safari y Opera).

#### 6.2.1.2.3 Requisitos de usabilidad

**RNFUsabilidad.1** La web del sistema deberá ser usable, esto es, la interfaz deberá de ser sencilla e intuitiva para que los usuarios no familiarizados con ella puedan usarla correctamente.

**RNFUsabilidad.2** La web del sistema seguirá un código de colores:

**RNFUsabilidad.2.1** Usará el color verde para mostrar al usuario alumno que ha respondido correctamente.

**RNFUsabilidad.2.2** Usará el color rojo para mostrar al usuario alumno que ha respondido incorrectamente.

#### 6.2.1.2.4 Requisitos de seguridad

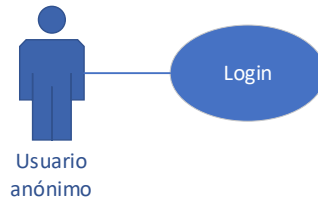
**RNFSeguridad.1** El sistema deberá encriptar todas las contraseñas mediante bcrypt.

### 6.2.2 Identificación de Actores del Sistema

Los actores que interactúan con el sistema han sido nombrados anteriormente. Estos son:

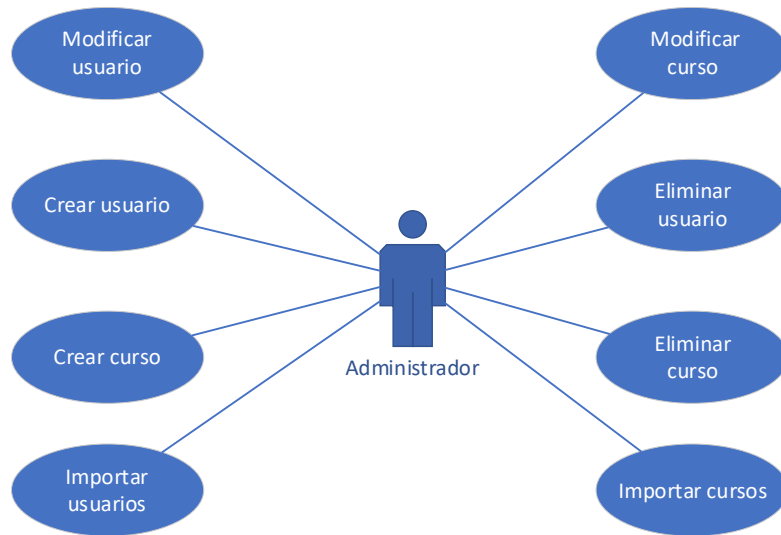
- **Usuario anónimo:** Se trata de los usuarios que no han sido identificados en el sistema, por lo que solo tienen la opción de acceder al sistema con sus credenciales.
- **Usuario administrador:** Solo existe un usuario administrador del sistema, encargado de la gestión de todos los usuarios y de los cursos.
- **Usuario profesor:** Los usuarios profesores se hacen cargo de la creación tanto de preguntas como de cuestionarios. Además, son lo que tienen la posibilidad de añadir los usuarios alumnos a los diferentes cursos y de iniciar un juego.
- **Usuario alumno:** Los usuarios alumnos son el tipo mayoritario de alumnos. Estos pueden jugar cuestionarios lanzados por el profesor y consultar los resultados de los cuestionarios jugados.

### 6.2.3 Especificación de Casos de Uso



*Ilustración 2 DIAGRAMA DE CASOS DE USO 1*

<b>Nombre del Caso de Uso</b>
Login
<b>Descripción</b>
<i>Un usuario no identificado en el sistema podrá autenticarse con sus credenciales (nombre de usuario y contraseña).</i>



*Ilustración 3 DIAGRAMA DE CASOS DE USO 2*

<b>Nombre del Caso de Uso</b>
Crear usuario
<b>Descripción</b>
El administrador podrá crear un usuario nuevo indicando el nombre y el rol del mismo.

<b>Nombre del Caso de Uso</b>
Crear curso
<b>Descripción</b>
El administrador podrá crear un curso nuevo indicando el nombre y la descripción del mismo.



<b>Nombre del Caso de Uso</b>
Modificar usuario
<b>Descripción</b>
El administrador podrá seleccionar cualquier usuario y modificar sus datos. Podrá modificar tanto su nombre como su rol.

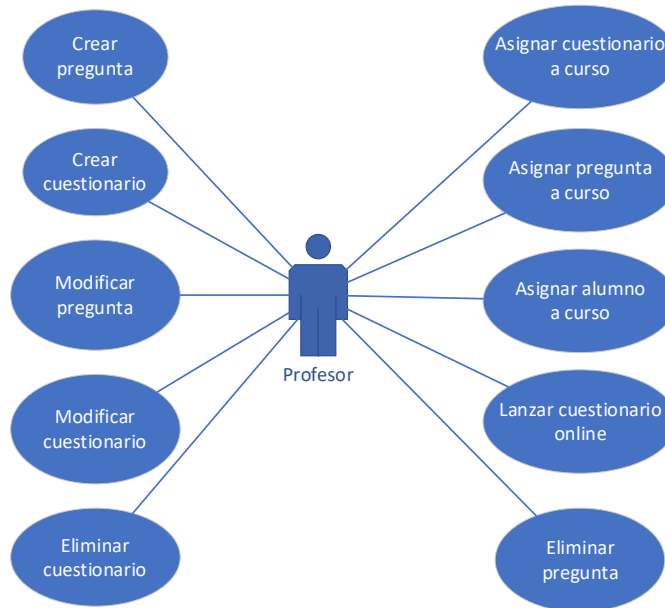
<b>Nombre del Caso de Uso</b>
Modificar curso
<b>Descripción</b>
El administrador podrá seleccionar cualquier curso y modificar sus datos. Podrá modificar tanto su nombre como su descripción.

<b>Nombre del Caso de Uso</b>
Eliminar usuario
<b>Descripción</b>
El usuario podrá eliminar cualquier usuario.

<b>Nombre del Caso de Uso</b>
Eliminar curso
<b>Descripción</b>
El usuario podrá eliminar cualquier curso.

<b>Nombre del Caso de Uso</b>
Importar usuarios
<b>Descripción</b>
El administrador podrá importar usuarios mediante un fichero en formato csv.

<b>Nombre del Caso de Uso</b>
Importar cursos
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.



**Ilustración 4 DIAGRAMA DE CASOS DE USO 3**

<b>Nombre del Caso de Uso</b>
Crear pregunta
<b>Descripción</b>
<p>El profesor podrá crear una pregunta que se podrá usar en varios cuestionarios. El usuario deberá especificar el enunciado, imagen (opcional), tiempo de respuesta y tipo de pregunta. Hay tres tipos:</p> <ul style="list-style-type: none"> <li>• Multi opción: la pregunta tendrá cuatro posibles respuestas, y al menos una deberá ser correcta.</li> <li>• Verdadero o falso: la pregunta tendrá dos posibles respuestas, y solo una será correcta.</li> <li>• Respuesta corta: la pregunta podrá tener hasta cuatro respuestas posibles, y al menos una. Todas las respuestas serán correctas.</li> </ul>

<b>Nombre del Caso de Uso</b>
Crear cuestionario
<b>Descripción</b>
<p>El profesor podrá crear cuestionarios con las preguntas que haya creado. Podrá añadir múltiples preguntas, así como eliminarlas y cambiarlas de orden. Además, deberá añadirle un título, y opcionalmente, una imagen.</p>

<b>Nombre del Caso de Uso</b>
Modificar pregunta
<b>Descripción</b>
<p>El profesor podrá seleccionar cualquier pregunta que haya creado anteriormente y modificar cualquiera de sus campos.</p>



<b>Nombre del Caso de Uso</b>
Modificar cuestionario
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.

<b>Nombre del Caso de Uso</b>
Eliminar cuestionario
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.

<b>Nombre del Caso de Uso</b>
Eliminar pregunta
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.

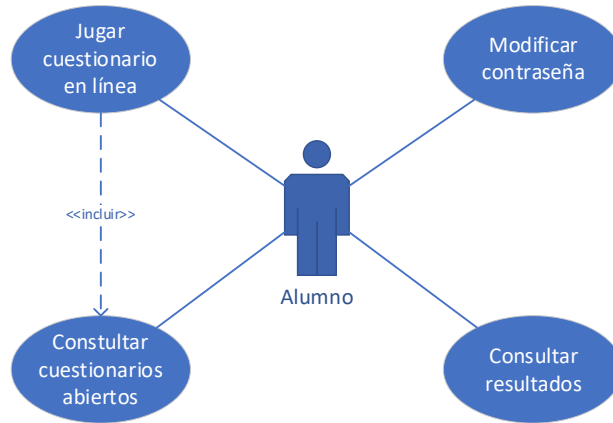
<b>Nombre del Caso de Uso</b>
Asignar cuestionario a un curso
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.

<b>Nombre del Caso de Uso</b>
Asignar pregunta a un curso
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.

<b>Nombre del Caso de Uso</b>
Asignar alumno a un curso
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.

<b>Nombre del Caso de Uso</b>
Lanzar cuestionario online
<b>Descripción</b>
El administrador podrá importar cursos mediante un fichero en formato csv.





**Ilustración 5 DIAGRAMA DE CASOS DE USO 4**

<b>Nombre del Caso de Uso</b>
Jugar cuestionario en línea
<b>Descripción</b>
El alumno podrá pasar a jugar cualquier cuestionario abierto.

<b>Nombre del Caso de Uso</b>
Modificar contraseña
<b>Descripción</b>
El alumno podrá modificar en cualquier momento su contraseña.

<b>Nombre del Caso de Uso</b>
Consultar cuestionarios abiertos
<b>Descripción</b>
El alumno podrá consultar en la pantalla principal todos los cuestionarios abiertos en ese momento.

<b>Nombre del Caso de Uso</b>
Consultar resultados
<b>Descripción</b>
El alumno podrá consultar los resultados de los cuestionarios realizados de cada uno de los cursos a los que pertenezca.



## 6.3 ASI 3: IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS

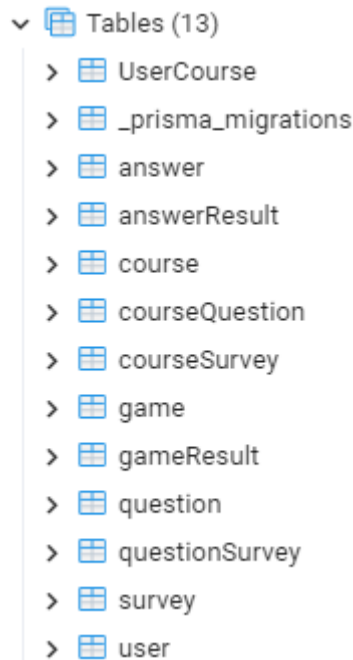
En esta sección se descompondrá el sistema en partes más pequeñas (subsistemas) para facilitar su posterior análisis. Estos son:

- Base de Datos.
- Servidor web
- Servidor de sockets
- Aplicación web

### 6.3.1 Descripción de los Subsistemas

#### 6.3.1.1 Base de Datos

La base de datos es la encargada de almacenar toda la información relevante y de proveer esta al resto de subsistemas para el correcto funcionamiento del sistema. Para el desarrollo de este subsistema se ha escogido un enfoque relacional, concretamente, el sistema de gestión de bases de datos (SGBD) es PostgreSQL. Además, también se ha utilizado el cliente PgAdmin4 para gestionar y administrar las tablas. Las tablas del sistema son las siguientes:



**Ilustración 6 TABLAS DE LA BASE DE DATOS**

La tabla `_prisma_migrations` es una autogenerada por el ORM utilizado, por lo que no se tiene en cuenta.



### 6.3.1.2 Servidor web

Este subsistema se puede dividir a su vez en otro dos, debido a que cada parte tiene una función diferenciada. Estas son:

#### 6.3.1.2.1 API Rest

Este subsistema es el intermediario entre la base de datos y la aplicación web. Su función principal es recibir peticiones de la aplicación web, procesarlas haciendo las consultas a la base de datos necesarias, y devolver la información requerida. Esto lo lleva a cabo mediante rutas, que se encargan de redirigir la petición al controlador correcto, donde se procesan las peticiones y se devuelve la información solicitada.

La API, como indica el nombre del subsistema, se ha realizado en base a REST con Nodejs y el framework Express, y para mapear las instancias del modelo se ha utilizado el ORM Prisma.

#### 6.3.1.2.2 Servidor de sockets

Este subsistema forma parte del servidor web, al igual que la API Rest, pero se ha decidido separar, puesto que tiene una función diferente a esta. Su función principal es llevar a cabo la sincronización necesaria para jugar los cuestionarios en línea. Este subsistema ha sido construido con la librería Socket.IO, explicada anteriormente, permitiendo una comunicación bidireccional basada en eventos entre el cliente y el servidor.

### 6.3.1.3 Aplicación web

Por último, se encuentra el subsistema correspondiente a la aplicación web, encargada de interactuar con los diferentes usuarios y de llevar a cabo todas las funcionalidades necesarias para cumplir los objetivos del sistema. Se ha construido usando el framework de JavaScript, Angular y algunas herramientas para los estilos de la página, como Bootstrap y Angular Material. La aplicación está dividida principalmente en componentes, que son los que se muestran al usuario, y servicios, especializados en comunicarse, tanto con el subsistema de la API Rest, como con el servicio de sockets.

## 6.3.2 Descripción de los Interfaces entre Subsistemas

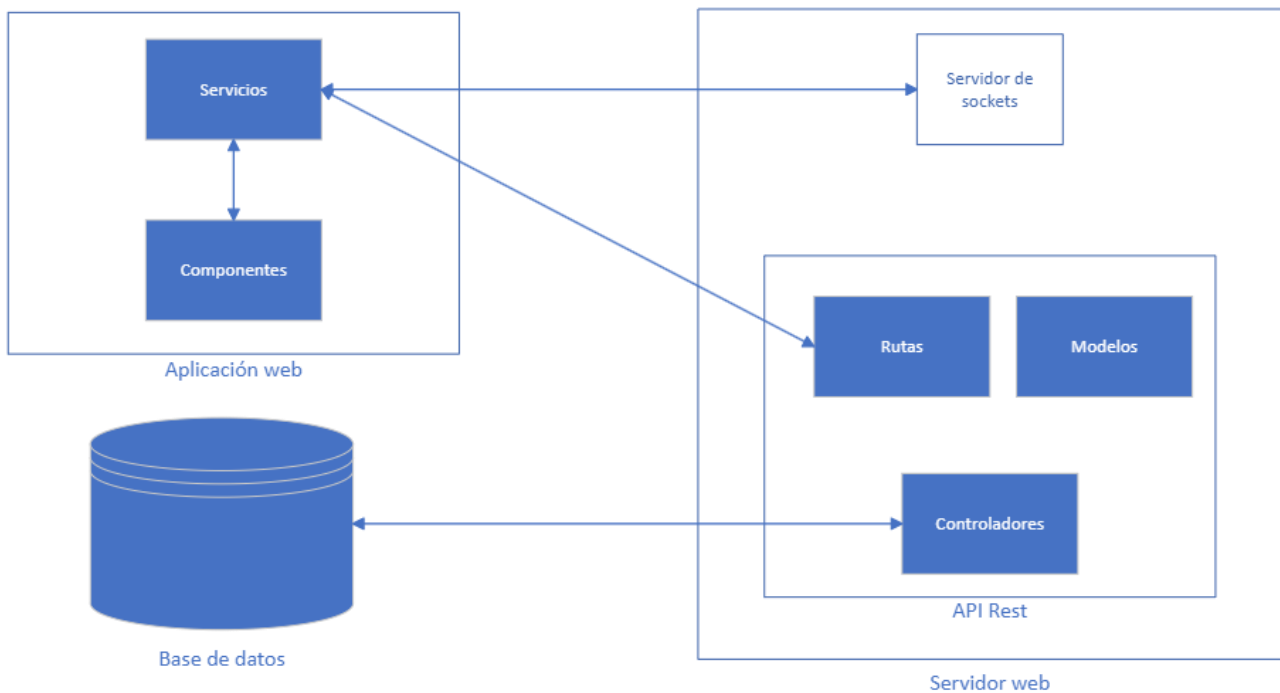
Las principales comunicaciones que se llevan a cabo son:

- **Entre la aplicación web y el servidor web:** esta comunicación será realizada mediante una API Rest, donde el servicio recibe peticiones en forma de URLs. La aplicación web está compuesta de varios servicios encargados de realizar las peticiones http correspondientes.
- **Entre la aplicación web y el servidor de sockets:** la comunicación, en este caso, se realiza con una librería construida sobre el protocolo de transmisión de datos, websocket. Esto permite

hacer que se puedan ver cambios en la aplicación web en tiempo real, y, por tanto, llevar a cabo un cuestionario en línea.

- Entre el servidor web y la base de datos: la comunicación se realiza haciendo uso del ORM prisma, que la abstrae y hace que sea muy sencilla la creación, modificación, borrado y la lectura de datos de la base de datos.

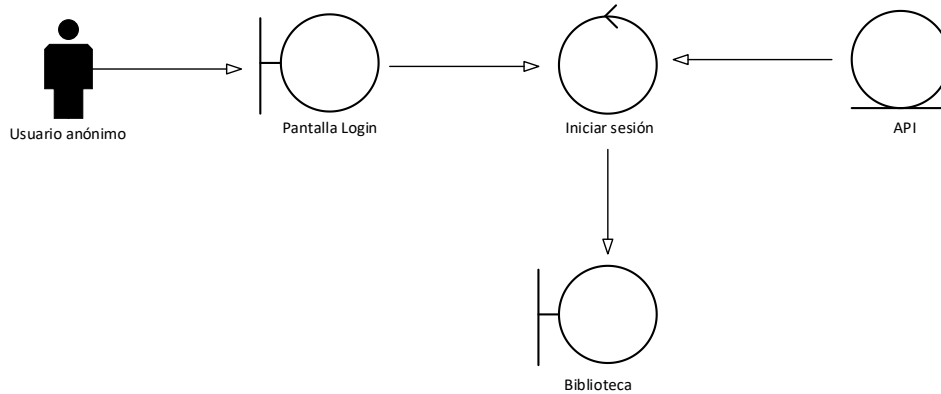
A continuación, se expone un diagrama que muestra las comunicaciones entre los subsistemas:



**Ilustración 7 DIAGRAMA DE COMUNICACIONES ENTRE SUBSISTEMAS**

## 6.4 ASI 4: ANÁLISIS DE LOS CASOS DE USO

### 6.4.1 Iniciar sesión



*Ilustración 8 DIAGRAMA DE ROBUSTEZ DE INICIO DE SESIÓN*

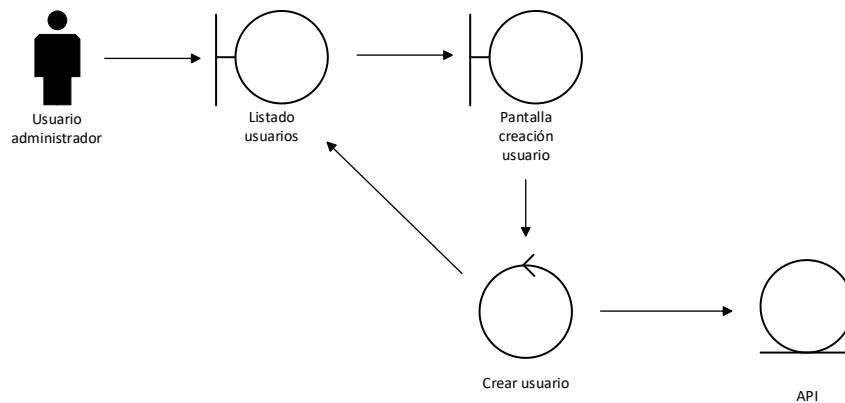
INICIAR SESIÓN	
<b>PRECONDICIONES</b>	El usuario debe existir en el sistema.
<b>POSCONDICIONES</b>	El usuario debe estar validado y con un rol asignado. Además, será redirigido a la página principal del administrador, profesor o alumno, dependiendo de su rol.
<b>ACTORES</b>	Iniciado y terminado por el mismo usuario (administrador, profesor o alumno)
<b>DESCRIPCIÓN</b>	<p>El usuario anónimo:</p> <p>El sistema muestra la pantalla de login</p> <p>El usuario rellena los campos con su nombre de usuario y su contraseña</p> <p>El usuario se identifica correctamente en el sistema y el sistema lo redirige a la pantalla principal.</p>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<p><b>Escenario Alternativo 1:</b> Inicio de sesión erróneo por credenciales incorrectas.</p> <p>El sistema notifica el hecho al usuario, sin dar detalles de lo que falta por seguridad.</p> <p>Volver al paso 1 del escenario principal</p> <p><b>Escenario Alternativo 2:</b> El usuario no rellena todos los campos obligatorios.</p> <p>El sistema notifica el usuario el motivo del error.</p> <p>Volver al paso 1 del escenario principal</p>

**EXCEPCIONES**

**La API no está disponible:** No es posible recibir la respuesta de la API.  
Notificar un error asociado al problema encontrado

**NOTAS**

### 6.4.2 Crear usuario



**Ilustración 9 DIAGRAMA DE ROBUSTEZ DE LA CREACIÓN DE UN USUARIO**

CREAR USUARIO	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como administrador
<b>POSCONDICIONES</b>	El usuario será añadido al sistema, reflejado como una nueva fila en la tabla de usuarios de la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de administrador.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador se encuentra en el listado de usuarios.</li> <li>2. El usuario administrador presiona la acción de crear usuario.</li> <li>3. El sistema muestra al usuario la pantalla de creación de un nuevo usuario.</li> <li>4. El usuario introduce el nombre del usuario y su rol.</li> <li>5. El sistema guarda el nuevo usuario en la base de datos.</li> <li>6. El sistema muestra un mensaje de éxito al usuario.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<p><b>Escenario Alternativo 1:</b> El usuario no rellena el campo del nombre. El sistema muestra un mensaje de error indicando el motivo.</p> <p><b>Escenario Alternativo 2:</b> El usuario introduce un nombre idéntico a otro ya existente. El sistema muestra un mensaje de error indicando el motivo.</p>

## EXCEPCIONES

**La base de datos no está disponible:** No se pueden obtener usuarios. El sistema notifica el error al usuario.

**El servicio no está disponible:** La API no valida el nuevo usuario ni se conecta con la base de datos. El sistema notifica el error al usuario.

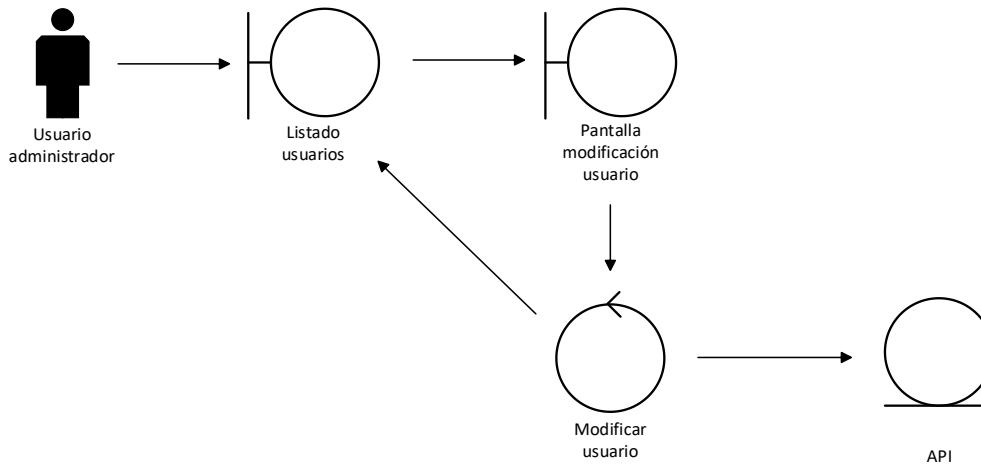
## NOTAS

### 6.4.3 Crear curso

El diagrama de robustez para crear un curso es exactamente igual al de crear usuario anterior cambiando el listado de usuarios por el listado de cursos, la pantalla de creación de usuarios por la pantalla de creación de cursos y la acción de crear usuario se corresponde con crear curso.

CREAR CURSO	
PRECONDICIONES	El usuario debe estar identificado como administrador
POSCONDICIONES	El usuario será añadido al sistema, reflejado como una nueva fila en la tabla de cursos de la base de datos.
ACTORES	Iniciada y finalizada por el usuario con el rol de administrador.
DESCRIPCIÓN	<ol style="list-style-type: none"> <li>1. El usuario administrador se encuentra en el listado de cursos.</li> <li>2. El usuario administrador presiona la acción de crear curso.</li> <li>3. El sistema muestra al usuario la pantalla de creación de un nuevo curso.</li> <li>4. El usuario introduce el nombre del curso y su descripción.</li> <li>5. El sistema guarda el nuevo curso en la base de datos.</li> <li>6. El sistema muestra un mensaje de éxito al usuario.</li> </ol>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<p><b>Escenario Alternativo 1:</b> El usuario no introduce alguno de los dos campos obligatorios. El sistema muestra un mensaje de error indicando el motivo.</p> <p><b>Escenario Alternativo 2:</b> El usuario introduce un nombre idéntico a otro ya existente. El sistema muestra un mensaje de error indicando el motivo.</p> <p><b>Escenario Alternativo 3:</b> Usuario y/o contraseña inválidos introducidos más de 5 veces seguidas</p>
EXCEPCIONES	<p><b>La base de datos no está disponible:</b> No se pueden obtener cursos. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no valida el nuevo curso ni se conecta con la base de datos. El sistema notifica el error al usuario.</p>
NOTAS	

## 6.4.4 Modificar usuario



*Ilustración 10 DIAGRAMA DE ROBUSTEZ DE LA MODIFICACIÓN DE UN USUARIO*

MODIFICAR USUARIO	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como administrador. El usuario a modificar debe existir.
<b>POSCONDICIONES</b>	El usuario modificado será actualizado en la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de administrador.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador se encuentra en el listado de usuarios.</li> <li>2. El usuario administrador presiona en un usuario.</li> <li>3. El sistema muestra al usuario la pantalla de modificación de usuario.</li> <li>4. El usuario introduce el nuevo nombre o rol.</li> <li>5. El sistema actualiza dicho usuario.</li> <li>6. El sistema muestra un mensaje de éxito al usuario.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<b>Escenario Alternativo 1:</b> El usuario introduce un nombre idéntico a otro ya existente. El sistema muestra un mensaje de error indicando el motivo.
<b>EXCEPCIONES</b>	<b>La base de datos no está disponible:</b> No se pueden obtener usuarios. El sistema notifica el error al usuario. <b>El servicio no está disponible:</b> La API no valida la modificación ni se conecta con la base de datos. El sistema notifica el error al usuario.
<b>NOTAS</b>	

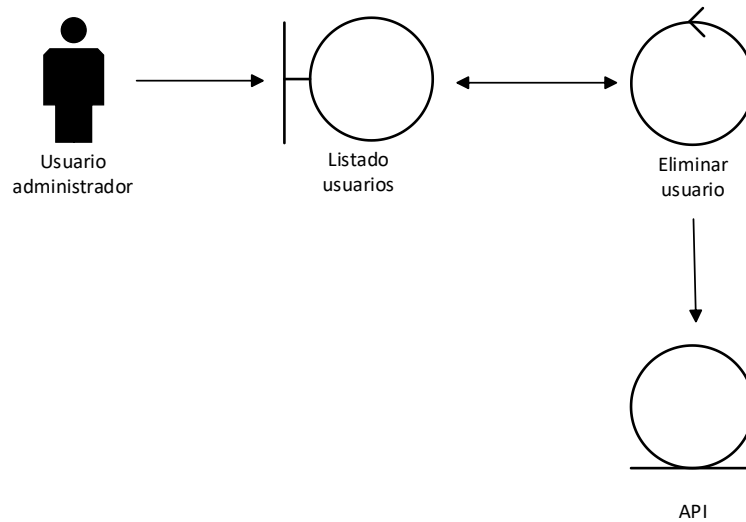


## 6.4.5 Modificar curso

El diagrama de robustez es similar al caso de uso de modificar usuario. Las diferencias son que, en vez del listado de usuarios, el usuario empieza en el listado de cursos, la pantalla de modificación de usuarios ahora es de modificación de cursos, y la acción es modificar cursos.

MODIFICAR CURSO	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como administrador. El curso a modificar debe existir.
<b>POSCONDICIONES</b>	El curso modificado será actualizado en la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de administrador.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"><li>1. El usuario administrador se encuentra en el listado de cursos.</li><li>2. El usuario administrador presiona en un curso.</li><li>3. El sistema muestra al usuario la pantalla de modificación de curso.</li><li>4. El usuario introduce el nuevo nombre o descripción del curso.</li><li>5. El sistema actualiza dicho curso.</li><li>6. El sistema muestra un mensaje de éxito al usuario.</li></ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<b>Escenario Alternativo 1:</b> El usuario no introduce ningún dato. El sistema muestra un mensaje de error indicando el motivo. <b>Escenario Alternativo 2:</b> El usuario introduce un nombre idéntico a otro ya existente. El sistema muestra un mensaje de error indicando el motivo.
<b>EXCEPCIONES</b>	<b>La base de datos no está disponible:</b> No se pueden obtener cursos. El sistema notifica el error al usuario. <b>El servicio no está disponible:</b> La API no valida la modificación ni se conecta con la base de datos. El sistema notifica el error al usuario.
<b>NOTAS</b>	

## 6.4.6 Eliminar usuario



*Ilustración 11 DIAGRAMA DE ROBUSTEZ DE LA ELIMINACIÓN DE UN USUARIO*

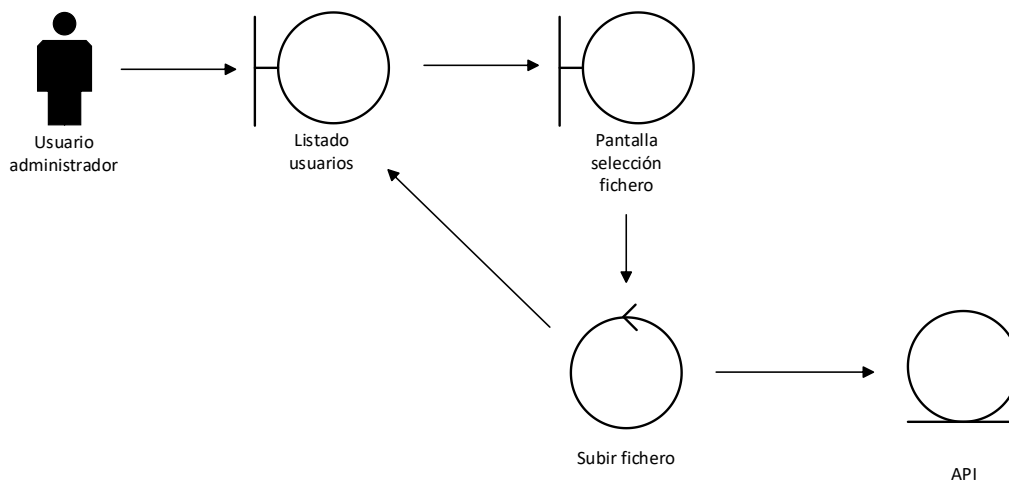
ELIMINAR USUARIO	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como administrador. El usuario a eliminar debe existir.
<b>POSCONDICIONES</b>	El usuario será eliminado de la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de administrador.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador se encuentra en el listado de usuarios.</li> <li>2. El usuario administrador elimina alguno de los usuarios existentes.</li> <li>3. El sistema elimina dicho usuario.</li> <li>4. El sistema muestra un mensaje de éxito al usuario.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se pueden obtener usuarios. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no elimina al usuario de la base de datos. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

### 6.4.7 Eliminar curso

El diagrama de robustez es idéntico al del caso de uso de eliminar usuario, cambiando solo el listado de usuarios por el listado de cursos y la acción de eliminar usuario por la acción de eliminar curso.

ELIMINAR CURSO	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como administrador. El curso a eliminar debe existir.
<b>POSCONDICIONES</b>	El curso será eliminado de la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de administrador.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario administrador se encuentra en el listado de cursos.</li> <li>2. El usuario administrador elimina alguno de los cursos existentes.</li> <li>3. El sistema elimina dicho curso.</li> <li>4. El sistema muestra un mensaje de éxito al usuario.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS) EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se pueden obtener curso. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no elimina al curso de la base de datos. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

### 6.4.8 Importar usuarios



*Ilustración 12 DIAGRAMA DE ROBUSTEZ DEL IMPORTE DE USUARIOS*



IMPORTAR USUARIOS	
PRECONDICIONES	El usuario debe estar identificado como administrador.
POSCONDICIONES	El o los usuarios contenidos dentro del fichero son creados en la base de datos.
ACTORES	Iniciada y finalizada por el usuario con el rol de administrador.
DESCRIPCIÓN	<ol style="list-style-type: none"><li>1. El usuario administrador se encuentra en el listado de usuarios.</li><li>2. El usuario administrador selecciona la opción de importar usuarios.</li><li>3. El sistema muestra la ventana de selección de fichero.</li><li>4. El usuario selecciona el fichero correspondiente.</li><li>5. El sistema lo válida y lo sube al servidor.</li></ol>
VARIACIONES (ESCENARIOS SECUNDARIOS)	<p><b>Escenario alternativo 1:</b> El fichero no está en formato csv. El sistema no sube el fichero y muestra el error al usuario.</p> <p><b>Escenario alternativo 2:</b> El fichero está en formato csv, pero no tiene internamente el formato de csv adecuado. El sistema muestra el mensaje de error al usuario.</p>
EXCEPCIONES	<p><b>La base de datos no está disponible:</b> No se pueden añadir los usuarios. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no recibe ni procesa el fichero con los usuarios. El sistema notifica el error al usuario.</p>
NOTAS	

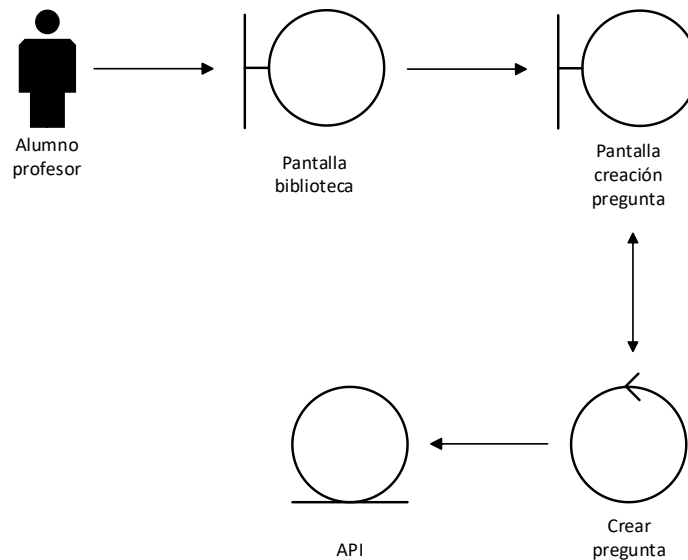
## 6.4.9 Importar cursos

El diagrama de robustez de este caso de uso es idéntico al anterior, cambiando únicamente la pantalla con el listado de usuarios a pantalla con el listado de cursos.

IMPORTAR CURSOS	
PRECONDICIONES	El usuario debe estar identificado como administrador.
POSCONDICIONES	El o los cursos contenidos dentro del fichero son creados en la base de datos.
ACTORES	Iniciada y finalizada por el usuario con el rol de administrador.
DESCRIPCIÓN	<ol style="list-style-type: none"><li>1. El usuario administrador se encuentra en el listado de cursos.</li><li>2. El usuario administrador selecciona la opción de importar cursos.</li><li>3. El sistema muestra la ventana de selección de fichero.</li><li>4. El usuario selecciona el fichero correspondiente.</li><li>5. El sistema lo válida y lo sube al servidor.</li></ol>

<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<p><b>Escenario alternativo 1:</b> El fichero no está en formato csv. El sistema no sube el fichero y muestra el error al usuario.</p> <p><b>Escenario alternativo 2:</b> El fichero está en formato csv, pero no tiene internamente el formato de csv adecuado. El sistema muestra el mensaje de error al usuario.</p>
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se pueden añadir los cursos. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no recibe ni procesa el fichero con los usuarios. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

### 6.4.10 Crear pregunta



**Ilustración 13 DIAGRAMA DE ROBUSTEZ DE CRACIÓN DE PREGUNTA**

<b>CREAR PREGUNTA</b>	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor.
<b>POSCONDICIONES</b>	La pregunta, así como las respuestas que la acompañen, serán insertadas en la base de datos en sus respectivas tablas.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario profesor, desde la pantalla de biblioteca, selecciona la opción de crear pregunta.</li> <li>2. El sistema redirige al usuario a la pantalla de creación de pregunta.</li> <li>3. El usuario introduce los siguientes campos: <ol style="list-style-type: none"> <li>a. Enunciado de la pregunta.</li> <li>b. Tiempo para realizar la pregunta.</li> </ol> </li> </ol>



	<p>c. Tipo de pregunta.</p> <p>d. Respuestas.</p> <p>e. Opcionalmente puede añadir una imagen.</p> <p>4. El sistema valida y guarda la pregunta.</p>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<p><b>Escenario alternativo 1:</b> El usuario profesor no ha introducido todos los campos obligatorios. El sistema muestra un mensaje de error.</p> <p><b>Escenario alternativo 2:</b> El fichero no es de un tipo válido de imagen. El sistema muestra un mensaje de error indicando el motivo.</p> <p><b>Escenario alternativo 3:</b> La pregunta es de tipo multiopción y el usuario no ha seleccionado al menos una respuesta correcta. El sistema se lo notifica al usuario.</p> <p><b>Escenario alternativo 4:</b> La pregunta es de tipo verdadero o falso y el usuario no ha seleccionado una respuesta correcta. El sistema se lo notifica al usuario.</p>
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se pueden añadir las preguntas ni las respuestas.. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La aplicación no puede comunicarse con el servicio para crear la pregunta. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

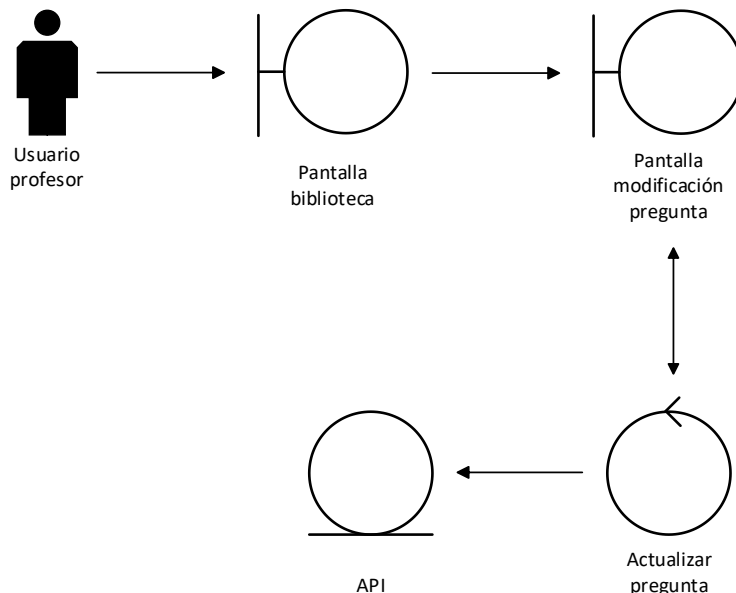
### 6.4.11 Crear cuestionario

El diagrama de robustez es igual que el anterior cambiando la pantalla de creación de preguntas por la pantalla de creación de cuestionario y la acción de crear pregunta por crear cuestionario.

<b>CREAR CUESTIONARIO</b>	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor y debe de haber creado al menos una pregunta.
<b>POSCONDICIONES</b>	El cuestionario será añadido a su respectiva tabla en la base de datos, así como las referencias a las preguntas que lo compongan.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario profesor, desde la pantalla de biblioteca, selecciona la opción de crear cuestionario.</li> <li>2. El sistema redirige al usuario a la pantalla de creación de cuestionario.</li> <li>3. El usuario deberá introducir el nombre del cuestionario y, opcionalmente, podrá añadir una imagen.</li> </ol>

	<p>4. El usuario podrá ir añadiendo y eliminando preguntas que haya creado con anterioridad al cuestionario. También podrá cambiarlas de orden.</p> <p>5. El sistema valida y guarda el cuestionario.</p>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<p><b>Escenario alternativo 1:</b> El usuario profesor no ha introducido el nombre del cuestionario. El sistema muestra un mensaje de error.</p> <p><b>Escenario alternativo 2:</b> El usuario profesor no añade al menos una pregunta. El sistema muestra un mensaje de error indicando el motivo.</p> <p><b>Escenario alternativo 3:</b> El usuario profesor intenta subir un fichero no correspondiente a una imagen. El sistema muestra un mensaje de error indicando el motivo.</p>
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se puede añadir el cuestionario. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La aplicación no puede comunicarse con el servicio para crear el cuestionario. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

## 6.4.12 Modificar pregunta



**Ilustración 14 DIAGRAMA DE ROBUSTEZ DE LA MODIFICACIÓN DE PREGUNTA**

<b>MODIFICAR PREGUNTA</b>	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor y la pregunta debe existir.



<b>POSCONDICIONES</b>	La pregunta se actualizará en la base de datos, así como las respuestas.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario profesor, desde la pantalla de biblioteca, selecciona la opción de modificar pregunta.</li> <li>2. El sistema redirige al usuario a la pantalla de modificación de pregunta.</li> <li>3. El usuario podrá modificar cualquiera de los campos.</li> <li>4. El sistema valida y actualiza la pregunta.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<p><b>Escenario alternativo 1:</b> El usuario profesor borra todos los campos e intenta actualizar. El sistema muestra un mensaje de error indicando el motivo.</p> <p><b>Escenario alternativo 2:</b> En el caso de que la pregunta sea de tipo multiopción, el usuario marca todas las preguntas como falsas. Debe haber al menos una verdadera, por lo que el sistema muestra un mensaje de error.</p> <p><b>Escenario alternativo 3:</b> El usuario añade o modifica el campo de imagen para subir un archivo que no corresponda con una. El sistema muestra un mensaje de error</p>
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se puede actualizar la pregunta. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La aplicación no puede comunicarse con el servicio para actualizar la pregunta. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

### 6.4.13 Modificar cuestionario

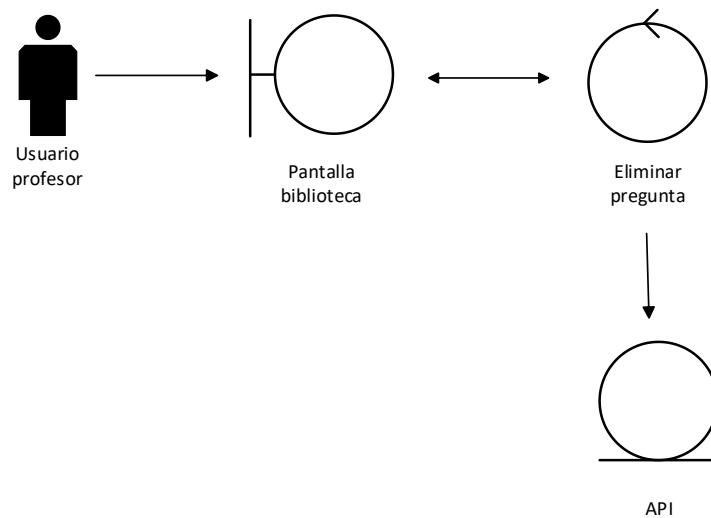
El diagrama para el caso de uso de modificar cuestionario es idéntico al anterior, cambiando la pantalla de modificación de pregunta por la pantalla de modificación de cuestionario y la acción de actualizar pregunta por actualizar cuestionario.

<b>MODIFICAR PREGUNTA</b>	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor y el cuestionario debe existir.
<b>POSCONDICIONES</b>	El cuestionario se actualizará en la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario profesor, desde la pantalla de biblioteca, selecciona la opción de modificar cuestionario.</li> <li>2. El sistema redirige al usuario a la pantalla de modificación de cuestionario.</li> </ol>



	<p>3. El usuario podrá modificar tanto el título, como la imagen y las preguntas que contiene le cuestionarios. También podrá cambiar el orden de las preguntas.</p> <p>4. El sistema valida y actualiza el cuestionario.</p>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<p><b>Escenario alternativo 1:</b> El usuario profesor elimina todas las preguntas del cuestionario y lo envía. Un cuestionario debe tener al menos una pregunta, por lo que el sistema muestra un mensaje de error al usuario.</p> <p><b>Escenario alternativo 2:</b> El</p>
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se puede actualizar el cuestionario. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La aplicación no puede comunicarse con el servicio para actualizar el cuestionario. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

#### 6.4.14 Eliminar pregunta



*Ilustración 15 DIAGRAMA DE ROBUSTEZ DE LA ELIMINACIÓN DE UNA PREGUNTA*

ELIMINAR PREGUNTA	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor. La pregunta a eliminar debe existir.
<b>POSCONDICIONES</b>	Tanto la pregunta como todas sus respuestas son eliminadas de la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	1. El usuario profesor se encuentra en la pantalla de biblioteca.



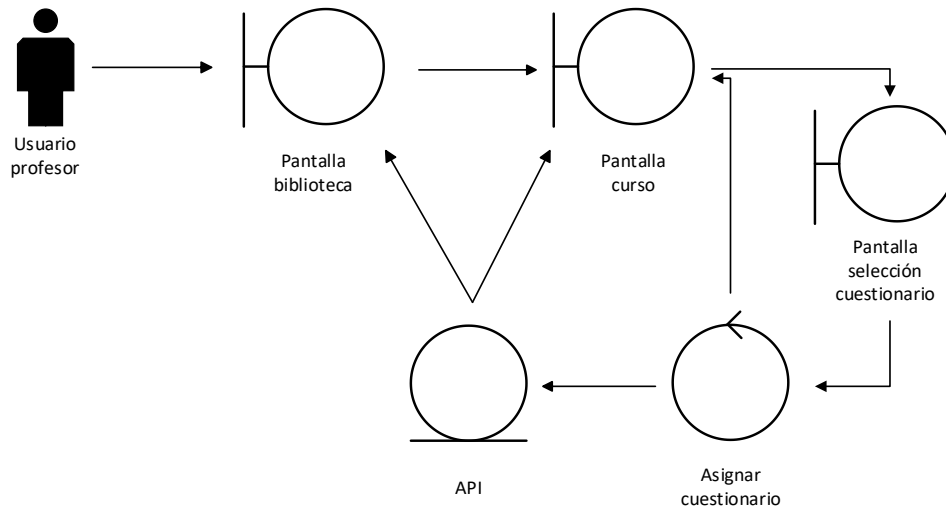
	<ol style="list-style-type: none"><li>2. El usuario profesor elimina alguna pregunta creada con anterioridad.</li><li>3. El sistema elimina dicha pregunta.</li><li>4. El sistema muestra un mensaje de éxito al usuario.</li></ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se pueden eliminar la pregunta. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no elimina la pregunta de la base de datos. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

## 6.4.15 Eliminar cuestionario

El diagrama de robustez es igual que en el caso anterior, solo cambia la acción de eliminar pregunta por eliminar cuestionario.

<b>ELIMINAR CUESTIONARIO</b>	
<b>PRECONDICIONES</b>	<p>El usuario debe estar identificado como profesor.</p> <p>El cuestionario a eliminar debe existir.</p>
<b>POSCONDICIONES</b>	<p>El cuestionario correspondiente se elimina de la base de datos.</p>
<b>ACTORES</b>	<p>Iniciada y finalizada por el usuario con el rol de profesor.</p>
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"><li>1. El usuario profesor se encuentra en la pantalla de biblioteca.</li><li>2. El usuario profesor elimina algún cuestionario creado con anterioridad.</li><li>3. El sistema elimina dicho cuestionario.</li><li>4. El sistema muestra un mensaje de éxito al usuario.</li></ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se pueden eliminar el cuestionario. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no elimina el cuestionario de la base de datos. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

## 6.4.16 Asignar cuestionario a un curso



**Ilustración 16 DIAGRAMA DE ROBUSTEZ DE INSERCIÓN DE CUESTIONARIO EN UN CURSO**

<b>ASIGNAR CUESTIONARIO A UN CURSO</b>	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor. El curso y el cuestionario deben existir.
<b>POSCONDICIONES</b>	El cuestionario se añade al curso mediante una relación en la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario profesor se encuentra en la pantalla de biblioteca.</li> <li>2. El usuario profesor elije un curso y navega hasta la pantalla de curso.</li> <li>3. El usuario elije la opción de asignar cuestionario.</li> <li>4. El sistema muestra, en la pantalla de selección de cuestionarios, todos los cuestionarios creados por el usuario que no hayan sido ya asignados al curso.</li> <li>5. El usuario elije el cuestionario que quiere asignar.</li> <li>6. El sistema añade el cuestionario al curso.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se puede crear la relación entre el curso y el cuestionario. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no procesa la petición de asignar el cuestionario al curso. El sistema notifica el error al usuario.</p>



## NOTAS

### 6.4.17 Asignar pregunta a un curso

El diagrama de robustez de este caso de uso es idéntico al caso de uso anterior, cambiando la pantalla de selección de cuestionario por pantalla de selección de pregunta y la acción de asignar cuestionario por asignar pregunta.

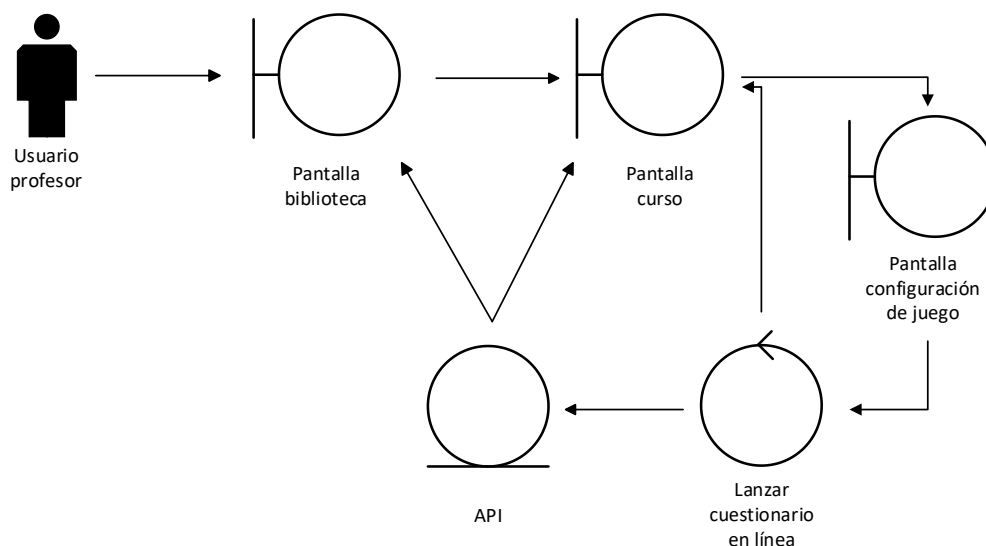
ASIGNAR PREGUNTA A UN CURSO	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor. El curso y la pregunta deben existir.
<b>POSCONDICIONES</b>	La pregunta se añade al curso mediante una relación en la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"><li>1. El usuario profesor se encuentra en la pantalla de biblioteca.</li><li>2. El usuario profesor elige un curso y navega hasta la pantalla de curso.</li><li>3. El usuario elige la opción de asignar pregunta.</li><li>4. El sistema muestra, en la pantalla de selección de preguntas, todas las preguntas creadas por el usuario que no hayan sido ya asignadas.</li><li>5. El usuario elige la pregunta que quiere asignar.</li><li>6. El sistema asigna la pregunta al curso.</li></ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	
<b>EXCEPCIONES</b>	<b>La base de datos no está disponible:</b> No se puede crear la relación entre el curso y la pregunta. El sistema notifica el error al usuario. <b>El servicio no está disponible:</b> La API no procesa la petición de asignar la pregunta al curso. El sistema notifica el error al usuario.
<b>NOTAS</b>	

### 6.4.18 Asignar usuario a un curso

El diagrama de robustez de este caso de uso es idéntico al caso de uso anterior, cambiando la pantalla de selección de cuestionario por pantalla de selección de usuario y la acción de asignar cuestionario por asignar usuario.

ASIGNAR USUARIO A UN CURSO	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como profesor. El usuario a asignar y la pregunta deben existir.
<b>POSCONDICIONES</b>	El usuario se añade al curso mediante una relación en la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de profesor.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario profesor se encuentra en la pantalla de biblioteca.</li> <li>2. El usuario profesor elije un curso y navega hasta la pantalla de curso.</li> <li>3. El usuario elije la opción de asignar usuario.</li> <li>4. El sistema muestra, en la pantalla de selección de usuarios, todos los alumnos que no hayan sido ya asignados.</li> <li>5. El usuario elije el usuario que quiere asignar.</li> <li>6. El sistema asigna la pregunta al curso.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se puede crear la relación entre el curso y el usuario. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> La API no procesa la petición de añadir el usuario al curso. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

### 6.4.19 Lanzar cuestionario en línea



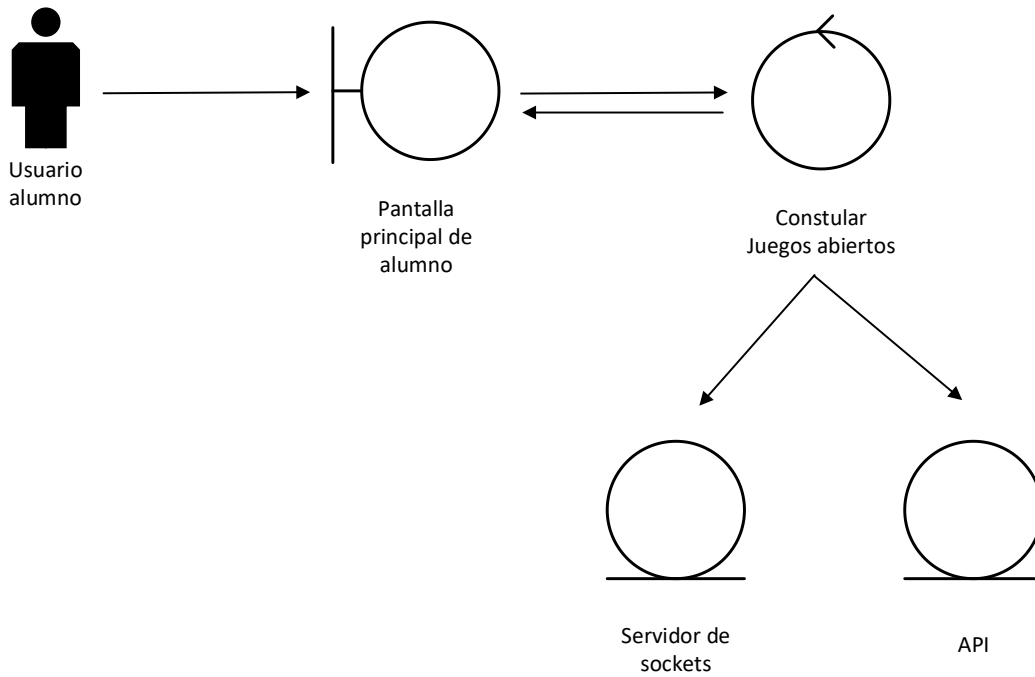
**Ilustración 17 DIAGRAMA DE ROBUSTEZ DE LANZAMIENTO DE CUESTIONARIO EN LÍNEA**



## LANZAR CUESTIONARIO EN LÍNEA

<b>PRECONDICIONES</b>	<p>El usuario debe estar identificado como profesor.</p> <p>El cuestionario a lanzar debe existir y estar asignado al curso donde se va a lanzar.</p>
<b>POSCONDICIONES</b>	<p>El usuario profesor lanza el juego y se crea una entidad juego en la base de datos.</p>
<b>ACTORES</b>	<p>Iniciada y finalizada por el usuario con el rol de profesor.</p>
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"><li>1. El usuario profesor se encuentra en la pantalla de biblioteca.</li><li>2. El usuario profesor elige un curso y navega hasta la pantalla de curso.</li><li>3. El usuario elige la opción de lanzar juego en uno de los cuestionarios que haya asignado al curso anteriormente.</li><li>4. El sistema muestra, en la pantalla de configuración de juego, las opciones del juego. Estas son:<ol style="list-style-type: none"><li>a. Si los alumnos pueden ver el enunciado en sus dispositivos.</li><li>b. El tipo de puntuación (normal, doble, sin puntuación).</li></ol></li><li>5. El usuario acepta.</li><li>6. El sistema lanza el juego.</li></ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	
<b>EXCEPCIONES</b>	<p><b>La base de datos no está disponible:</b> No se puede crear el juego. El sistema notifica el error al usuario.</p> <p><b>El servicio de sockets no está disponible:</b> El servicio no provee los sockets ni los eventos necesarios para ejecutar el juego. El sistema notifica el error al usuario.</p>
<b>NOTAS</b>	

## 6.4.20 Consultar cuestionarios abiertos



*Ilustración 18 DIAGRAMA DE ROBUSTEZ PARA LA CONSULTA DE CUESTIONARIOS ABIERTOS*

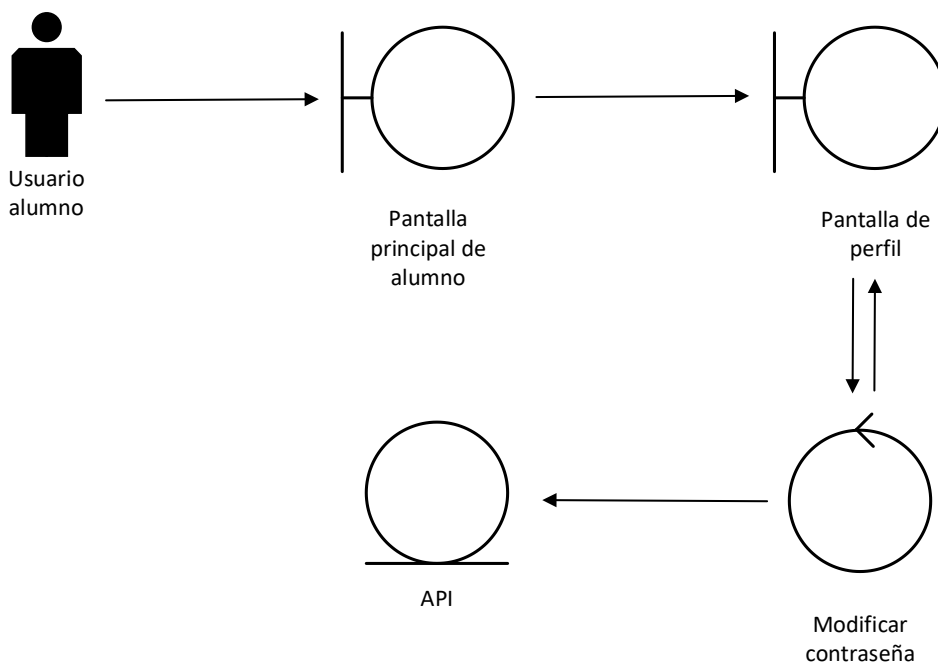
CONSTULAR CUESTIONARIOS ABIERTOS	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como alumno. El cuestionario debe haber sido lanzado anteriormente por el profesor y el alumno debe estar asignado al mismo curso al que lo está el cuestionario.
<b>POSCONDICIONES</b>	El usuario alumno verá todos los cuestionarios abiertos de los cursos en los que está asignado, si los hubiera.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de alumno.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario alumno se encuentra en la página principal.</li> <li>2. El sistema realiza una petición a la API para consultar si hay juegos abiertos.</li> <li>3. Mediante el servidor de sockets, el sistema queda a la espera por juegos, mostrando en tiempo real si un juego es abierto.</li> <li>4. El sistema muestra todos los juegos abiertos al usuario.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<b>Escenario alternativo 1:</b> no existe ningún juego abierto, por lo que la pantalla principal no se muestra ninguno.
<b>EXCEPCIONES</b>	<b>La base de datos no está disponible:</b> No se pueden obtener los juegos abiertos. El sistema notifica el error al usuario.

**El servicio de sockets no está disponible:** El servicio no provee los sockets ni los eventos necesarios para ver si se abre algún juego. El sistema notifica el error al usuario.

**El servicio no está disponible:** el servicio de la API no está disponible y no procesa la petición de consultar juegos abiertos. El sistema muestra un error al usuario.

NOTAS

### 6.4.21 Cambiar contraseña



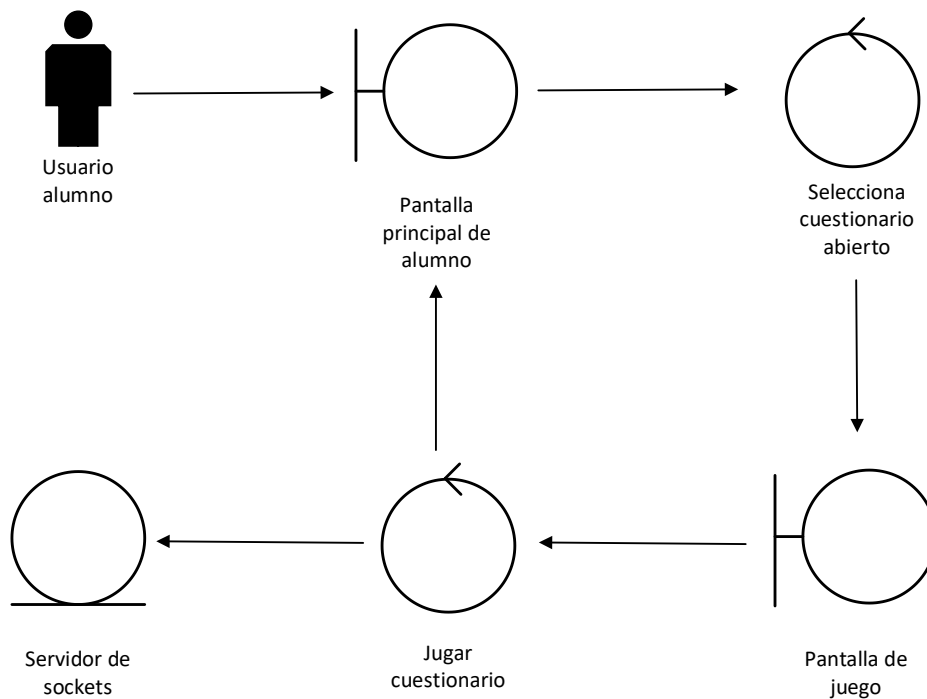
*Ilustración 19 DIAGRAMA DE ROBUSTEZ DEL CAMBIO DE CONTRASEÑA*

CAMBIAR CONTRASEÑA	
<b>PRECONDICIONES</b>	El usuario debe estar identificado como alumno.
<b>POSCONDICIONES</b>	La contraseña del usuario será modifica en la fila correspondiente de la base de datos.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de alumno.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario alumno se encuentra en la página principal.</li> <li>2. El usuario alumno navega hasta la pantalla de perfil.</li> <li>3. El usuario rellena un campo con la contraseña nueva y otro para confirmarla.</li> <li>4. El sistema modifica la contraseña y muestra un mensaje de éxito.</li> </ol>



<b>VARIACIONES (ESCENARIOS SECUNDARIOS) EXCEPCIONES</b>	<p><b>Escenario alternativo 1:</b> el jugador no rellena los dos campos. El sistema muestra un mensaje de error indicando el motivo.</p> <p><b>La base de datos no está disponible:</b> No se puede modificar la contraseña en la base de datos. El sistema notifica el error al usuario.</p> <p><b>El servicio no está disponible:</b> el servicio de la API no procesa el cambio de contraseña. El sistema muestra un error al usuario.</p>
<b>NOTAS</b>	

### 6.4.22 Jugar cuestionario en línea



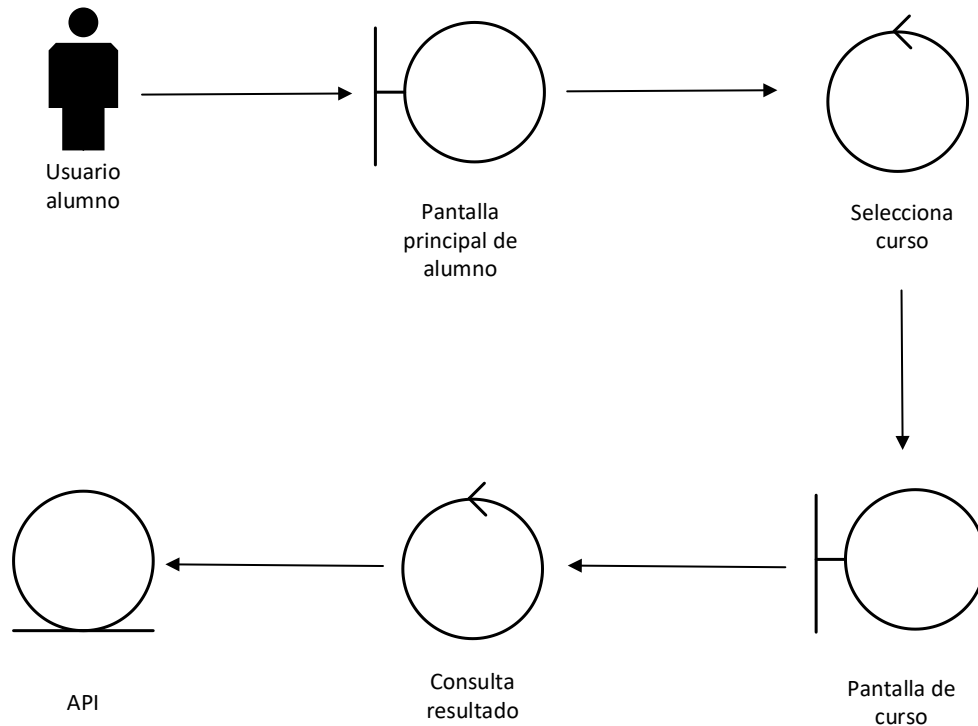
**Ilustración 20 DIAGRAMA DE ROBUSTEZ PARA JUGAR UN CUESTIONARIO EN LÍNEA**

<b>JUGAR CUESTIONARIO EN LÍNEA</b>	
<b>PRECONDICIONES</b>	<p>El usuario debe estar identificado como alumno.</p> <p>El usuario profesor tiene que haber abierto el juego previamente.</p> <p>El usuario alumno debe pertenecer al mismo curso al que pertenece el cuestionario.</p>
<b>POSCONDICIONES</b>	<p>El usuario alumno habrá jugado el cuestionario y, en consecuencia, existirá un resultado que será guardado en la base de datos.</p>
<b>ACTORES</b>	<p>Iniciada y finalizada por el usuario con el rol de alumno.</p>



<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"><li>1. El usuario alumno se encuentra en la página principal.</li><li>2. El usuario selecciona un juego abierto disponible.</li><li>3. El sistema redirige al usuario alumno a la pantalla de juego, donde se desarrollará el cuestionario.</li><li>4. Una vez terminado el juego, el sistema redirigirá al usuario alumno a la pantalla inicial.</li></ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<b>Escenario alternativo 1:</b> el usuario alumno se desconecta del juego. En todo momento, mientras el juego siga en curso, el jugador podrá volver a conectarse, volviendo al paso 1 del diagrama de robustez.
<b>EXCEPCIONES</b>	<b>La base de datos no está disponible:</b> No se puede consultar juegos abiertos. El sistema notifica el error al usuario. <b>El servicio de sockets no está disponible:</b> El servicio no provee los sockets ni los eventos necesarios para ver si se abre algún juego. Tampoco está disponible para llevar a cabo el juego. El sistema notifica el error al usuario. <b>El servicio no está disponible:</b> el servicio de la API no está disponible y no procesa la petición de consultar juegos abiertos. El sistema muestra un error al usuario.
<b>NOTAS</b>	

### 6.4.23 Consultar resultados



**Ilustración 21 DIAGRAMA DE ROBUSTEZ DE LA CONSULTA DE RESULTADOS**

CONSULTAR RESULTADOS	
<b>PRECONDICIONES</b>	<p>El usuario debe estar identificado como alumno.</p> <p>Para que un usuario alumno tenga resultados de un cuestionario, tiene que haberlo jugado.</p> <p>Para que un usuario alumno pueda consultar los resultados de un curso, primero tiene que pertenecer a este.</p>
<b>POSCONDICIONES</b>	El sistema mostrará todos los resultados de los cuestionarios jugados en ese curso.
<b>ACTORES</b>	Iniciada y finalizada por el usuario con el rol de alumno.
<b>DESCRIPCIÓN</b>	<ol style="list-style-type: none"> <li>1. El usuario alumno se encuentra en la página principal.</li> <li>2. El sistema muestra todos los cursos a los que pertenece el usuario alumno.</li> <li>3. El alumno selecciona un curso del que le interese conocer los resultados de los cuestionarios jugados.</li> <li>4. El sistema redirige al usuario alumno a la pantalla del curso.</li> <li>5. El sistema muestra los resultados de los cuestionarios jugados por el usuario alumno en ese curso.</li> </ol>
<b>VARIACIONES (ESCENARIOS SECUNDARIOS)</b>	<b>Escenario alternativo 1:</b> el usuario alumno no ha jugado ningún juego en ese curso, por lo que la vista de resultados estará vacía.



## EXCEPCIONES

**La base de datos no está disponible:** No se puede consultar los resultados ni los cursos. El sistema notifica el error al usuario.

**El servicio no está disponible:** la API no provee los cursos a los que pertenece el usuario ni los resultados de los mismos. El sistema muestra un error al usuario.

## NOTAS

## 6.5 ASI 5: ANÁLISIS DE CLASES

---

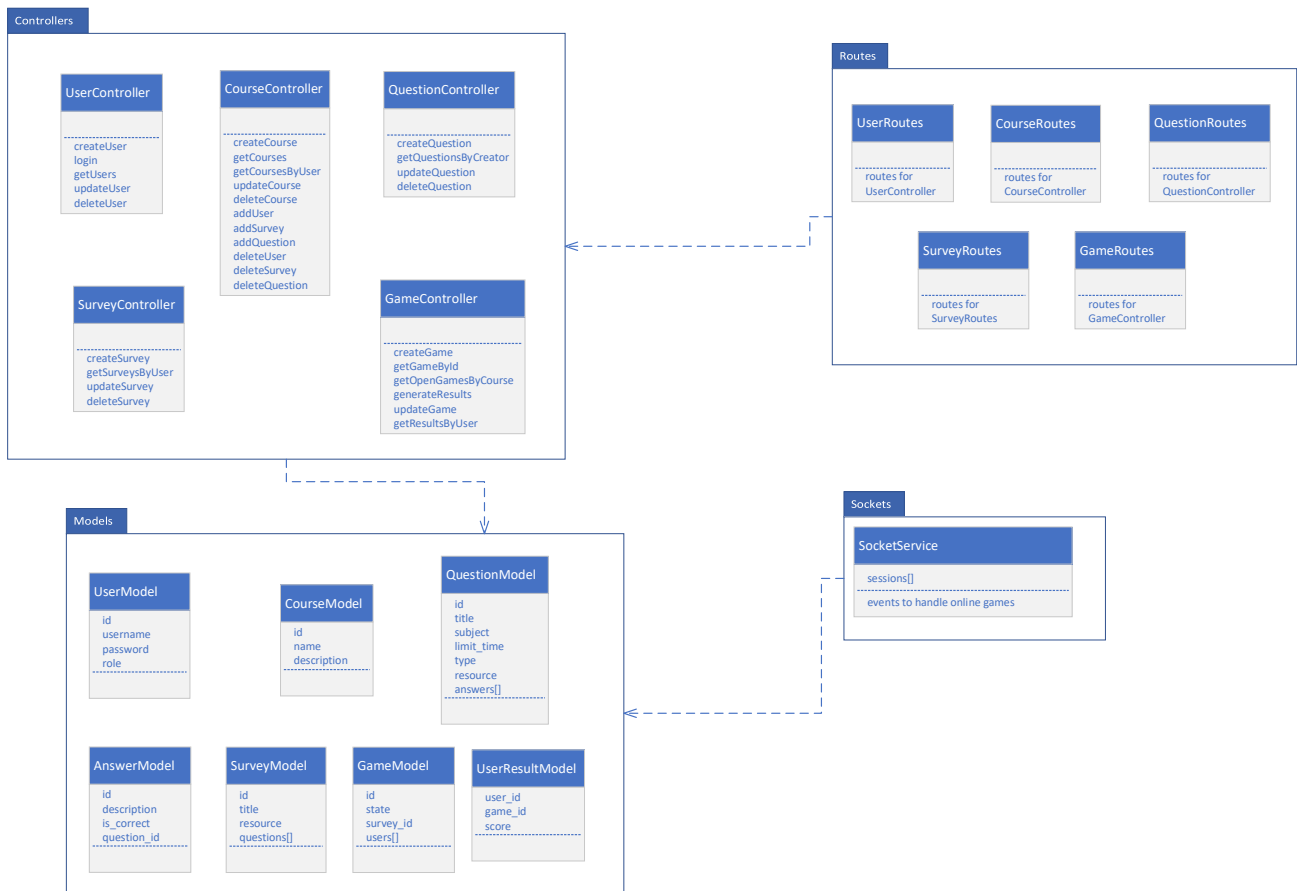
En este apartado se verán las clases que han sido identificadas en la fase de análisis del sistema y posteriormente, se verán las características de cada una. En primer lugar, se mostrarán los diagramas de clases de cada uno de los subsistemas identificados anteriormente, exceptuando el subsistema de base de datos, para luego entrar en detalle de cada una de las clases.

Cabe destacar que los diagramas no van a contener clases como tal, sino que, Nodejs utiliza ficheros .ts (TypeScript) y Angular componentes, que no son clases al uso.

### 6.5.1 Diagrama de Clases

Como comentaba antes, se subdividirá el diagrama de clases en varios más pequeños para que resulte más sencilla la comprensión.

### 6.5.1.1 Diagrama de clases del servidor web



**Ilustración 22 DIAGRAMA DE CLASES DEL SERVIDOR WEB**

## 6.5.1.2 Diagrama de clases de la aplicación web

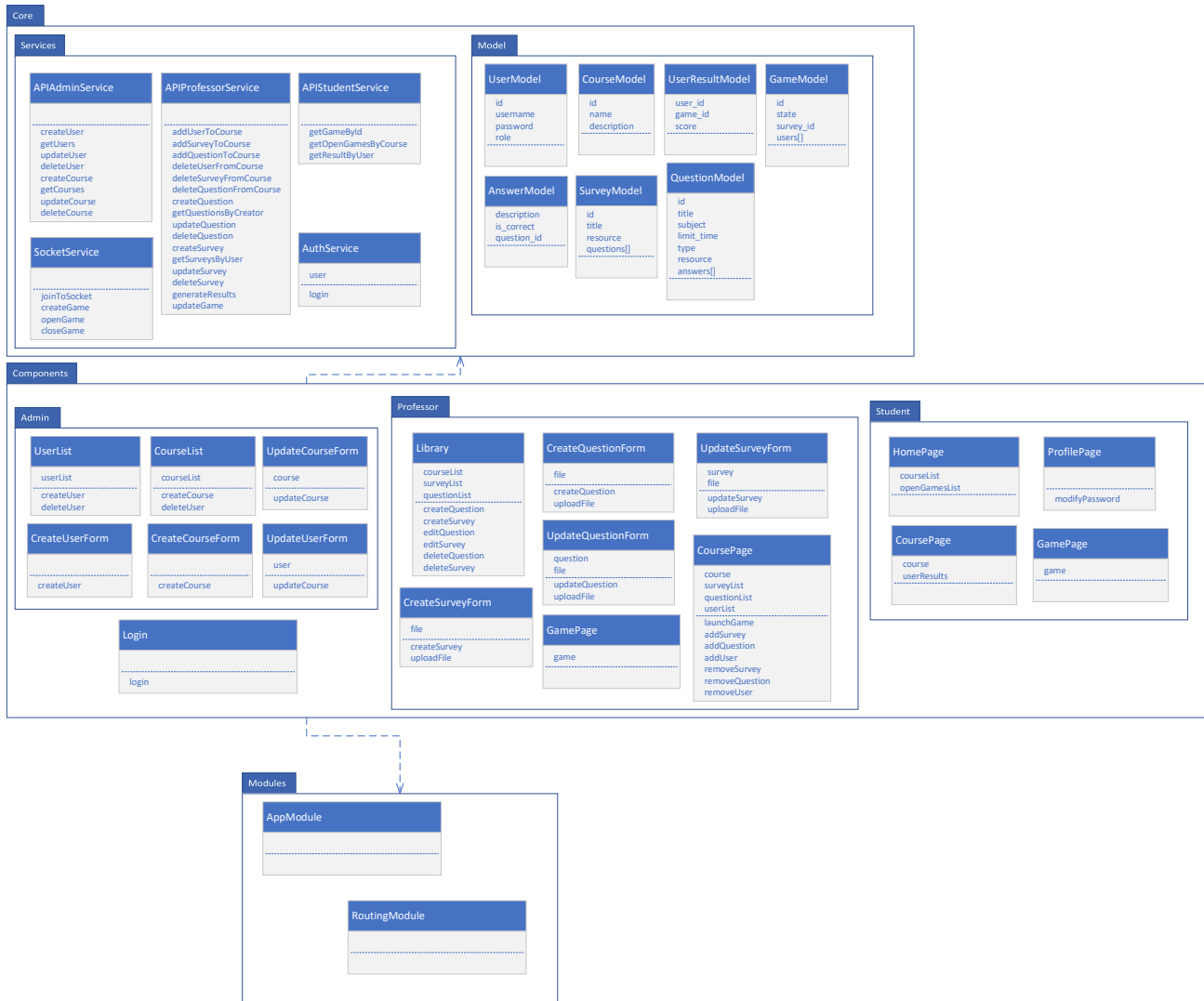


Ilustración 23 DIAGRAMA DE CLASES DE LA APLICACIÓN WEB

## 6.5.2 Descripción de las Clases

A continuación, se detallarán cada una de las clases vistas en cada uno de los diagramas, indicando la responsabilidad de cada una de ellas.

### 6.5.2.1 Servidor web

#### 6.5.2.1.1 Controllers

<b>Nombre de la clase</b>	
UserController	
<b>Descripción</b>	



Procesa las peticiones correspondientes a la gestión de usuarios y lleva a cabo una acción para realizar dicha gestión. Es la responsable de comunicarse con la base de datos para hacer el CRUD de la entidad usuario, así como de la funcionalidad de iniciar sesión en el sistema.

#### Responsabilidades

- CRUD de usuarios
- Iniciar sesión

#### Atributos Propuestos

#### Métodos Propuestos

**createUser:** creará un usuario con los parámetros pasados en la petición POST. Además, asignará una contraseña aleatoria a dicho usuario.

**login:** procesará una petición POST con los parámetros de un usuario que quiere iniciar sesión. Devolverá un token si el inicio de sesión es correcto, si no, enviará un mensaje de error.

**getUsers:** se trata de una petición GET donde devuelve todos los usuarios que existen en el sistema.

**updateUser:** petición PUT que actualiza un usuario determinado con los parámetros que le llegan.

**deleteUser:** petición DELETE que elimina un usuario según el id que le llega por parámetro.

#### Nombre de la clase

CourseController

#### Descripción

Procesa las peticiones correspondientes a la gestión de cursos, y realiza una acción para llevar a cabo dicha gestión. Se encarga de comunicarse con la base de datos para hacer el CRUD de la entidad curso y asignar usuarios, preguntas y cuestionarios a cursos.

#### Responsabilidades

- CRUD de cursos.
- Asignar usuarios, preguntas y cuestionarios a cursos.

#### Atributos Propuestos

#### Métodos Propuestos

**createCourse:** crea un curso con los parámetros que se le pasa mediante una petición POST.

**getCourses:** es una petición GET que devuelve todos los cursos que existen en el sistema.

**getCoursesByUser:** petición GET que devuelve los cursos en los que está asignado un alumno.

**updateCourse:** petición POST que actualiza un curso con los datos que le llegan.

**deleteUser:** petición DELETE que elimina el curso con el id que se le pasa por parámetro.

**addUser:** se encarga de asignar el usuario pasado por parámetro a un curso, pasado por parámetro también.

**addSurvey:** idéntico al anterior, pero con los cuestionarios.

**addQuestion:** idéntico al anterior, pero con las preguntas.

**deleteUser:** elimina la asignación de un usuario a un curso, ambos pasados por parámetro.



**deleteSurvey:** idéntico al anterior, pero con los cuestionarios.

**deleteQuestion:** idéntico al anterior, pero con las preguntas.

<b>Nombre de la clase</b>
---------------------------

QuestionController
--------------------

<b>Descripción</b>
--------------------

Procesa las peticiones correspondientes con la gestión de preguntas. Se comunica con la base de datos para realizar el CRUD de la entidad pregunta, así como la gestión de las respuestas.

<b>Responsabilidades</b>
--------------------------

- CRUD de preguntas.

<b>Atributos Propuestos</b>
-----------------------------

<b>Métodos Propuestos</b>
---------------------------

**createQuestion:** crea una pregunta y sus respuestas asociadas con los parámetros que le llagan de la petición.

**getQuestionsByCreator:** petición GET que devuelve todas las preguntas creadas por el creador cuyo id se pasa por parámetro.

**updateQuestion:** petición PUT que actualiza una pregunta con los parámetros pasados.

**deleteQuestions:** petición DELETE que elimina la pregunta cuyo id es pasado por parámetro.

<b>Nombre de la clase</b>
---------------------------

SurveyController
------------------

<b>Descripción</b>
--------------------

Procesa las peticiones necesarias para llevar a cabo la gestión de cuestionarios. Se comunica con la base de datos para realizar el CRUD de cuestionarios.

<b>Responsabilidades</b>
--------------------------

- CRUD de cuestionarios.

<b>Atributos Propuestos</b>
-----------------------------

<b>Métodos Propuestos</b>
---------------------------

**createSurvey:** crea un cuestionario con los parámetros que se le pasan. Además, crea las relaciones con las preguntas asociadas al cuestionario.

**getSurveysByUser:** retorna una lista de cuestionarios creados por el usuario pasado por parámetro.

**updateSurvey:** actualiza un cuestionario con los datos pasados por parámetro.

**deleteSurvey:** elimina el cuestionario cuyo id se pasa por parámetro.

<b>Nombre de la clase</b>
---------------------------

GameController
----------------





Descripción
Procesa las peticiones correspondientes para llevar a cabo la gestión de juegos. Se encarga de comunicarse con la base de datos para realizar el CRUD de la entidad juego.
Responsabilidades
<ul style="list-style-type: none"><li>• CRUD de juegos</li></ul>
Atributos Propuestos
Métodos Propuestos
<b>createGame:</b> crea un juego con los parámetros pasados. Además, crea la relación en la base de datos con el cuestionario al que corresponde el juego. <b>getGameById:</b> retorna el juego correspondiente al id que se pasa por parámetro. <b>getOpenGamesByCourse:</b> devuelve todos los juegos asociados a un curso pasado por parámetro que estén en estado abierto. <b>generateResults:</b> crea los resultados de todos los alumnos que hayan participado en un juego. <b>getResultsByUser:</b> devuelve todos los resultados del usuario pasado por parámetro.

#### 6.5.2.1.2 Routes

Nombre de la clase
UserRoutes
Descripción
Clase encargada de redirigir las peticiones correspondientes a los usuarios al UserController según la URL pasada y el método HTTP utilizado.
Atributos Propuestos
Métodos Propuestos
Mismo número de métodos que el UserController. Cada uno de estos llama al método correspondiente del controlador

Nombre de la clase
CourseRoutes
Descripción
Clase encargada de redirigir las peticiones correspondientes a los cursos al CourseController según la URL pasada y el método HTTP utilizado.
Atributos Propuestos
Métodos Propuestos
Mismo número de métodos que el CourseController. Cada uno de estos llama al método correspondiente del controlador



<b>Nombre de la clase</b>
SurveyRoutes
<b>Descripción</b>
Clase encargada de redirigir las peticiones correspondientes a los cursos al SurveyController según la URL pasada y el método HTTP utilizado.
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
Mismo número de métodos que el SurveyController. Cada uno de estos llama al método correspondiente del controlador

<b>Nombre de la clase</b>
QuestionRoutes
<b>Descripción</b>
Clase encargada de redirigir las peticiones correspondientes a las preguntas al QuestionController según la URL pasada y el método HTTP utilizado.
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
Mismo número de métodos que el QuestionController. Cada uno de estos llama al método correspondiente del controlador

<b>Nombre de la clase</b>
GameRoutes
<b>Descripción</b>
Clase encargada de redirigir las peticiones correspondientes a los juegos al GameController según la URL pasada y el método HTTP utilizado.
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
Mismo número de métodos que el GameController. Cada uno de estos llama al método correspondiente del controlador

#### 6.5.2.1.3 Models

<b>Nombre de la clase</b>
UserModel
<b>Descripción</b>
Clase encargada de modelar la entidad usuario.



### Atributos Propuestos

**id:** identificar único del usuario en la base de datos. Generado automáticamente.

**username:** nombre del usuario.

**password:** contraseña del usuario. En todo momento se almacena su hash.

**role:** rol que desempeña el usuario, ya sea administrador, profesor o alumno.

### Métodos Propuestos

#### Nombre de la clase

CourseModel

#### Descripción

Clase que representa la entidad curso.

#### Atributos Propuestos

**id:** identificador único del curso en la base de datos. Se establece automáticamente.

**name:** nombre del curso.

**description:** descripción del curso

#### Métodos Propuestos

#### Nombre de la clase

QuestionModel

#### Descripción

Clase que representa la entidad pregunta

#### Atributos Propuestos

**id:** identificador único de la pregunta en la base de datos. Se establece automáticamente.

**title:** enunciado de la pregunta.

**subject:** tema de la pregunta.

**limit\_time:** tiempo límite para contestar la pregunta.

**type:** tipo de la pregunta. Ya sea multiopción, verdadero o falso o de respuesta corta.

**resource:** imagen asociada al enunciado de la pregunta.

**answers[]:** lista de respuestas asociadas a la pregunta.

#### Métodos Propuestos

#### Nombre de la clase

AnswerModel

#### Descripción

Clase que representa la entidad respuesta.

#### Atributos Propuestos



**id:** identificador único de la respuesta en la base de datos. Se genera automáticamente.

**description:** descripción de la respuesta.

**is\_correct:** indica si la respuesta es correcta o no.

**question\_id:** identificador de la pregunta a la que corresponde la pregunta.

#### Métodos Propuestos

#### Nombre de la clase

SurveyModel

#### Descripción

Clase que representa a la entidad cuestionario.

#### Atributos Propuestos

**id:** identificador único del cuestionario en la base de datos. Se genera automáticamente.

**title:** título del cuestionario.

**resource:** imagen asociada al cuestionario.

**questions[]:** lista de preguntas que componen el cuestionario.

#### Métodos Propuestos

#### Nombre de la clase

GameModel

#### Descripción

Clase que representa la entidad juego.

#### Atributos Propuestos

**id:** identificador único del juego en la base de datos. Se genera automáticamente.

**state:** indica el estado en el que se encuentra el juego. Puede ser creado, abierto, cerrado.

**survey\_id:** identificador del cuestionario asociado al juego.

**users[]:** listado de usuarios que han participado en el juego.

#### Métodos Propuestos

#### Nombre de la clase

UserResulModel

#### Descripción

Clase que representa la entidad resultado de usuario.

#### Atributos Propuestos

**user\_id:** identificador del usuario al que pertenece.

**game\_id:** identificador del juego al que pertenece.

**score:** puntuación obtenida por el usuario en el juego.



## Métodos Propuestos

### 6.5.2.1.4 Sockets

<b>Nombre de la clase</b>
SocketService
<b>Descripción</b>
Esta clase es la encargada de gestionar los sockets que permiten la sincronización entre profesores y alumnos para llevar a cabo un juego en línea.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Permite gestionar los sockets para:<ul style="list-style-type: none"><li>○ Jugar un cuestionario en línea.</li><li>○ Que los alumnos vean en tiempo real los cuestionarios abiertos.</li></ul></li></ul>
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
Métodos necesarios para manejar un juego online y mostrar a los usuarios los juegos abiertos en tiempo real.

### 6.5.2.2 Aplicación web

#### 6.5.2.2.1 Core

##### 6.5.2.2.1.1 Services

<b>Nombre de la clase</b>
APIAdminService
<b>Descripción</b>
Esta clase se encarga de contener el servicio que se encargue de realizar todas las llamadas a la API correspondientes al rol de usuario administrador.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Permite realizar las peticiones para:<ul style="list-style-type: none"><li>○ CRUD de usuarios.</li><li>○ CRUD de cursos.</li></ul></li></ul>
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
<b>createUser:</b> realiza la petición para crear un usuario. <b>getUsers:</b> obtiene los usuarios existentes en el sistema.



**updateUser:** realiza la petición correspondiente para actualizar los datos de un usuario.  
**deleteUser:** permite eliminar un usuario.  
**createCourse:** realiza la petición para crear un curso.  
**getCourses:** obtiene todos los cursos existentes en la base de datos.  
**updateCourse:** realiza la petición para actualizar los datos de un curso.  
**deleteCourse:** permite eliminar un curso.

#### Nombre de la clase

APIProfessorService

#### Descripción

Esta clase se encarga de contener el servicio que se encargue de realizar todas las llamadas a la API correspondientes al rol de usuario profesor.

#### Responsabilidades

- CRUD de preguntas.
- CRUD de cuestionarios.
- Asignar preguntas a cursos o eliminarlas.
- Asignar cuestionarios a cursos o eliminarlos.
- Asignar alumnos a cursos o eliminarlos.

#### Atributos Propuestos

#### Métodos Propuestos

**addUserToCourse:** realiza la petición para asignar un usuario a un curso.  
**addSurveyToCourse:** realiza la petición para asignar un cuestionario a un curso.  
**addQuestionToCourse:** realiza la petición para asignar una pregunta a un curso.  
**deleteUserFromCourse:** realiza la petición para eliminar la asignación de un usuario a un curso.  
**deleteSurveyFromCourse:** realiza la petición para eliminar la asignación de un cuestionario a un curso.  
**deleteQuestionFromCourse:** realiza la petición para eliminar la asignación de una pregunta a un curso.  
**createQuestion:** permite realizar una petición para crear una pregunta.  
**getQuestionsByCreator:** obtiene todas las preguntas creadas por un usuario.  
**updateQuestions:** realiza una petición para actualizar una pregunta.  
**deleteQuestion:** realiza una petición para eliminar una pregunta.  
**createSurvey:** permite realizar una petición para crear un cuestionario.  
**getSurveysByUser:** obtiene todos los cuestionarios creados por un usuario.  
**updateSurvey:** realiza una petición para actualizar un cuestionario.  
**deleteSurvey:** realiza una petición para eliminar un cuestionario.  
**generateResults:** realiza una petición para crear resultados de los usuarios que hayan jugado un juego.  
**updateGame:** permite realizar una petición para actualizar un juego.



<b>Nombre de la clase</b>
APIStudentService
<b>Descripción</b>
Esta clase se encarga de contener el servicio que se encargue de realizar todas las llamadas a la API correspondientes al rol de usuario alumno.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Permite realizar las peticiones para:<ul style="list-style-type: none"><li>○ Obtener juegos por id.</li><li>○ Obtener los juegos abiertos de los cursos a los que está asignado.</li><li>○ Obtener los resultados de los juegos en los que haya participado.</li></ul></li></ul>
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
<b>getGameById:</b> obtiene un juego dado su identificador. <b>getOpenGamesByCourse:</b> obtiene todos los juegos abiertos pertenecientes a los cursos en los que está asignado el alumno. <b>getResultsByUser:</b> obtiene los resultados de un determinado usuario.

<b>Nombre de la clase</b>
AuthService
<b>Descripción</b>
Esta clase se encarga de contener el servicio que se encargue de realizar el inicio de sesión de la aplicación.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Permite iniciar sesión en la aplicación.</li></ul>
<b>Atributos Propuestos</b>
<b>user:</b> usuario identificado en el sistema.
<b>Métodos Propuestos</b>
<b>login:</b> realiza la petición para que un usuario inicie sesión en el sistema.

<b>Nombre de la clase</b>
SocketService
<b>Descripción</b>
Esta clase se encarga de contener el servicio que se encargue de contener los servicios que se comuniquen con el servidor mediante sockets.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Conectarse a un socket.</li></ul>



- Ejecutar juego en línea.
- Mostrar en tiempo real a un alumno los juegos abiertos.

#### Atributos Propuestos

#### Métodos Propuestos

**joinToSocket:** conecta al cliente web a un socket.

**createGame:** permite al profesor crear un juego, reflejando el resultado en cada una de las pantallas de los alumnos correspondientes.

**startGame:** permite empezar un juego.

**closeGame:** permite dar fin a un juego.

#### 6.5.2.2.1.2 Model

Las clases son idénticas al modelo especificado en el subsistema anterior.

#### 6.5.2.2.2 Components

Nombre de la clase
Login
Descripción
Contiene la vista con el formulario que utiliza el usuario para iniciar sesión en el sistema.
Responsabilidades
<ul style="list-style-type: none"><li>• Permite al usuario introducir sus credenciales de acceso y validarlos.</li></ul>
Atributos Propuestos
Métodos Propuestos
<b>login:</b> recoge las credenciales y se las envía al servicio de login.

#### 6.5.2.2.2.1 Admin

Nombre de la clase
UserList
Descripción
Esta clase contiene la vista que muestra la lista de usuarios del sistema al usuario administrador.
Responsabilidades
<ul style="list-style-type: none"><li>• Permite que el usuario vea todos los usuarios existentes.</li><li>• Permite al usuario administrador moverse a la pantalla de creación de usuarios.</li><li>• Permite al usuario administrador eliminar usuarios.</li></ul>
Atributos Propuestos
<b>userList:</b> lista que contiene todos los usuarios, resultado de realizar la petición correspondiente con el servicio APIAdminService.





Métodos Propuestos
<p><b>createUser:</b> navega al usuario hasta la página de creación de usuarios.</p> <p><b>deleteUser:</b> permite al administrador eliminar cualquier usuario.</p>

Nombre de la clase
CreateUserForm
Descripción
Esta clase contiene la vista que permite al usuario introducir los datos de un nuevo usuario y llamar al servicio para crearlo.
Responsabilidades
<ul style="list-style-type: none"> <li>Permite al usuario introducir los datos de un nuevo usuario y realizar la llamada para crearlo.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<p><b>createUser:</b> recoge los datos introducidos por el usuario en el formulario y llama al servicio pasándole los datos para crear un usuario.</p>

Nombre de la clase
CourseList
Descripción
Esta clase se contener la vista que muestra la lista de cursos del sistema al usuario administrador.
Responsabilidades
<ul style="list-style-type: none"> <li>Permite que el usuario vea todos los cursos existentes.</li> <li>Permite al usuario administrador moverse a la pantalla de creación de cursos.</li> <li>Permite al usuario administrador eliminar cursos.</li> </ul>
Atributos Propuestos
<p><b>courseList:</b> lista que contiene todos los cursos, resultado de realizar la petición correspondiente con el servicio APIAdminService.</p>
Métodos Propuestos
<p><b>createCourse:</b> navega al usuario hasta la página de creación de cursos.</p> <p><b>deleteCourse:</b> permite al administrador eliminar cualquier curso.</p>

Nombre de la clase
CreateCourseForm
Descripción
Esta clase contiene la vista que permite al usuario introducir los datos de un nuevo curso y llamar al servicio para crearlo.



<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>Permite al usuario introducir los datos de un nuevo curso y realizar la llamada para crearlo.</li></ul>
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
<b>createCourse:</b> recoge los datos introducidos por el usuario en el formulario y llama al servicio pasándole los datos para crear un curso.

<b>Nombre de la clase</b>
UpdateUserForm
<b>Descripción</b>
Esta clase contiene la vista que permite al usuario modificar un usuario ya existente.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>Permite al usuario introducir los datos de un usuario existente y realizar la llamada para modificarlo.</li></ul>
<b>Atributos Propuestos</b>
<b>user:</b> usuario que está siendo modificado.
<b>Métodos Propuestos</b>
<b>updateUser:</b> recoge los datos modificados por el usuario y utiliza el servicio para actualizar el usuario.

<b>Nombre de la clase</b>
UpdateCourseForm
<b>Descripción</b>
Esta clase contiene la vista que permite al usuario modificar un curso ya existente.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>Permite al usuario introducir los datos de un curso existente y realizar la llamada para modificarlo.</li></ul>
<b>Atributos Propuestos</b>
<b>course:</b> curso que está siendo modificado.
<b>Métodos Propuestos</b>
<b>updateCourse:</b> recoge los datos modificados por el usuario y utiliza el servicio para actualizar el curso.

#### 6.5.2.2.2 Professor

<b>Nombre de la clase</b>
Library
<b>Descripción</b>



Esta clase contiene la vista principal del usuario profesor. Permite ver todos los cursos y acceder a ellos, así como todos los cuestionarios y preguntas creadas por él. Además, redirige al usuario a las pantallas de creación y modificación de cuestionarios y cursos y también puede eliminarlos.

**Responsabilidades**

- Mostrar todos los cursos y dar acceso a la pantalla de curso.
- Mostar todos los cuestionarios creados por el usuario, eliminarlos y navegar al usuario a la pantalla de creación y modificación de cuestionario.
- Mostar todas las preguntas creadas por el usuario, eliminarlas y navegar al usuario a la pantalla de creación y modificación de preguntas.

**Atributos Propuestos**

**courseList:** lista que contiene todos los cursos.

**surveyList:** lista que contiene todos los cuestionarios creados por el usuario.

**questionList:** lista que contiene todas las preguntas creadas por el usuario.

**Métodos Propuestos**

**createQuestion:** navega al usuario a la pantalla de creación de preguntas.

**createSurvey:** navega al usuario a la pantalla de creación de preguntas.

**editQuestion:** navega al usuario a la pantalla de modificación de preguntas.

**editSurvey:** navega al usuario a la pantalla de modificación de cuestionarios.

**deleteQuestions:** llama al servicio correspondiente para eliminar una pregunta.

**deleteSurvey:** llama al servicio correspondiente para eliminar un cuestionario.

**Nombre de la clase**

CreateQuestionForm

**Descripción**

Esta clase contiene la vista que permite al usuario profesor introducir los datos de una nueva pregunta y sus respuestas asociadas y llamar al servicio correspondiente para crearla.

**Responsabilidades**

- Permitir al usuario introducir los datos de una nueva pregunta.
- Llamar al servicio de crear pregunta pasándole los datos introducidos por el usuario.

**Atributos Propuestos**

**file:** imagen subida por el usuario.

**Métodos Propuestos**

**createQuestion:** recoge los datos introducidos por el usuario y se los pasa al servicio para crear la pregunta.

**uploadFile:** permite al usuario subir a la aplicación la imagen asociada a la pregunta.

**Nombre de la clase**

CreateSurveyForm

**Descripción**



Esta clase contiene la vista que permite al usuario profesor introducir los datos de un nuevo cuestionario y llamar al servicio correspondiente para crearlo.

#### Responsabilidades

- Permitir al usuario introducir los datos de un nuevo cuestionario.
- Llamar al servicio de crear cuestionario pasándole los datos introducidos por el usuario.

#### Atributos Propuestos

#### Métodos Propuestos

**createSurvey:** recoge los datos introducidos por el usuario y se los pasa al servicio para crear el cuestionario.

**uploadFile:** permite al usuario subir una imagen a la aplicación.

#### Nombre de la clase

UpdateQuestionForm

#### Descripción

Esta clase contiene la vista que permite al usuario profesor modificar los datos de una pregunta ya creada y realizar la llamada al servicio para guardar los cambios.

#### Responsabilidades

- Permitir al usuario modificar los datos de una pregunta y realizar la llamada al servicio.

#### Atributos Propuestos

**question:** pregunta que se quiere modificar.

**file:** imagen subida a la aplicación.

#### Métodos Propuestos

**updateQuestion:** recoge los datos introducidos por el usuario y se los pasa al servicio.

**uploadFile:** permite al usuario subir una imagen a la aplicación.

#### Nombre de la clase

UpdateSurveyForm

#### Descripción

Esta clase contiene la vista que permite al usuario profesor modificar los datos de un cuestionario ya creado y realizar la llamada al servicio para guardar los cambios.

#### Responsabilidades

- Permitir al usuario modificar los datos de un cuestionario y realizar la llamada al servicio.

#### Atributos Propuestos

**survey:** cuestionario que se quiere modificar.

**file:** imagen subida a la aplicación.

#### Métodos Propuestos

**updateSurvey:** recoge los datos introducidos por el usuario y se los pasa al servicio.

**uploadFile:** permite al usuario subir una imagen a la aplicación.



<b>Nombre de la clase</b>
CoursePage
<b>Descripción</b>
Esta clase contiene la vista de detalle de un curso. Muestra al usuario la descripción del curso, así como los cuestionarios, preguntas y alumnos asociados al mismo. Desde esta pantalla se lanzan los cuestionarios en línea.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Muestra al usuario los detalles del curso.<ul style="list-style-type: none"><li>○ Descripción.</li><li>○ Cuestionarios, preguntas y alumnos asociados al curso.</li></ul></li><li>• Permite lanzar cualquier cuestionario asociado al curso.</li></ul>
<b>Atributos Propuestos</b>
<b>course:</b> contiene los datos del curso en el que se encuentra el usuario. <b>surveyList:</b> lista que contiene los cuestionarios asociados al curso. <b>questionList:</b> lista que contiene las preguntas asociadas al curso. <b>userList:</b> lista que contiene los usuarios asociados al curso.
<b>Métodos Propuestos</b>
<b>launchGame:</b> permite al usuario lanzar un juego y pasar a jugarlo en su pantalla correspondiente. <b>addSurvey:</b> llama al servicio para asociar un cuestionario al curso. <b>addQuestion:</b> llama al servicio para asociar una pregunta al curso. <b>addUser:</b> llama al servicio para asociar un alumno al curso. <b>removeSurvey:</b> llama al servicio para eliminar la asociación de un cuestionario al curso. <b>removeQuestion:</b> llama al servicio para eliminar la asociación de una pregunta al curso. <b>removeUser:</b> llama al servicio para eliminar la asociación de un alumno al curso.

<b>Nombre de la clase</b>
GamePage
<b>Descripción</b>
Esta clase contiene la vista donde se desarrolla el cuestionario en línea por parte del usuario profesor.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Permite jugar en línea el cuestionario previamente lanzado..</li></ul>
<b>Atributos Propuestos</b>
<b>game:</b> juego que se está jugando actualmente.
<b>Métodos Propuestos</b>



### 6.5.2.2.2.3 Student

<b>Nombre de la clase</b>
HomePage
<b>Descripción</b>
Esta clase contiene la vista principal del usuario alumno. Muestra los cursos a los que pertenece el usuario y los cuestionarios abiertos que le alumno podrá pasar a jugar.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Muestra los cursos a los que pertenece el usuario y permite navegar hasta la pantalla del curso.</li><li>• Muestra los cuestionarios abiertos y permite navegar hasta la pantalla de juego.</li></ul>
<b>Atributos Propuestos</b>
<b>courseList:</b> lista que contiene los cursos a los que pertenece el usuario. <b>openGamesList:</b> lista que contiene todos los juegos abiertos en los que puede participar el usuario.
<b>Métodos Propuestos</b>

<b>Nombre de la clase</b>
ProfilePage
<b>Descripción</b>
Esta clase contiene la vista del perfil del usuario. En ella, podrá modificar su contraseña.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Permite al usuario modificar su contraseña actual.</li></ul>
<b>Atributos Propuestos</b>
<b>Métodos Propuestos</b>
<b>modifyPassword:</b> recoge la nueva contraseña introducida por el usuario y llama al servicio para guardar los cambios.

<b>Nombre de la clase</b>
CoursePage
<b>Descripción</b>
Esta clase contiene la vista de un curso para el usuario alumno. Podrá consultar los cuestionarios abiertos para ese curso y los resultados de los juegos en los que haya participado.
<b>Responsabilidades</b>
<ul style="list-style-type: none"><li>• Mostrar cuestionarios abiertos para ese curso.</li><li>• Mostrar los resultados para los juegos en los que haya participado el alumno.</li></ul>
<b>Atributos Propuestos</b>
<b>course:</b> contiene los datos del curso en el que se encuentra el usuario.



**userResults:** contiene todos los resultados del usuario de los cuestionarios realizados en ese curso.

**Métodos Propuestos**

**Nombre de la clase**

GamePage

**Descripción**

Esta clase contiene la vista donde se desarrolla el cuestionario en línea por parte del usuario alumno.

**Responsabilidades**

- Permite jugar en línea el cuestionario en línea lanzado por un usuario profesor.

**Atributos Propuestos**

**game:** juego que se está jugando actualmente.

**Métodos Propuestos**

### 6.5.2.2.3 Modules

**Nombre de la clase**

AppModule

**Descripción**

Es un contenedor que define la aplicación web. Toda aplicación hecha con Angular debe tener al menos un módulo de aplicación.

**Responsabilidades**

- Coordina todos los servicios y componentes que se crean en la aplicación.

**Atributos Propuestos**

**Métodos Propuestos**

**Nombre de la clase**

RoutingModule

**Descripción**

Es un módulo de Angular donde se especifican las rutas de cada uno de los componentes que componen la aplicación.

**Atributos Propuestos**

**Métodos Propuestos**

## 6.6 ASI 8: DEFINICIÓN DE INTERFACES DE USUARIO

Una vez realizado todos los análisis previos, se procede a identificar cómo debe de ser la interfaz de usuario, desarrollando una serie de *mockups*, resultado de unos previos bocetos en papel. La importancia de este apartado radica en que la interfaz de usuario es el medio principal mediante el cual el usuario interactúa con el sistema, por ello, esta debe de ser fácil de usar e intuitiva. A continuación, se mostrarán los *mockups* de cada una de las pantallas que componen la aplicación web. Los *mockups* han sido realizados con la herramienta **MockFlow**.

### 6.6.1 Descripción de la Interfaz

#### 6.6.1.1 Pantalla de inicio de sesión

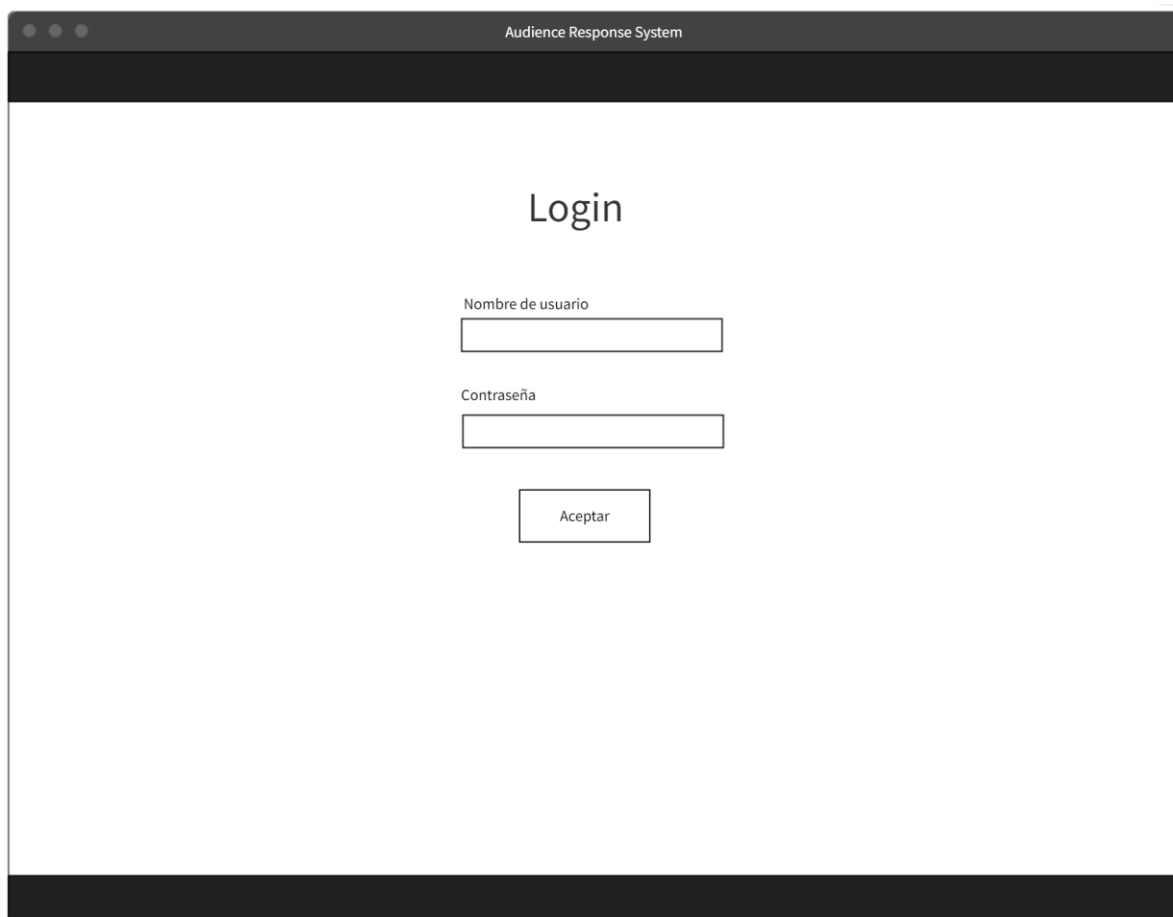
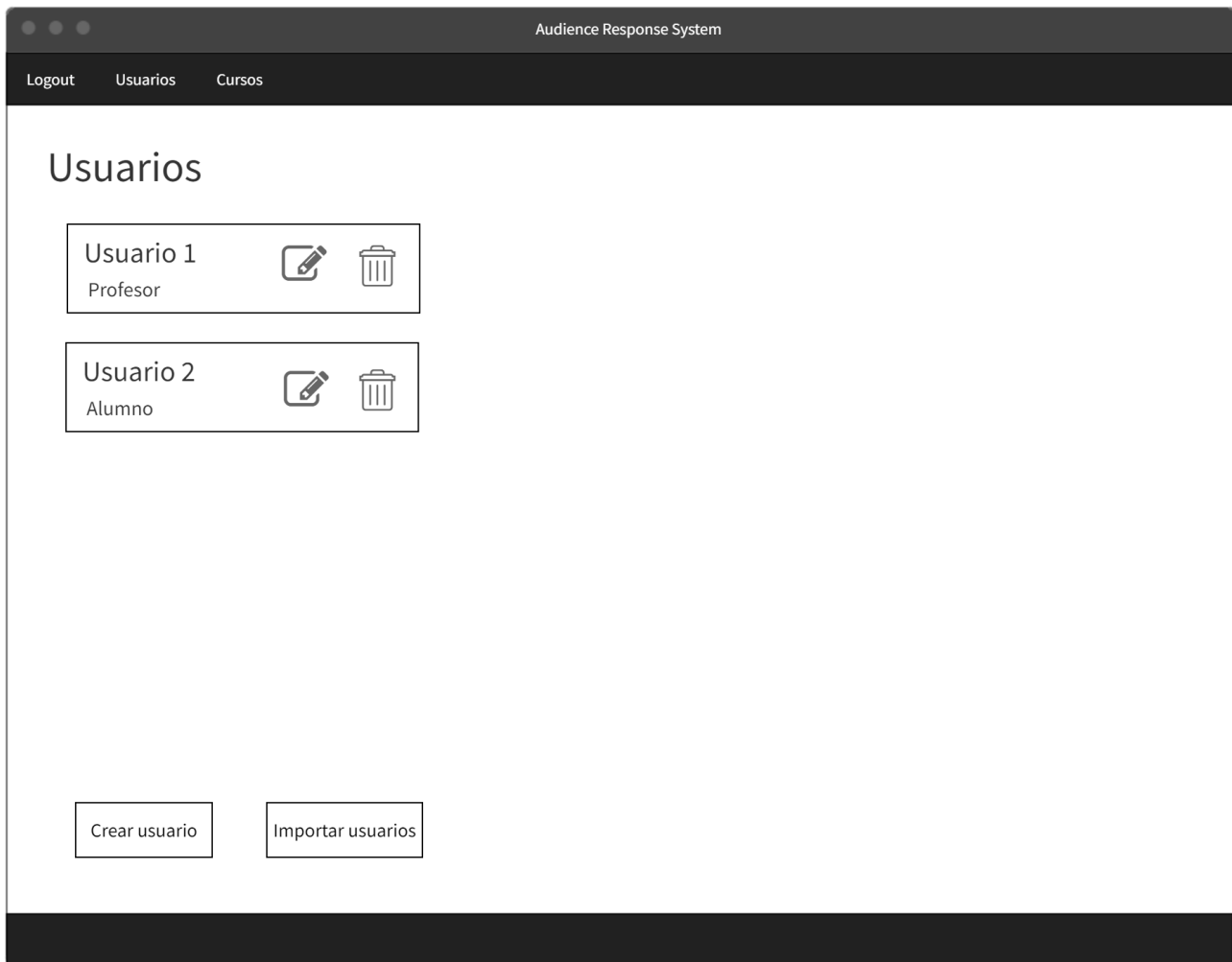


Ilustración de la pantalla de inicio de sesión (Login) del sistema "Audience Response System". La interfaz muestra un título "Login" centrado, seguido de dos campos de entrada: "Nombre de usuario" y "Contraseña". Debajo de los campos se encuentra un botón "Aceptar".

*Ilustración 24 MOCKUP DE LA PANTALLA DE INICIO DE SESIÓN*



### 6.6.1.2 Pantalla de lista de usuarios

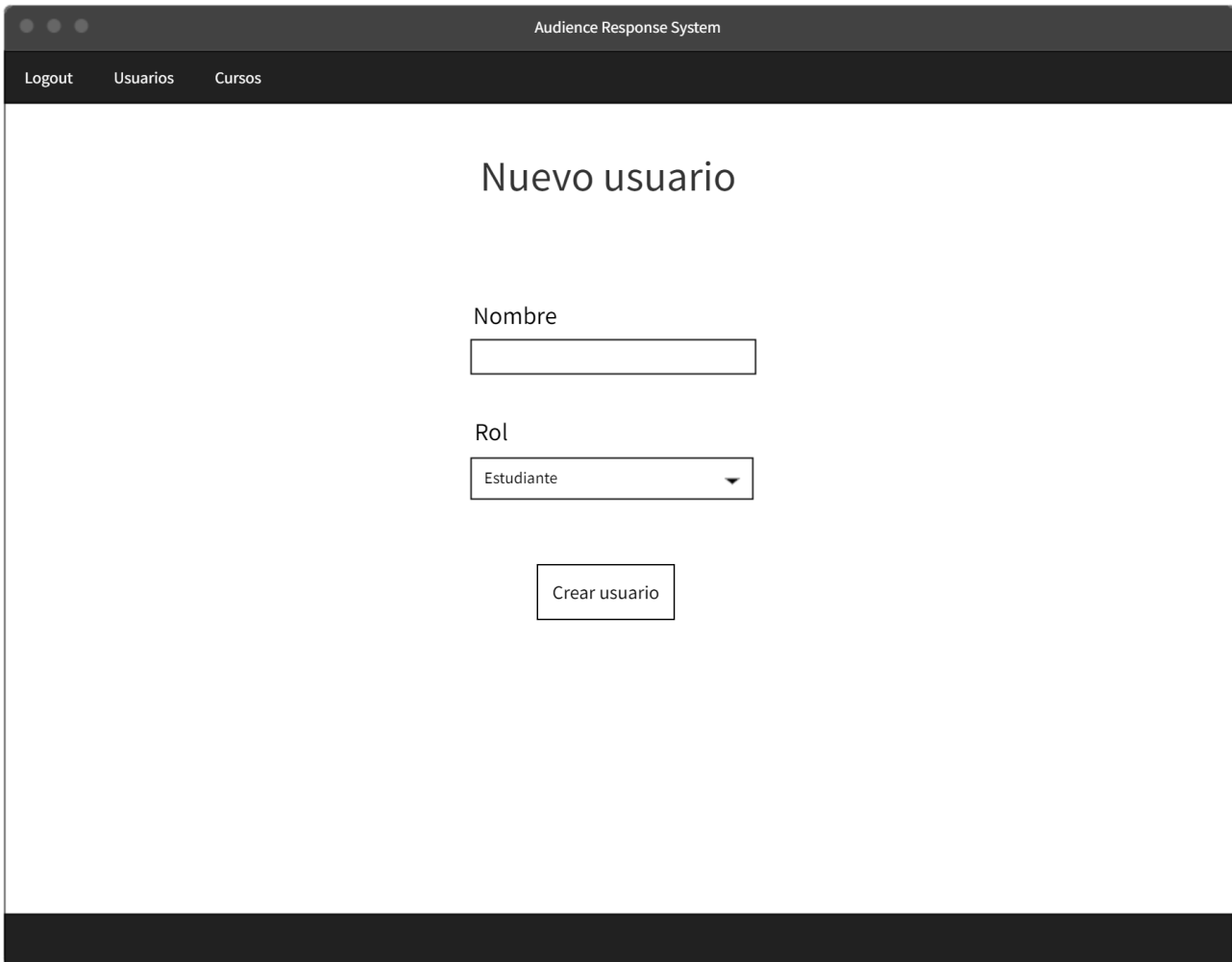


**Ilustración 25 MOCKUP DE LA PANTALLA DE LISTADO DE USUARIOS**

### 6.6.1.3 Pantalla de lista de cursos

Es idéntica a la anterior, cambiando solamente usuarios por cursos.

#### 6.6.1.4 Pantalla de creación de usuarios



Audience Response System

Logout Usuarios Cursos

## Nuevo usuario

Nombre

Rol

Estudiante ▼

Crear usuario

**Ilustración 26 MOCKUP DE LA PANTALLA DE CREACIÓN DE USUARIOS**

#### 6.6.1.5 Pantalla de creación de cursos

Idéntica a la pantalla de creación de usuarios, sustituyendo el campo de rol por un campo descripción.

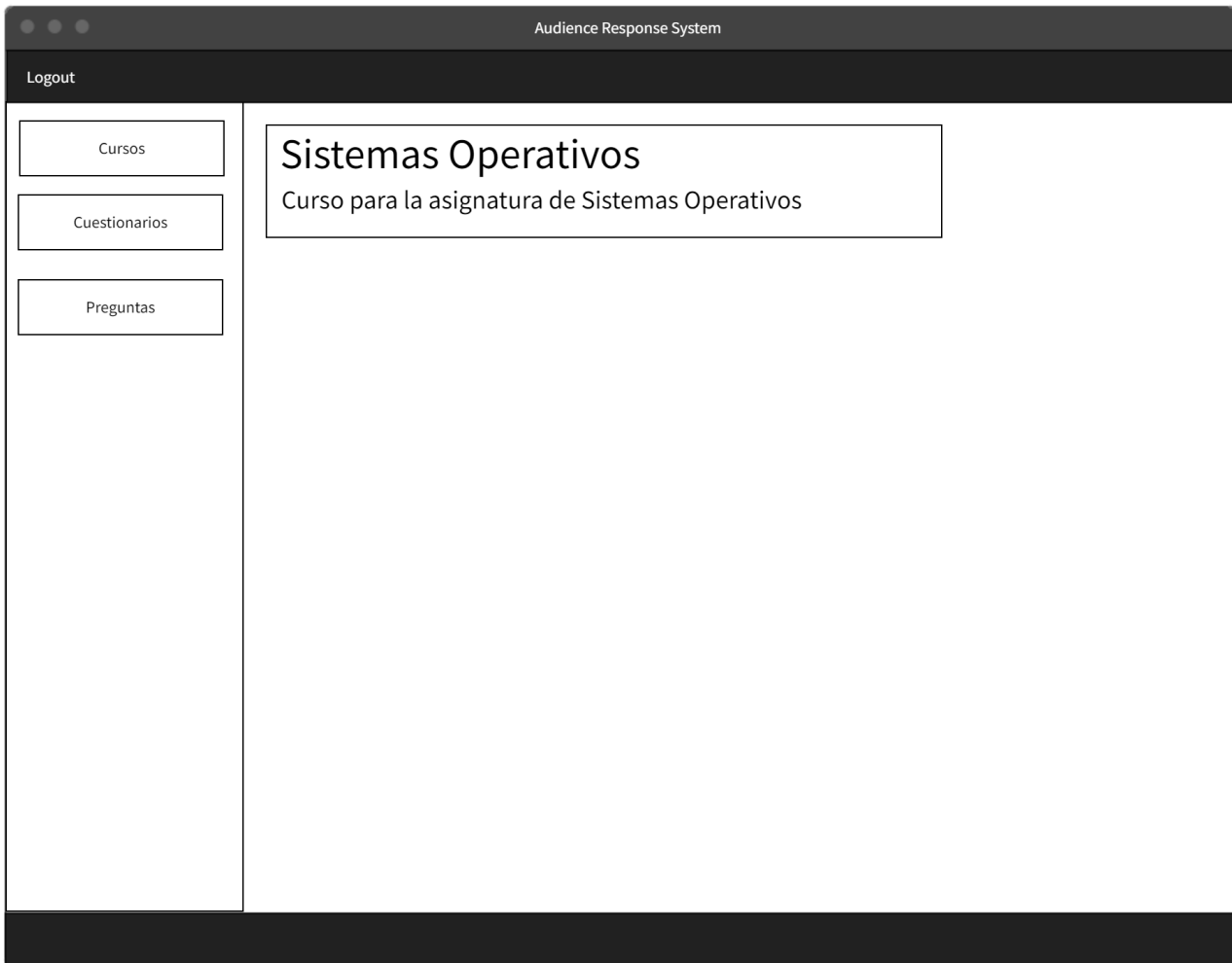
#### 6.6.1.6 Pantalla de modificación de usuarios

Idéntica a la pantalla de creación de usuarios.

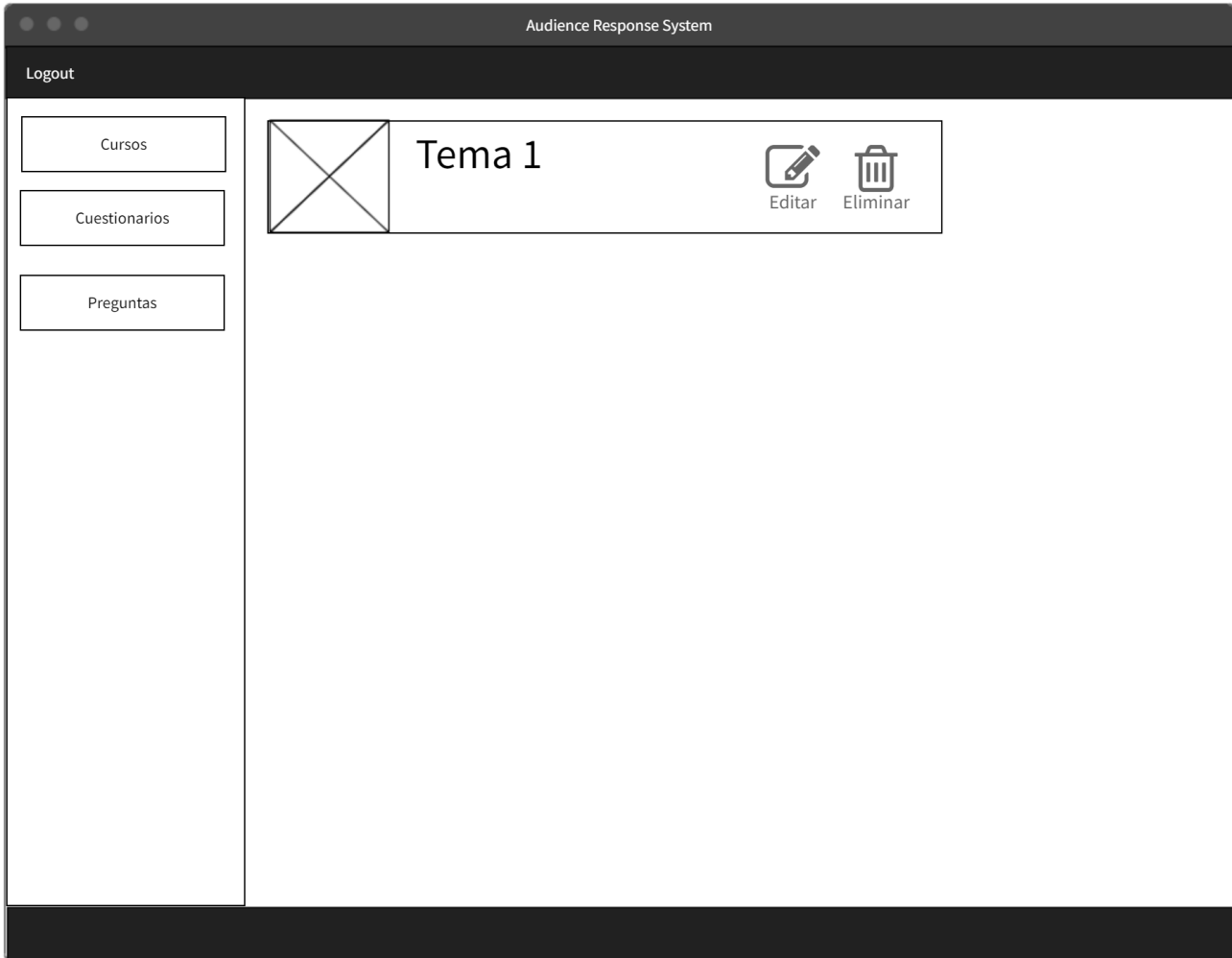
#### 6.6.1.7 Pantalla de modificación de cursos

Idéntica a la pantalla de creación de cursos.

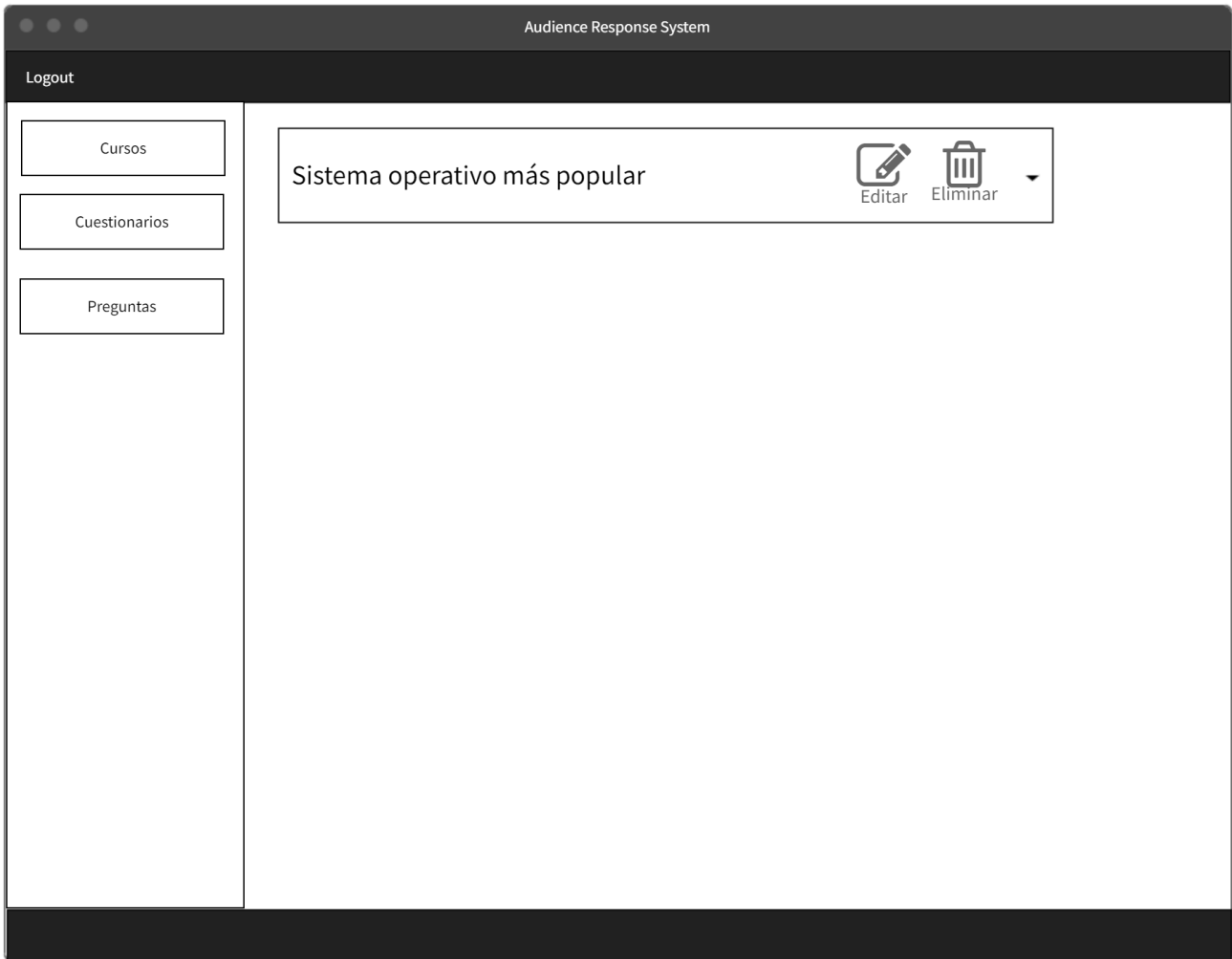
### 6.6.1.8 Pantalla de la biblioteca



*Ilustración 27 MOCKUP DE LA PANTALLA DE BIBLIOTECA (CURSOS)*

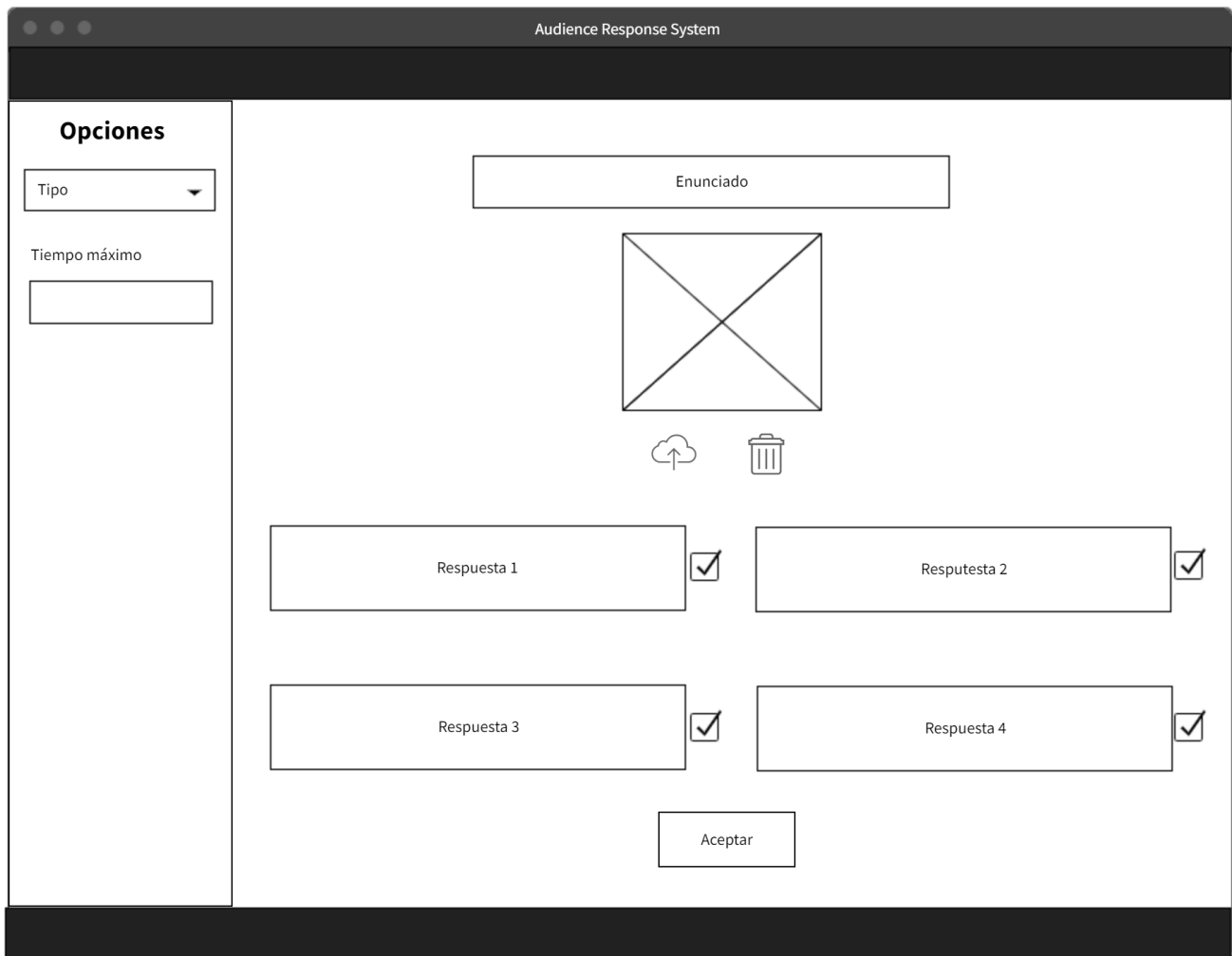


**Ilustración 28 MOCKUP DE LA PANTALLA DE LA BIBLIOTECA (CUESTIONARIOS)**



*Ilustración 29 MOCKUP DE LA PANTALLA DE LA BIBLIOTECA (PREGUNTAS)*

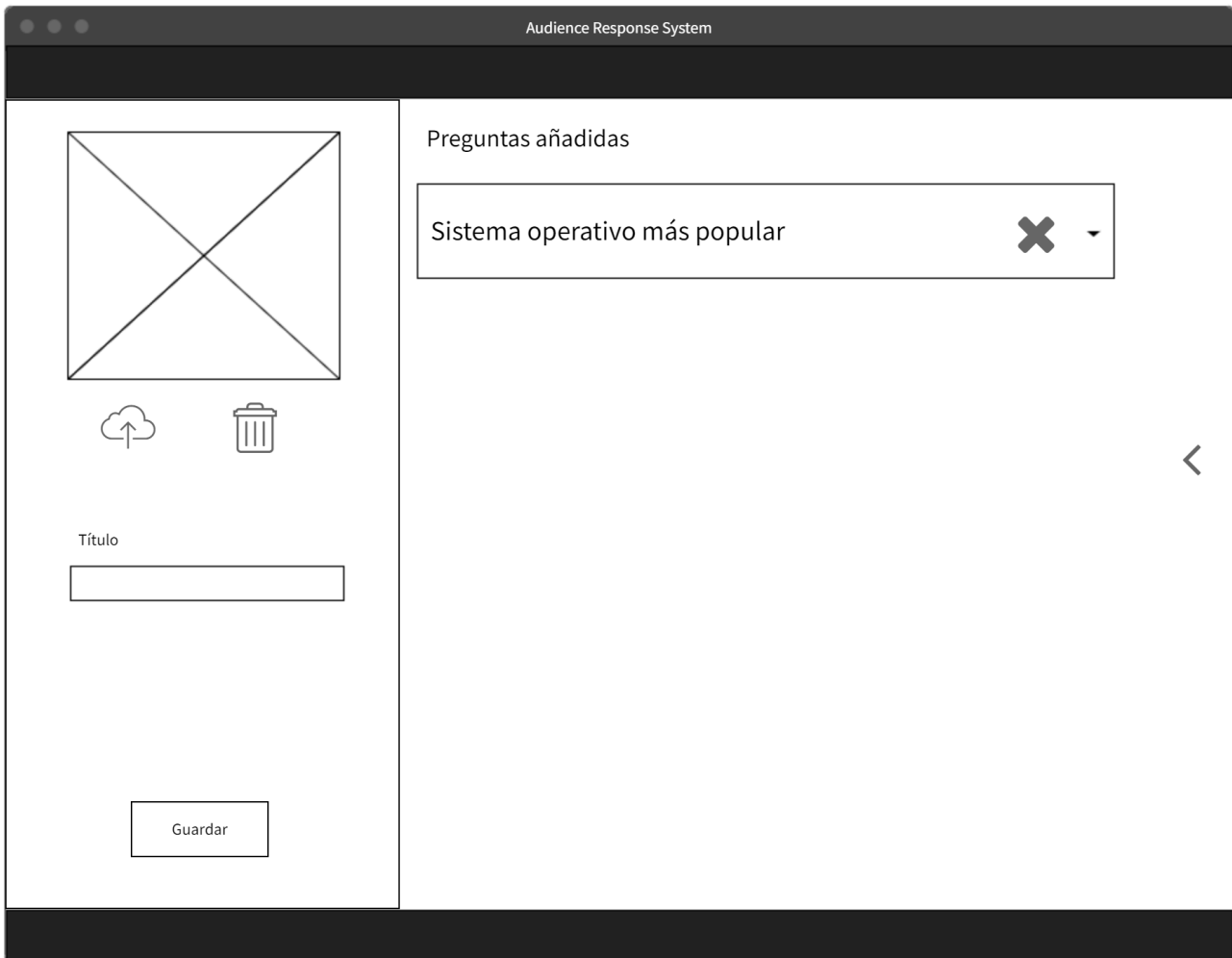
### 6.6.1.9 Pantalla de creación de preguntas



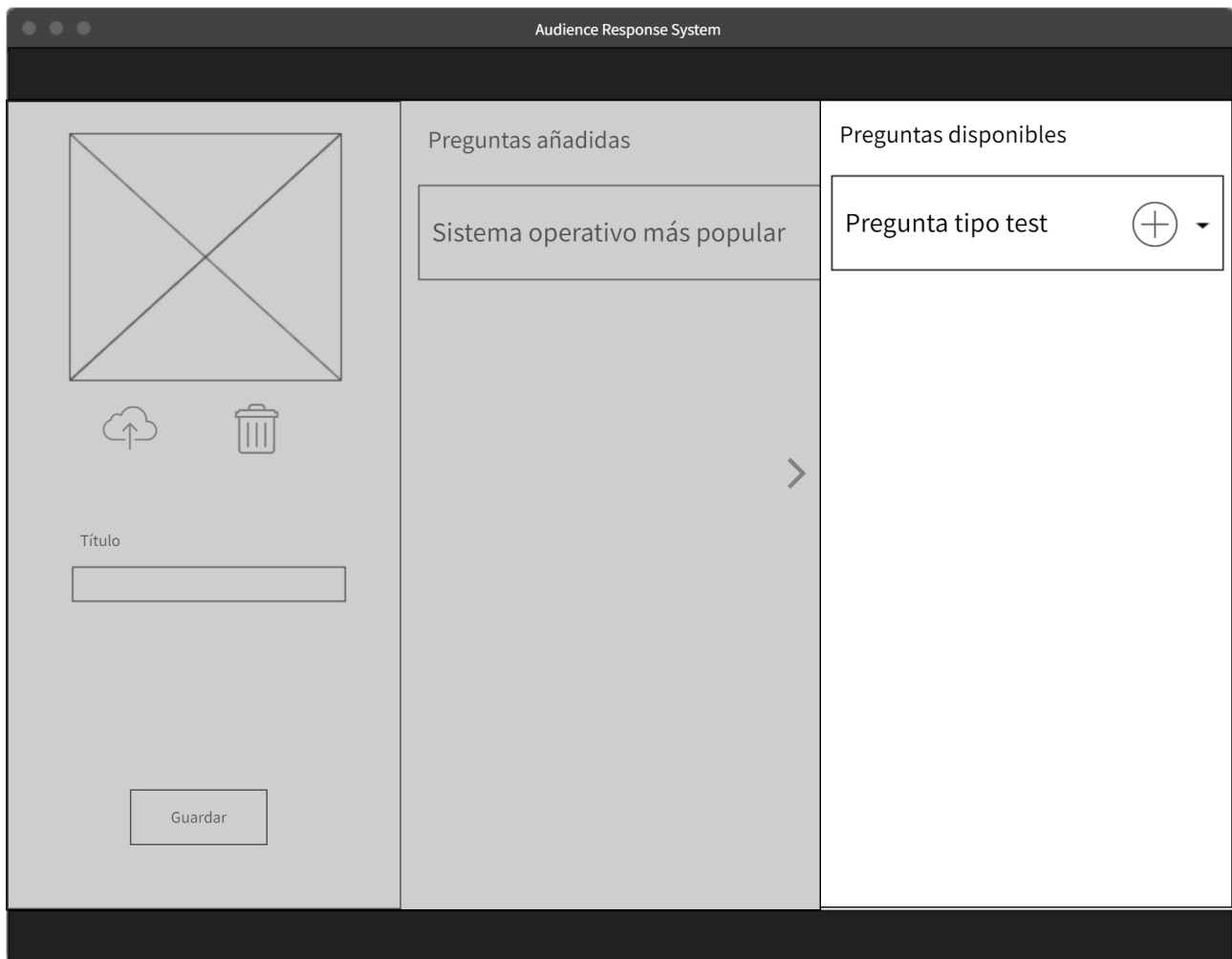
The screenshot shows a web application window titled "Audience Response System". On the left, there is a sidebar with the heading "Opciones" (Options). It contains a "Tipo" dropdown menu, a "Tiempo máximo" (Maximum time) input field, and a "Aceptar" (Accept) button. The main area is for creating a question. It features a text input field labeled "Enunciado" (Statement). Below it is a large square placeholder with a diagonal 'X', indicating where an image can be uploaded. Underneath the placeholder are icons for uploading a file (cloud with arrow) and deleting a file (trash). Below these are four text input fields for answers, labeled "Respuesta 1", "Resputesta 2", "Respuesta 3", and "Respuesta 4". Each answer field has a checkmark icon to its right, indicating that the answer is required. At the bottom center of the main area is a "Aceptar" (Accept) button.

**Ilustración 30 MOCKUP DE LA PANTALLA DE CREACIÓN DE PREGUNTAS**

### 6.6.1.10 Pantalla de creación de cuestionario



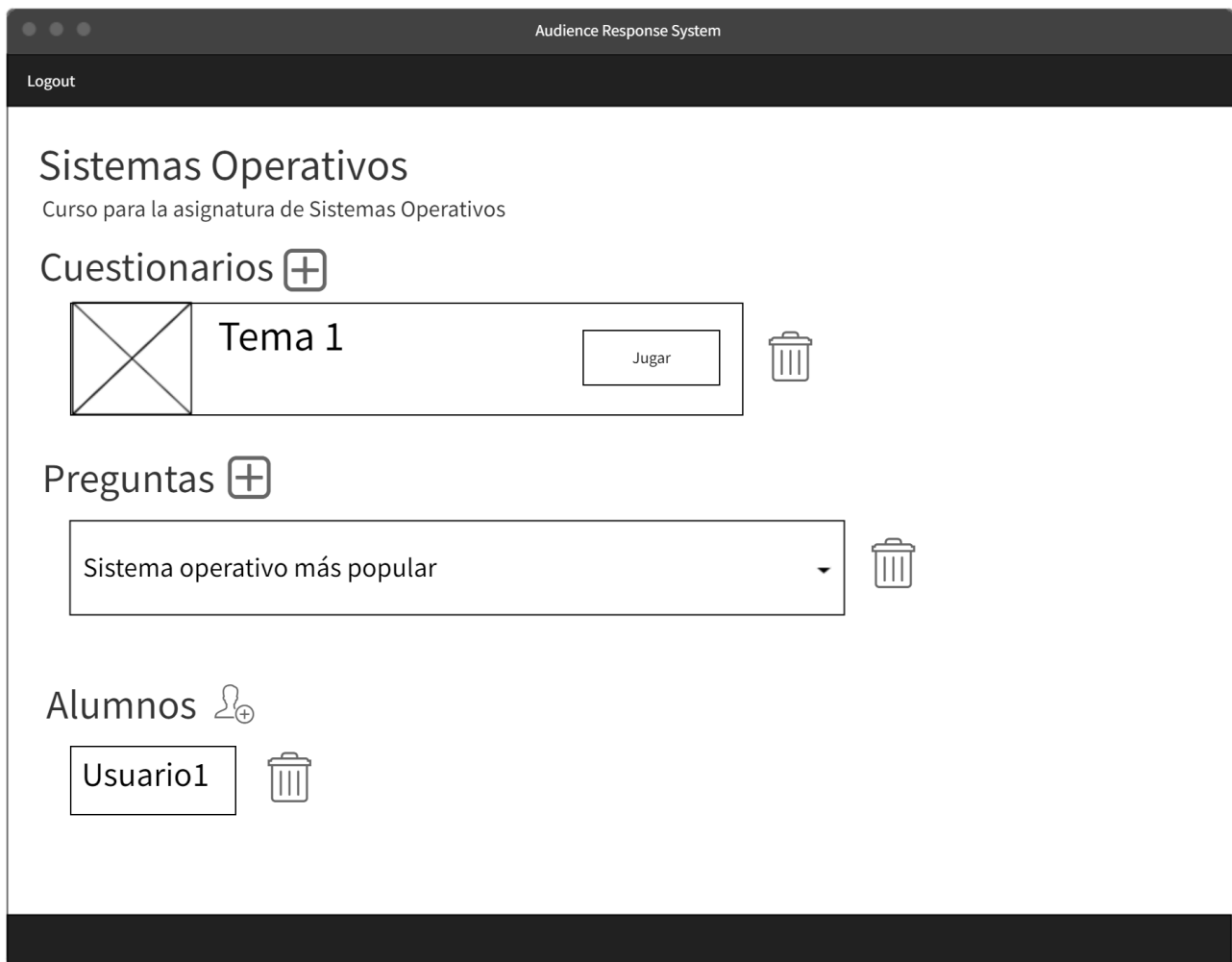
**Ilustración 31 MOCKUP DE LA PANTALLA DE CREACIÓN DE CUESTIONARIO 1**



**Ilustración 32 MOCKUP DE LA PANTALLA DE CREACIÓN DE CUESTIONARIO 2**



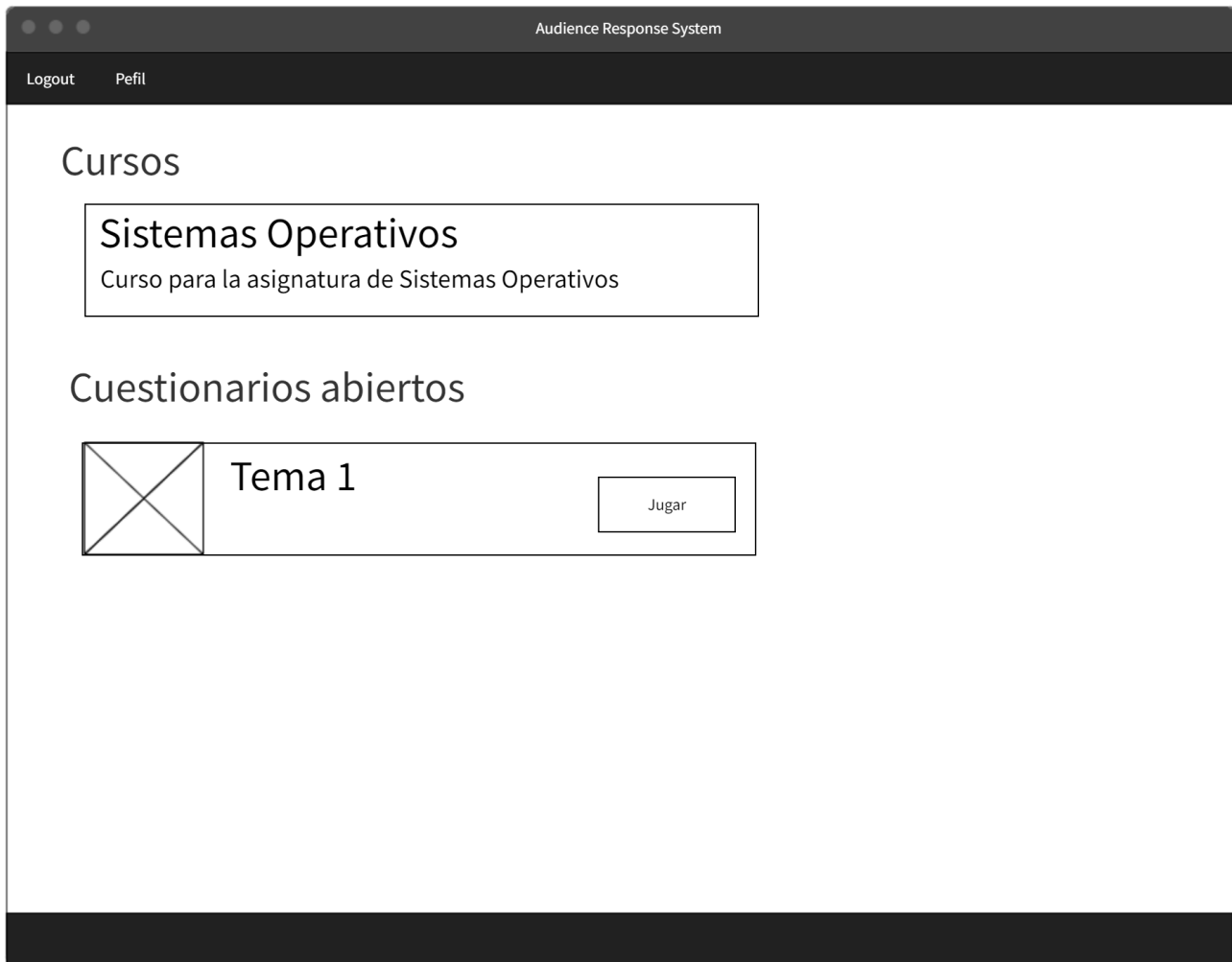
### 6.6.1.11 Pantalla de detalles de un curso (profesor)



*Ilustración 33 MOCKUP DE LA PANTALLA DE DETALLES DE UN CURSO (PROFESOR)*

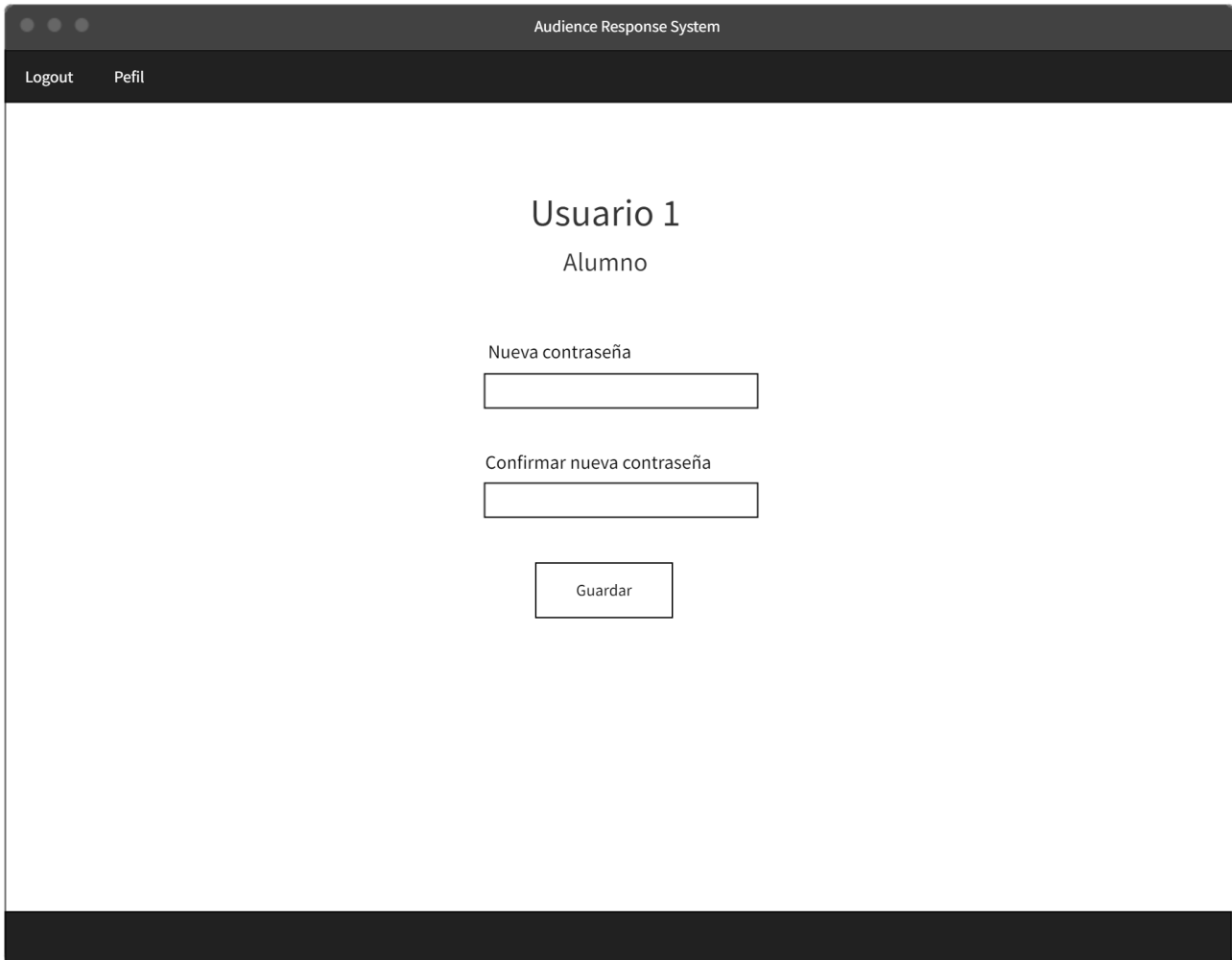


### 6.6.1.12 *Pantalla principal del alumno*



*Ilustración 34 MOCKUP DE LA PANTALLA PRINCIPAL DEL ALUMNO*

### 6.6.1.13 Pantalla de perfil del alumno



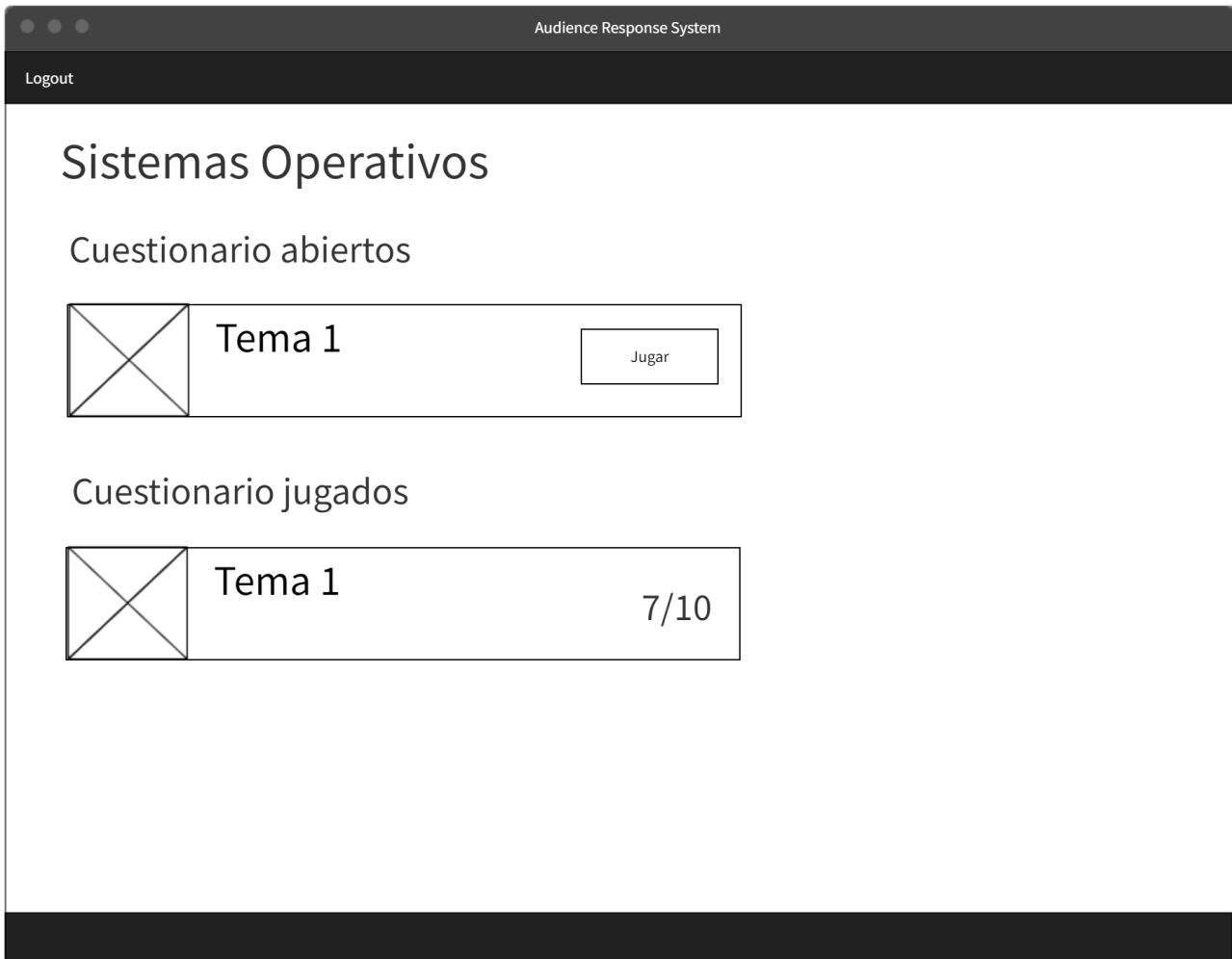
The screenshot shows a web browser window titled "Audience Response System". The browser's address bar and navigation buttons are visible. The page has a dark header with "Logout" and "Perfil" links. The main content area is white and contains the following elements:

- Header: "Audience Response System"
- Navigation: "Logout", "Perfil"
- User Information: "Usuario 1", "Alumno"
- Form Fields:
  - "Nueva contraseña" with an input field
  - "Confirmar nueva contraseña" with an input field
- Action: "Guardar" button

*Ilustración 35 MOCKUP DE PANTALLA DE PERFIL*

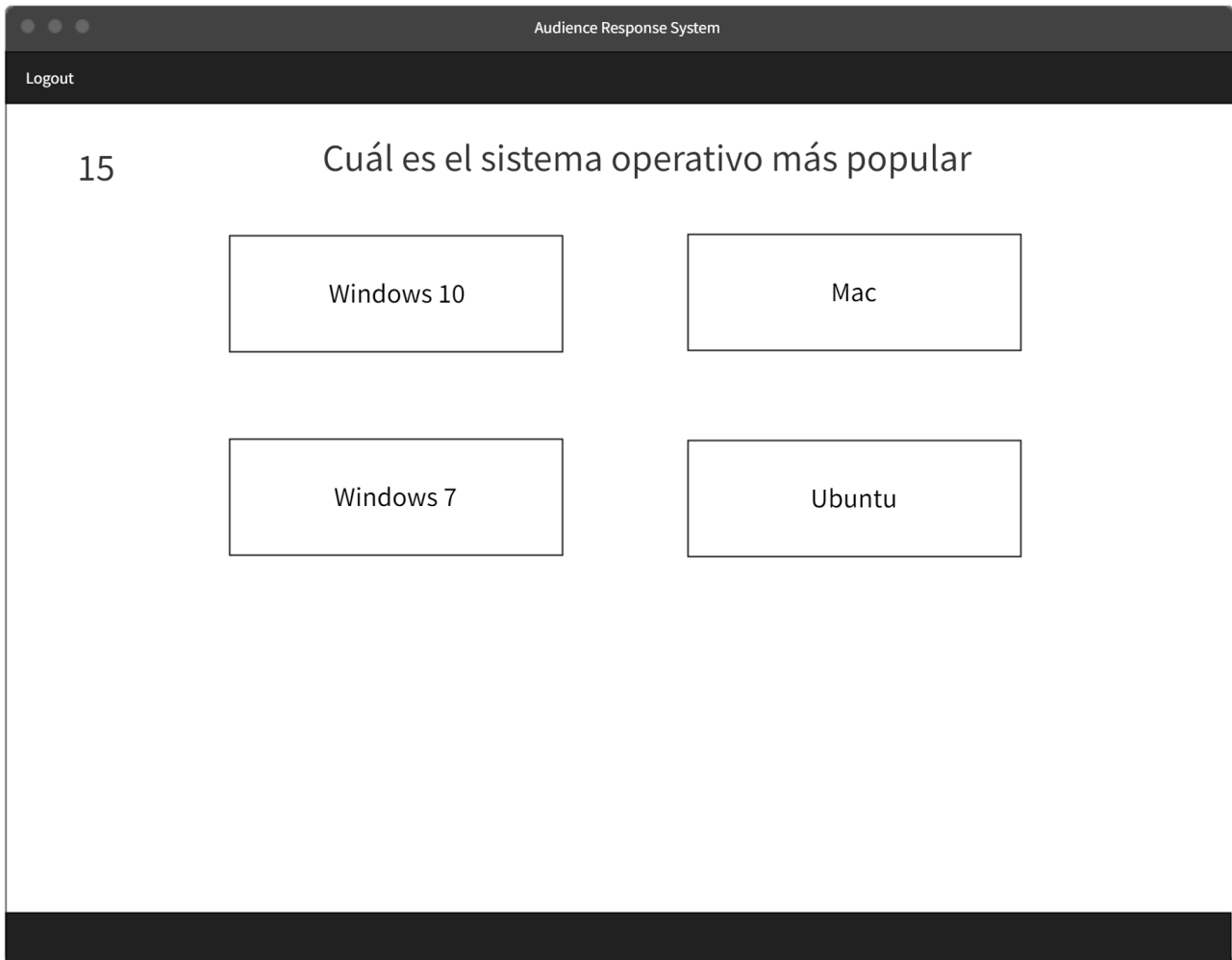


### 6.6.1.14 Pantalla de detalles de un curso (alumno)



*Ilustración 36 MOCKUP DE LA PANTALLA E DETALLES DE UN CURSO (ALUMNO)*

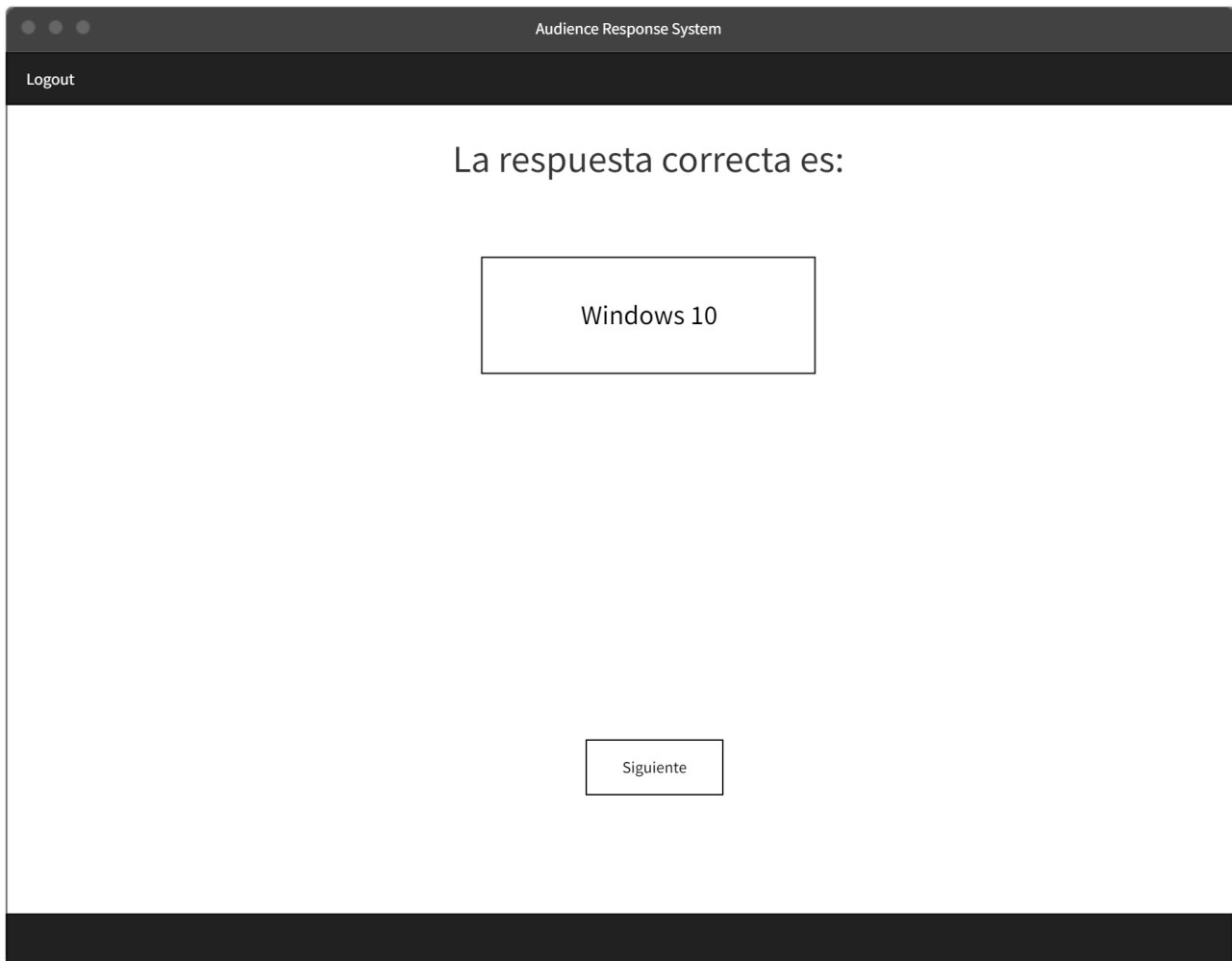
### 6.6.1.15 Pantalla de juego (mostrando pregunta)



*Ilustración 37 MOCKUP DE LA PANTALLA DE JUEGO (PREGUNTA)*

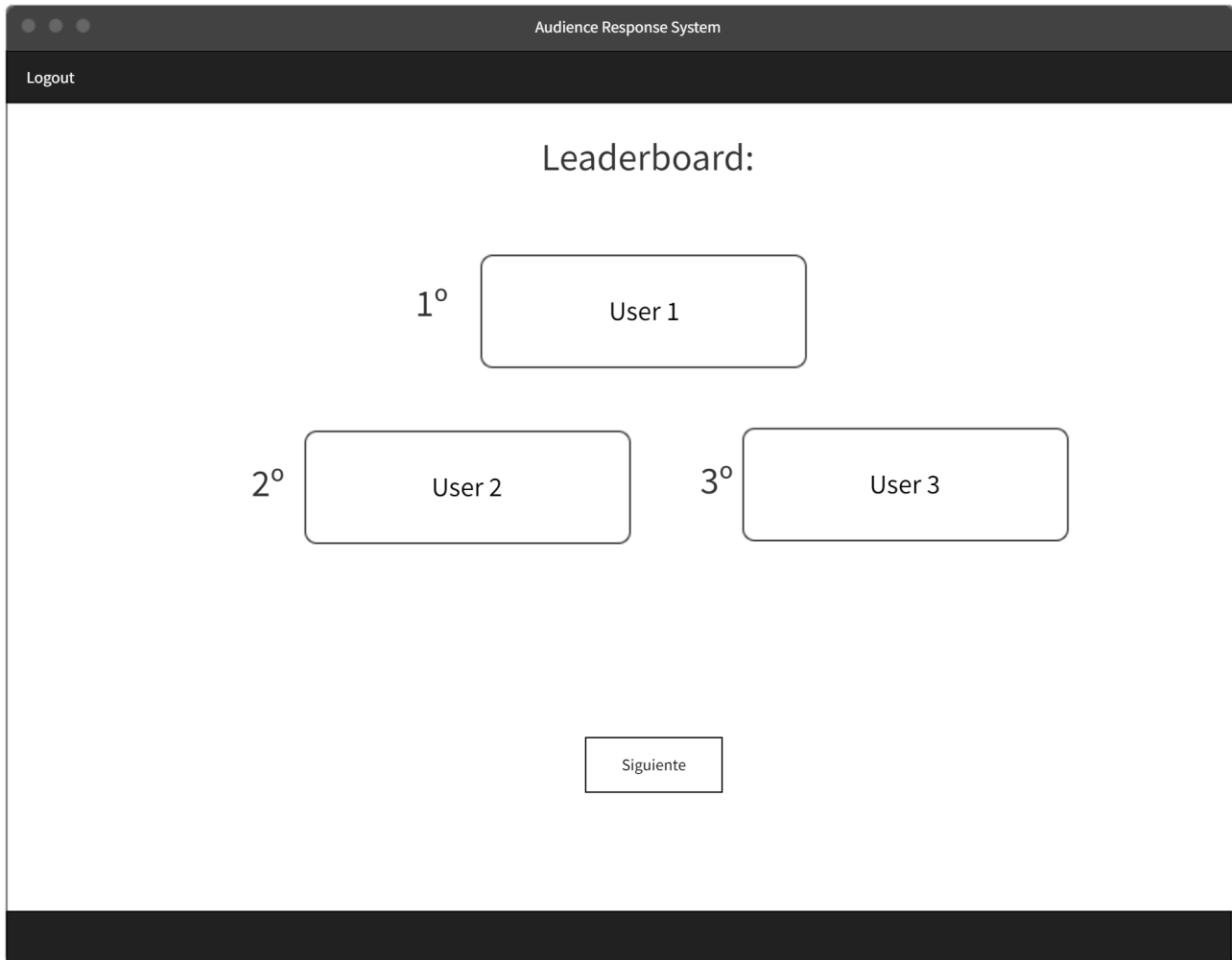


### 6.6.1.16 *Pantalla de juego (mostrando respuesta)*



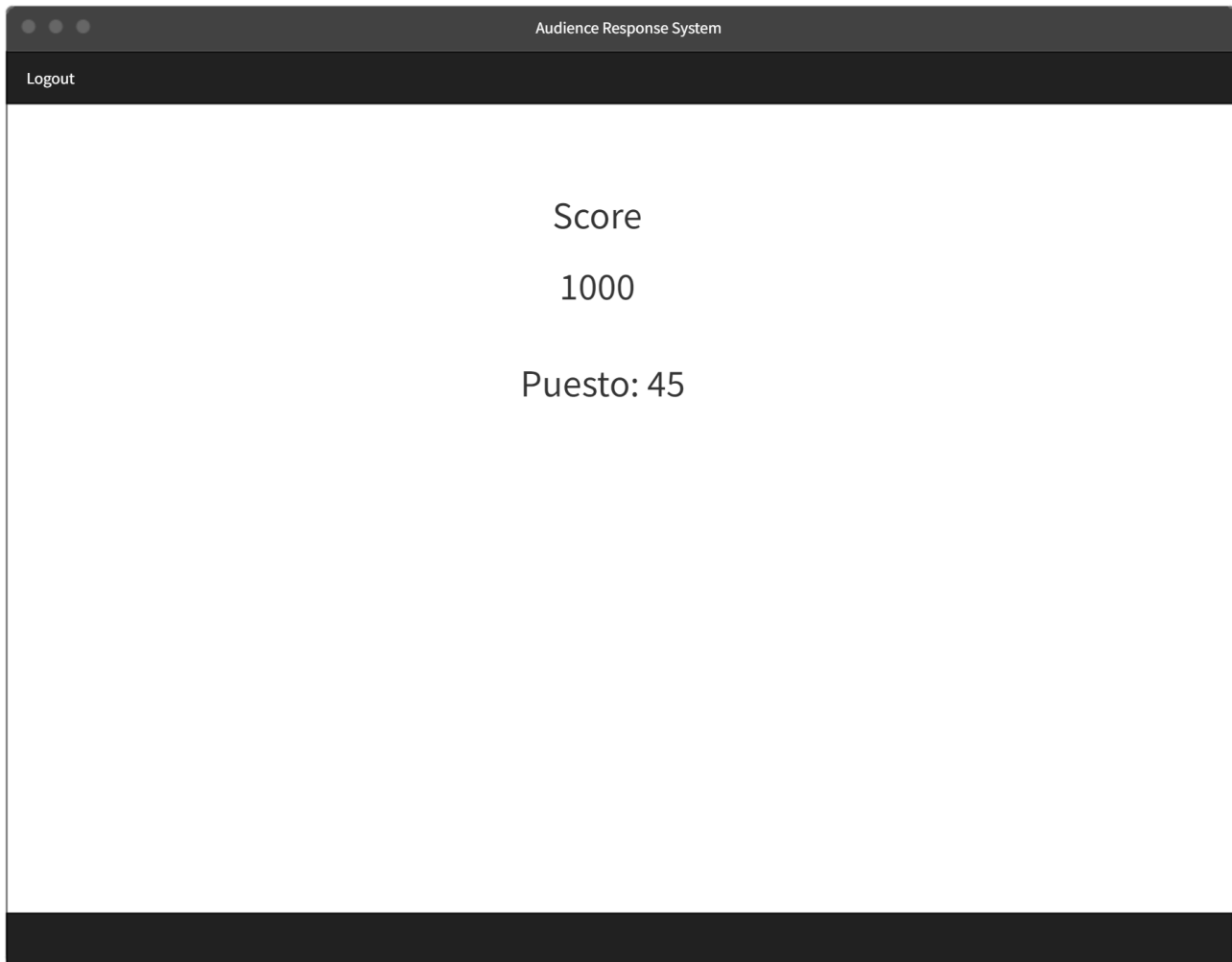
***Ilustración 38 MOCKAUP DE LA PANTALLA DE JUEGO (RESPUESTA)***

### 6.6.1.17 Pantalla de juego (mostrando podio)



*Ilustración 39 MOCKUP DE LA PANTALLA DE JUEGO (PODIO)*

### 6.6.1.18 Pantalla de juego (mostrando puntuación de un alumno)



*Ilustración 40 MOCKUP DE PANTALLA DE JUEGO (PUNTUACIÓN)*

## 6.6.2 Descripción del Comportamiento de la Interfaz

En el apartado anterior se han visto una aproximación de las pantallas que van a componer a la aplicación web. Sin embargo, hay ciertos comportamientos dinámicos, como menús laterales o desplegados, que no se pueden percibir con los mockups.

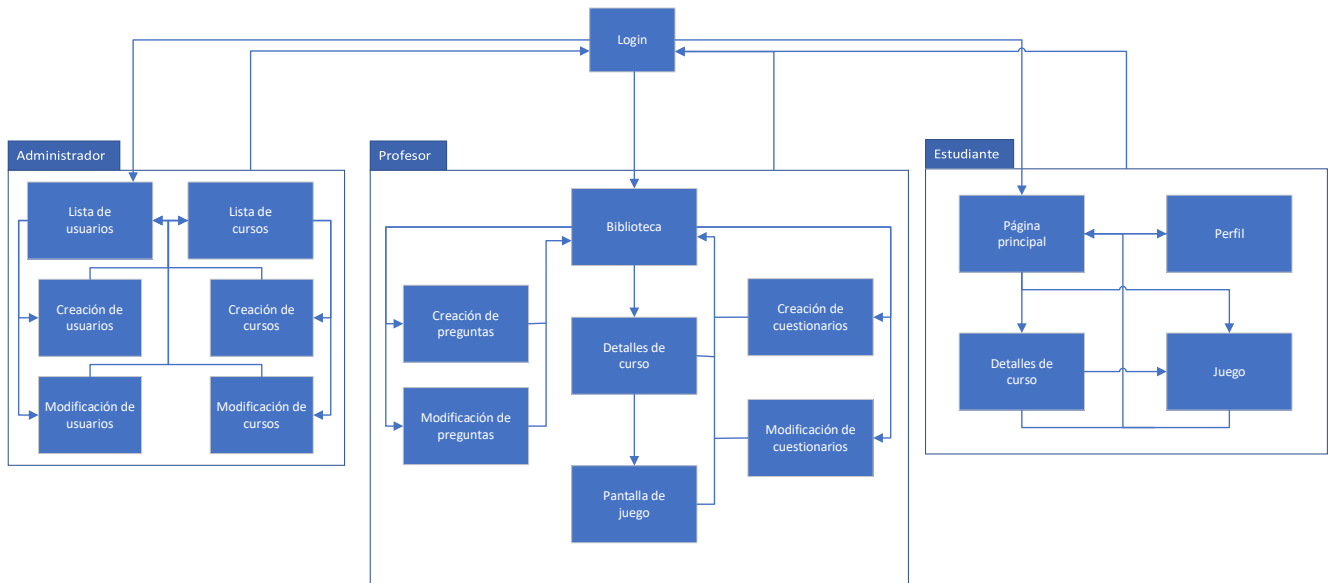
Todas las pantallas tienen tres elementos en común: una barra de navegación, un pie de página y el contenido de la página. La barra de navegación permite al usuario cambiar entre pantallas rápidamente y, la información de esta varía en función del rol del usuario identificado. El pie de página mostrará algún tipo de información, aunque en esta fase todavía no está claro cuál. El último elemento es en el que se muestra la información principal de cada una de las pantallas.

Cuando el sistema realice cualquier operación donde sea necesario llamar a cualquiera de los servicios, la aplicación mostrará una animación de espera, debido a que el tiempo de respuesta de la



petición puede llegar a demorarse. Además, para cualquier petición que requiera del usuario introducir datos, se mostrarán un mensaje de éxito o de error, dependiendo del caso, manteniendo en todo momento al usuario al tanto de lo que está ocurriendo en la aplicación. Por último, en el caso de algunas pantallas, como la de juego, si el usuario decide salirse de la pantalla o refrescar la página, se pedirá confirmación al usuario, debido a que se trata de una acción crítica.

### 6.6.3 Diagrama de Navegabilidad



*Ilustración 41 DIAGRAMA DE NAVEGABILIDAD*



## 6.7 ASI 10: ESPECIFICACIÓN DEL PLAN DE PRUEBAS

---

En esta sección, se realiza una primera aproximación del plan de pruebas, donde se indican los tipos de pruebas que se llevarán a cabo, su principal finalidad y los modelos del sistema que serán sometidos a ellas. En la sección [7.5 DSI 10: Especificación Técnica del Plan de Pruebas](#), se detallarán los aspectos técnicos de cada prueba, así como las herramientas utilizadas para llevarlas a cabo. El objetivo principal de estas pruebas es verificar el correcto funcionamiento del sistema y asegurar que cumple con los requisitos establecidos.

### 6.7.1 Pruebas unitarias

Las pruebas unitarias son un tipo de prueba en el ámbito del desarrollo de software que se centra en probar las unidades más pequeñas de código, conocidas como "unidades". Una unidad puede ser una función, un método o incluso una clase individual. El propósito principal de las pruebas unitarias es verificar que cada unidad funcione correctamente de forma aislada, es decir, independientemente del resto del sistema. Se realizarán pruebas unitarias para cada uno de los componentes de la aplicación web y para todos los controladores de la API Rest. Según la pirámide de Cohn [9], la mayoría de las pruebas de un proyecto deberían de ser unitarias, para formar una base sobre las que ejecutar los demás tipos de pruebas.

### 6.7.2 Pruebas de integración y de sistema

Las pruebas de integración se enfocan en probar la interacción entre los diferentes componentes o módulos del sistema. Estas pruebas se llevan a cabo después de las pruebas unitarias y tienen como objetivo identificar posibles problemas o fallos que puedan surgir al combinar las diferentes unidades de código. Se verifica que los componentes se integren correctamente, que los datos se transmitan de manera adecuada y que las interfaces entre los módulos funcionen según lo esperado. Para este proyecto se realizarán pruebas de integración en relación con la comunicación entre la aplicación web y el servidor web.

Por otro lado, las pruebas de sistema se centran en validar el sistema en su totalidad, una vez que todas las partes individuales se han integrado. Estas pruebas se realizan para evaluar el sistema completo en relación con los requisitos y especificaciones establecidos. El objetivo principal es verificar si el sistema cumple con los criterios de aceptación y si funciona correctamente en su entorno operativo. Esto implica probar todas las funcionalidades, las interfaces con otros sistemas externos, la capacidad de manejar una carga de trabajo esperada y comprobar si se cumplen los objetivos de rendimiento y seguridad. Concretamente, se llevarán a cabo pruebas relacionadas con la interfaz de usuario para verificar su correcto funcionamiento.



### 6.7.3 Pruebas de usabilidad

Las pruebas de usabilidad son un tipo de prueba que se lleva a cabo en el desarrollo de software para evaluar la facilidad de uso, la experiencia del usuario y la eficiencia de un producto o sistema en relación con sus usuarios reales o potenciales. El objetivo principal de estas pruebas es identificar problemas de usabilidad y obtener retroalimentación sobre cómo los usuarios interactúan con el sistema, con el fin de mejorar su diseño y funcionalidad. Se diseñarán unos formularios cuyo objetivo es evaluar la satisfacción de diferentes usuarios con la aplicación.

### 6.7.4 Pruebas de accesibilidad

Es muy importante tener en cuenta la accesibilidad a la hora de hacer una aplicación web. Esto permite que la mayoría de las personas puedan acceder y utilizar sin problema, independientemente de las capacidades físicas o motoras de las mismas. Para este proyecto, se llevarán a cabo una revisión preliminar de la accesibilidad, así como una evaluación de conformidad para, posteriormente, ver el nivel de accesibilidad (A, AA o AAA) según el WCAG, siendo el objetivo conseguir, al menos, un nivel AA.



# Capítulo 7 DISEÑO DEL SISTEMA DE INFORMACIÓN

**FASE DE DESARROLLO**

**DSI**

## 7.1 DSI 3: DISEÑO DE CASOS DE USO REALES

A contención, se procede a diseñar en detalle alguno de los casos de uso analizados en el capítulo anterior. No se detallarán todos, pues no sería necesario debido a la sencillez de algunos. Se detallarán los siguientes casos de uso:

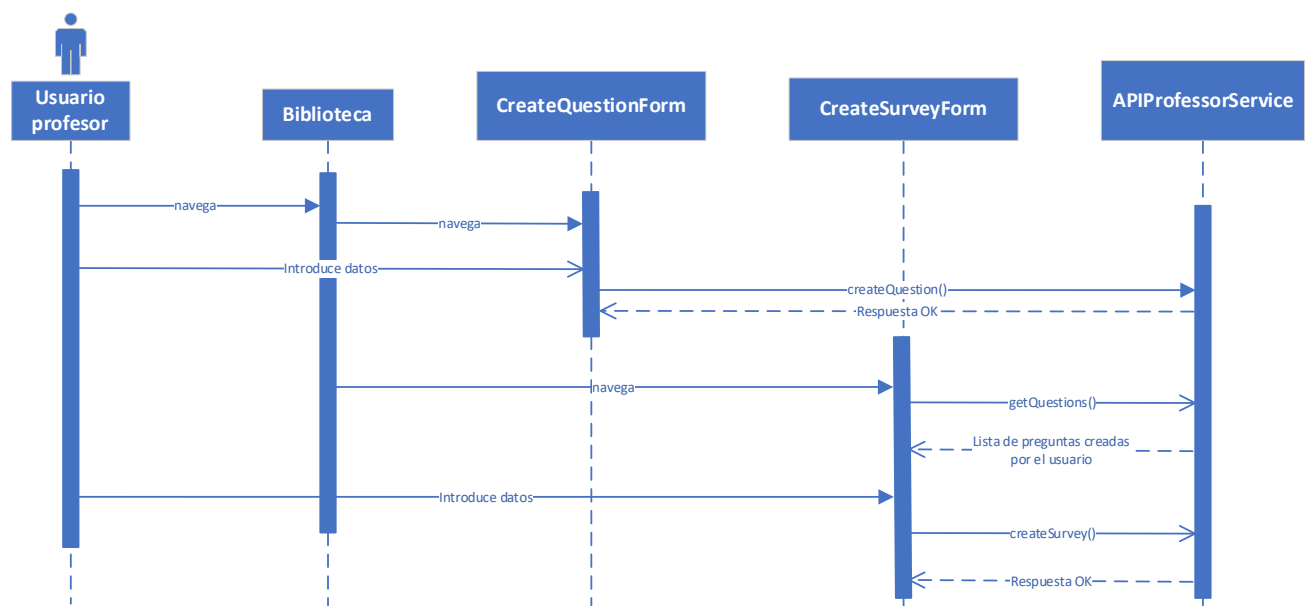
- Creación de cuestionarios.
- Lanzamiento y jugar cuestionario en línea (profesor)
- Jugar de cuestionario en línea (alumno)

Los dos últimos se detallarán en conjunto, puesto que se llevan a cabo a la vez.

### 7.1.1 Creación de cuestionarios

Es conveniente detallar este caso de uso puesto que, para poder llevarlo a cabo, es necesario que el profesor haya creado previamente alguna pregunta, lo que conlleva a que se tenga que realizar otro caso de uso antes que este.

#### 7.1.1.1 Diagramas de Interacción



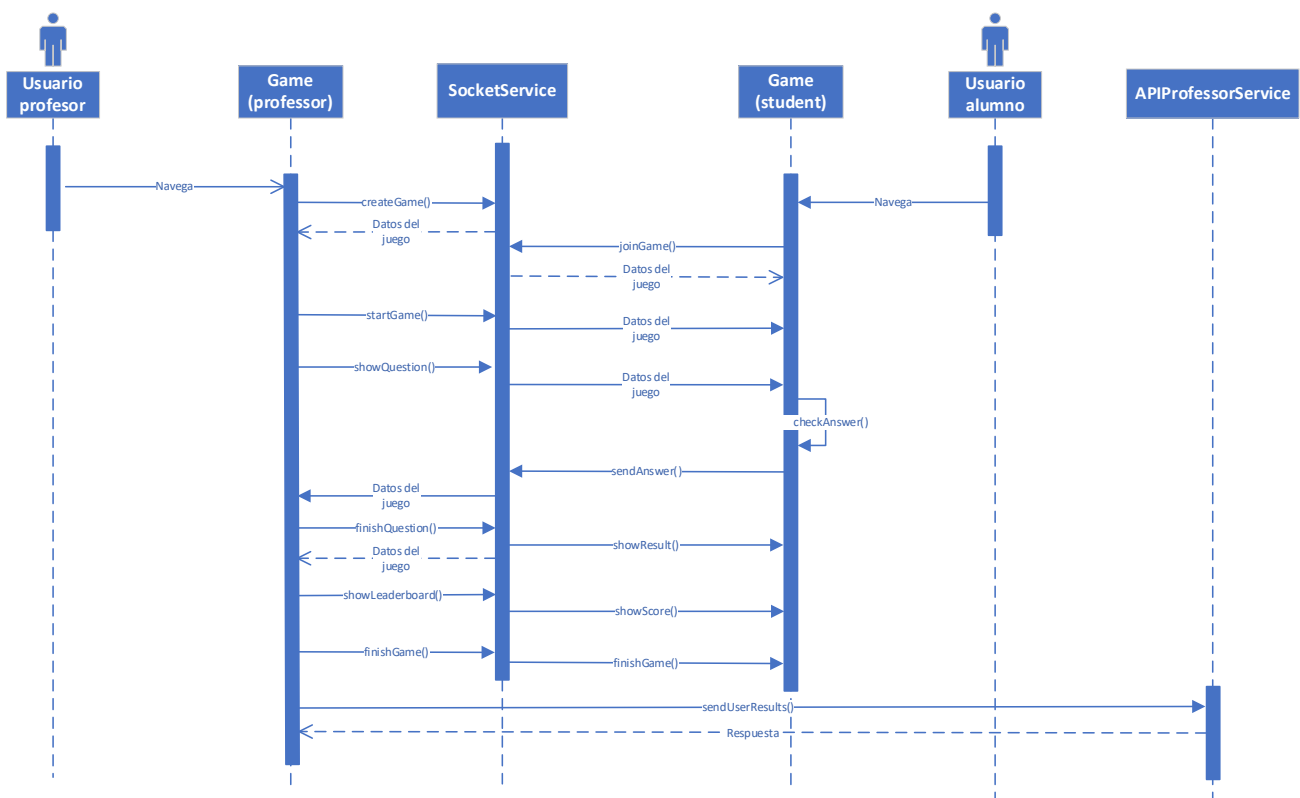
**Ilustración 42 DIAGRAMA DE INTERACCIÓN DE LA CREACIÓN DE CUESTIONARIOS**

## 7.1.2 Jugar cuestionario en línea

Es preciso estudiar este caso de uso dado que en él reside la funcionalidad principal del sistema. Vamos a tratarlo como un solo caso de uso a pesar de que se trata de dos casos separados, pues por un lado está la funcionalidad del usuario profesor y por otro la del usuario alumno, aunque ambos se complementan dado que se desarrollan conjuntamente.

### 7.1.2.1 Diagrama de Interacción

A continuación, se muestra el diagrama del desarrollo de un juego en línea, omitiendo el proceso de lanzamiento del juego, puesta que es sencillo y no es necesario añadirlo. En este diagrama, donde se desarrolla la acción del juego, solo se expresa como si el cuestionario solo tuviera una pregunta. Esto se ha hecho así porque los cuestionarios no tienen un límite de preguntas máximas, y para no alargarlo mucho se ha decidido hacerlo solo con una. En el caso de que fueran más, sería repetir el proceso establecido. Además, el diagrama solo muestra la interacción entre un profesor y un alumno, cuando en la realidad puede haber muchos alumnos. El proceso sería idéntico para cada uno de ellos.



**Ilustración 43 DIAGRAMA DE INTERACCIÓN DEL DESARROLLO DE UN CUESTIONARIO EN LÍNEA**



## 7.2 DSI 4: DISEÑO DE CLASES

---

### 7.2.1 Diagrama de Clases

A continuación, se mostrarán los diagramas de clase de la API Rest y de la aplicación web. Ambos diagramas no difieren mucho de los expuestos en la fase de análisis, sin embargo, los diagramas estarán más detallados y habrá responsabilidades y componentes nuevos, resultado de su identificación de esta fase. Los componentes relacionados con el testing se omiten para no sobrecargar los diagramas. Los cambios respecto a los diagramas anteriores son:

- En la aplicación web:
  - Se han añadido componentes utilizados para consultar los resultados de los cuestionarios que no se habían tenido en cuenta en la fase de análisis.
  - Se ha añadido otro componente al cual redirecciona la aplicación en caso de especificar una mala URL.
  - Se han añadido dos componentes más, correspondientes al encabezado y al pie de página.
- En el servidor web:
  - Los únicos cambios han sido añadir métodos que no se habían especificado.

#### 7.2.1.1 *Diagrama de clases de la aplicación web*

A continuación, se muestra el diagrama de clases final de la aplicación web dividido en partes para que sea más legible, omitiendo los componentes e interfaces propios del framework, así como los componentes utilizados únicamente para las pruebas unitarias. Hay que recordar que no es un diagrama de clases al uso, debido a que en el framework de Angular utiliza componentes en vez de las clases tradicionales.

Los diagramas están en formato svg, por lo que se pueden ampliar sin perder calidad.





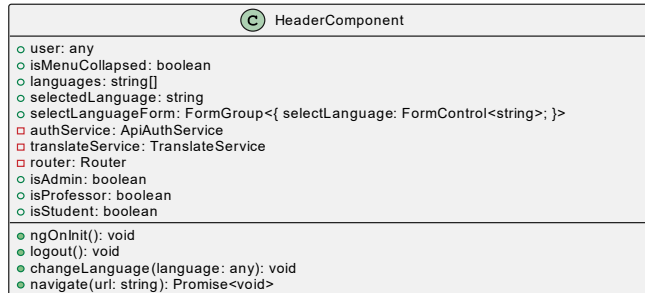
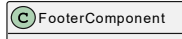
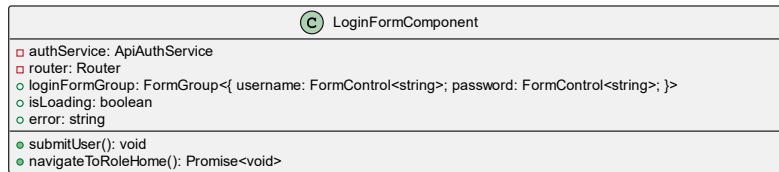


Ilustración 47 Diagrama de clases de los componentes compartidos

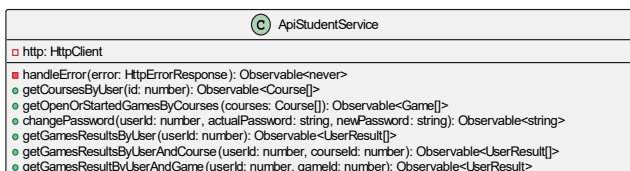
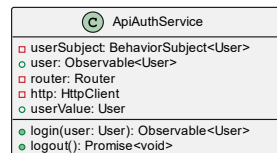
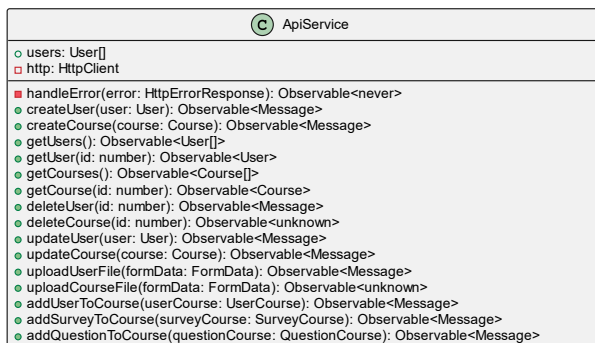


Ilustración 48 Diagrama de clases de los servicios

### 7.2.1.2 Diagrama de clases del servidor web

Cómo se ha comentado antes, los cambios respecto al diagrama de la fase de análisis no difieren en mucho. Al ser un proyecto en nodejs y el framework Expressjs, es necesario seguir una cierta estructura, por lo que ya en la fase de análisis se ha llegado a una visión bastante acertada de cómo iba a ser el diagrama en la fase de diseño. Al igual que en la aplicación web, el diagrama de clases del servidor web tampoco es un diagrama de clases como tal, si no ficheros .ts (TypeScript) y se mostrarán solo aquellos ficheros relevantes para el sistema,

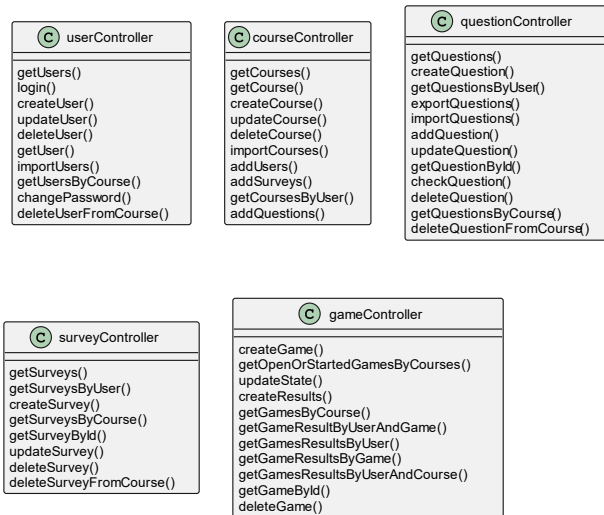


Ilustración 49 Diagrama de clases de los controladores

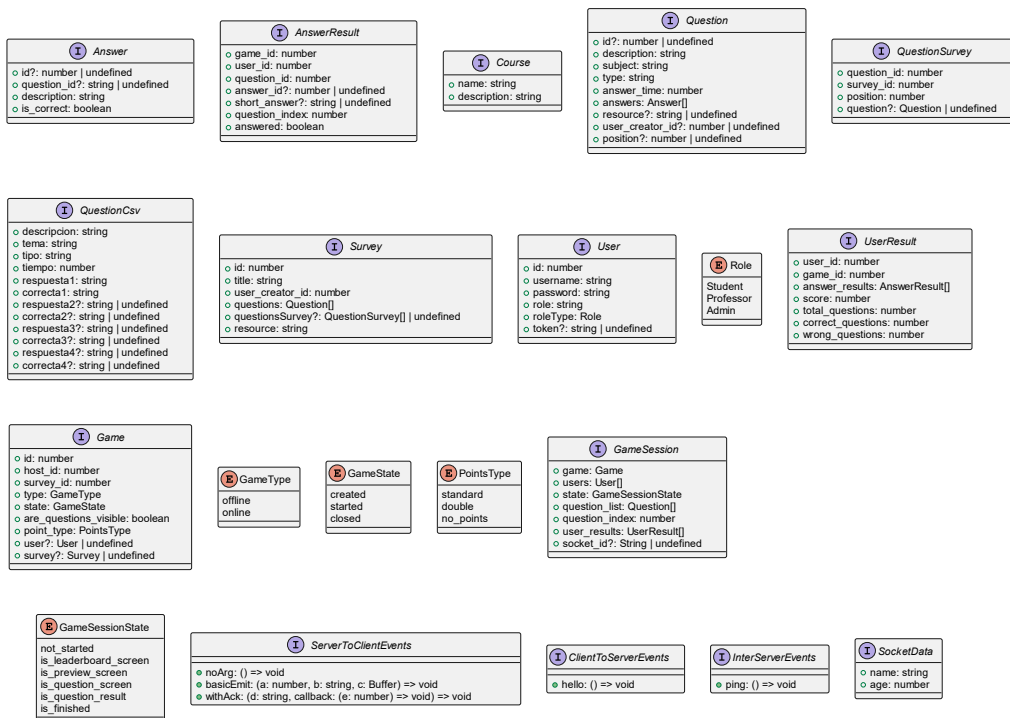


Ilustración 50 Diagrama de clases de los modelos

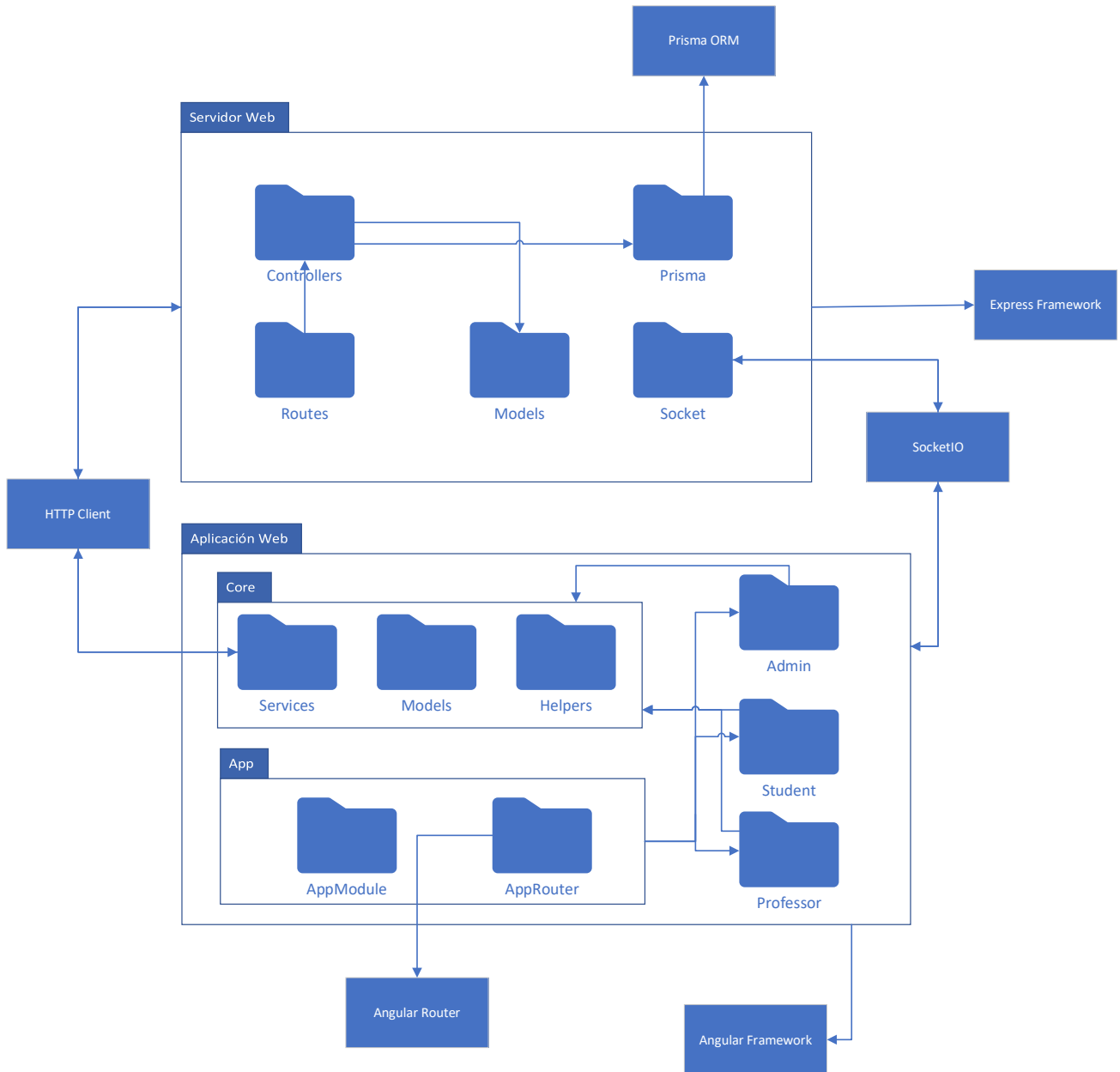


*Ilustración 51 Diagrama de clases de las rutas*

## 7.3 DSI 5: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA

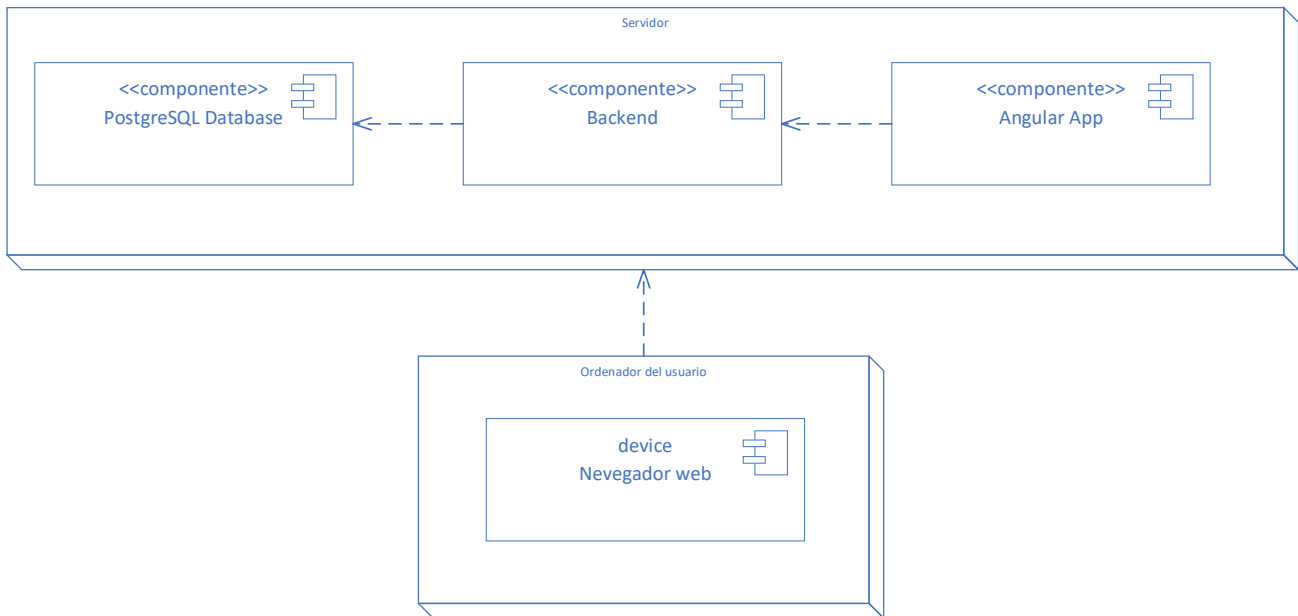
### 7.3.1 Diagrama de Paquetes

A continuación, se indica el diagrama de paquetes del sistema, estableciendo claramente los dos subsistemas existentes, así como las dependencias más importantes.



**Ilustración 52 DIAGRAMA DE PAQUETES DEL SISTEMA**

## 7.3.2 Diagrama de despliegue



*Ilustración 53 DIAGRAMA DE DESPLIEGUE DEL SISTEMA*

## 7.4 DSI 6: DISEÑO FÍSICO DE DATOS

### 7.4.1 Descripción del SGBD usado

El SGBD usado para el proyecto es PostgreSQL, visto ya en el capítulo 3 de este documento. Se trata de un candidato idóneo para aplicaciones distribuidas, además de ser uno de los SGBD de código abierto más populares empleados por la comunidad.



*Ilustración 54 LOGO DE POSTGRESQL*

## 7.4.2 Integración del SGBD en nuestro Sistema

Anteriormente se ha comentado que se va a hacer uso de un ORM para facilitar la integración y el uso de la base de datos en el sistema. Prisma [10] es un ORM para nodejs y Typescript que consiste en:

- **Prisma Client:** es un generador de consultar auto generado y seguro en cuanto a tipos, diseñado específicamente para tus datos.
- **Prisma Migrate:** herramienta de migración para evolucionar fácilmente el esquema de tu base de datos desde el prototipo hasta la producción.
- **Prisma Studio:** interfaz gráfica de usuarios que permite ver y editar los datos de la base de datos.

De estos tres componentes, se utilizan principalmente los dos primeros, el tercero es simplemente una ayuda para gestionar los datos.

Prisma trabaja con un esquema, quien es el encargado de la conexión de la base de datos como de declarar la forma que tienen las entidades dentro de ella. La conexión se establece de la siguiente manera:

```
datasource db {  
  provider = "postgresql"  
  url      = env("DATABASE_URL")  
}
```

### *Ilustración 55 CONEXIÓN A LA BASE DE DATOS MEDIANTE PRISMA*

En *url* se especifica la cadena de texto que contiene la URL de la base de datos. Dicha URL se encuentra dentro de un fichero *.env* (environment) para más seguridad. Una vez tenemos establecida la conexión, Prisma nos permite migrar al esquema las tablas que tengamos ya creadas en la base de datos, o al revés, migrar el esquema que hayamos establecido a la base de datos, es el ORM quien se encarga de realizar todas las modificaciones necesarias en la base de datos. Gracias a esto, es muy sencillo la gestión de la base de datos, pues solamente es necesario modificar el esquema, cuyos cambios se verán reflejados en la base de datos.

## 7.4.3 E-R del modelo de dominio

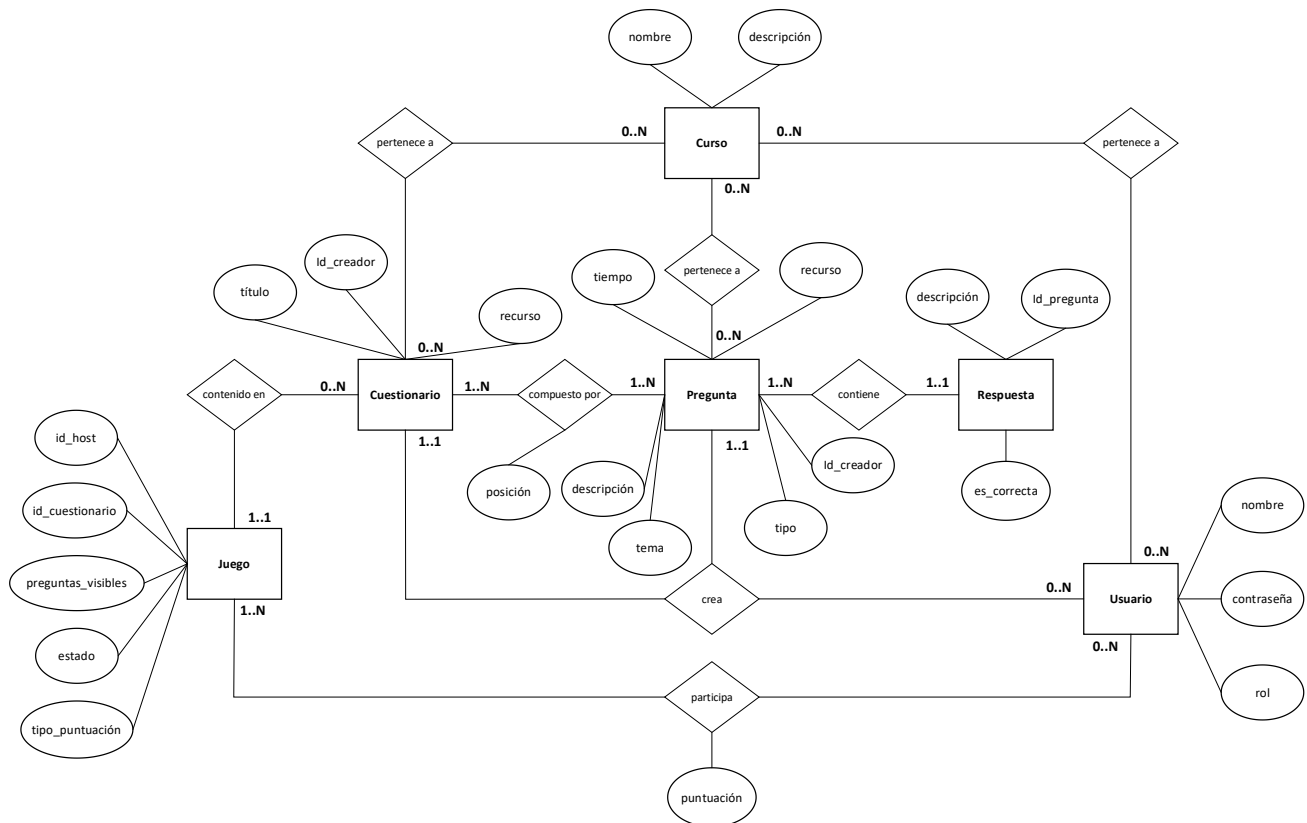


Ilustración 56 MODELO ENTIDAD RELACIÓN

## 7.5 DSI 10: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS

### 7.5.1 Pruebas Unitarias

Las pruebas unitarias se centran en verificar el funcionamiento individual de los componentes de código de un programa. Su objetivo es garantizar que cada uno de estos componentes funcione correctamente de manera aislada, independientemente de las demás partes del programa. Para este proyecto, se llevarán a cabo pruebas unitarias de los controladores de la API Rest y de cada uno de los componentes y servicios de la aplicación web para comprobar el correcto funcionamiento de cada uno de estos de forma aislada.

Cómo se ha comentado en la fase de análisis, según la **pirámide de Cohn**, el grueso de todas las pruebas del sistema han de ser unitarias, por lo que en este proyecto se ha seguido este enfoque y la mayoría de las pruebas realizadas son unitarias. Además, se ha establecido un flujo de **integración continua**, por lo que, cualquier cambio en el proyecto hace que se ejecuten estas pruebas, ayudando a preservar el correcto funcionamiento del sistema. Dicho flujo se puede encontrar en el fichero que se encuentra dentro de **.github/workflows**.





## 7.5.1.1 Controladores

### 7.5.1.1.1 Controlador de usuarios

Procedimiento	Base de datos	Entrada	Resultado esperado
Obtener todos los usuarios.	2 usuarios.		Devuelve los 2 usuarios existentes.
Obtener todos los usuarios	Sin usuarios		Devuelve un array vacío
Obtener usuario por id.	El usuario existe.	Id del usuario.	Devuelve el usuario con el id especificado.
Obtener usuario por id.	El usuario no existe.	Id del usuario.	Devuelve null.
Crear usuario no existente.	Sin usuarios.	Nombre de usuario y rol del usuario.	Creación del usuario, se le asigna el rol y su contraseña se encripta correctamente.
Crear usuario ya existente.	El usuario que se quiere crear.	Nombre de usuario de la base de datos y su rol.	Devuelve 400 y un mensaje de error diciendo que ya existe.
Crear usuario con campos vacíos.	Nada.	Algún campo vacío.	Devuelve 500 y un mensaje de error.
Crear usuario con campos con el tipo de datos incorrecto.	Nada.	Algún campo con un tipo incorrecto.	Devuelve 500 y un mensaje de error.
Iniciar sesión con credenciales válidas.	El usuario que inicia sesión.	Nombre de usuario y contraseña válidos.	Devuelve el usuario y un token.
Iniciar sesión con credenciales no coincidentes.	El usuario que inicia sesión.	Nombre de usuario y contraseña no coincidentes.	Devuelve un mensaje de usuario y contraseña incorrectos.
Iniciar sesión con un usuario inexistente.	Nada.	Nombre de usuario y contraseña.	Devuelve un mensaje de usuario y contraseña incorrectos.
Iniciar sesión con credenciales inválidas.	Nada.	Nombre o contraseña con un formato no válido.	Devuelve un mensaje de usuario y contraseña incorrectos.
Eliminar un usuario existente.	El usuario que se quiere eliminar.	Id del usuario.	Elimina correctamente el usuario y devuelve un mensaje.
Eliminar un usuario que no existe.	Nada.	Id del usuario.	Devuelve un 404 y un mensaje de error.
Eliminar un usuario sin pasarle un id.	El usuario que se quiere eliminar.	Id del usuario.	Devuelve un 400 y un mensaje de error.
Eliminar un usuario pasándole un id inválido.	Nada.	Id del usuario con un formato inválido.	Devuelve un 400 y un mensaje de error.
Importar usuarios con un fichero csv válido	Nada.	Fichero csv.	Los usuarios se crean correctamente y devuelve un mensaje.
Importar usuarios con un fichero vacío.	Nada.	Fichero vacío.	Devuelve un 500 y un mensaje de error.



Importar usuarios con un fichero con formato inválido.	Nada.	Fichero con formato inválido.	Devuelve un 500 y un mensaje de error.
Importar usuarios con un fichero csv al que le falten campos.	Nada.	Fichero csv con campos vacíos.	Devuelve un 500 y un mensaje de error.
Cambiar contraseña de un usuario existente y la contraseña antigua coincide.	Nada.	Id del usuario, la contraseña antigua y la nueva.	Actualiza correctamente la contraseña del usuario y devuelve un mensaje.
Cambiar contraseña de un usuario existente y la contraseña antigua no coincide.	Nada.	Id del usuario, la contraseña antigua y la nueva.	Devuelve un 400 y un mensaje de error.
Cambiar contraseña de un usuario con parámetros inválidos.	Nada.	Algún parámetro inválido.	Devuelve un error 500 y un mensaje de error.
Cambiar contraseña de un usuario con parámetros vacíos.	Nada.	Algún parámetro vacío.	Devuelve un error 500 y un mensaje de error.
Eliminar un usuario de un curso al que pertenece.	El usuario, el curso y su relación.	Id del usuario y del curso.	Se elimina correctamente la relación y devuelve un mensaje de error.
Eliminar un usuario de un curso al que no pertenece.	El usuario y el curso.	Id del usuario y del curso.	Devuelve un error 500 y un mensaje de error.
Eliminar un usuario de un curso sin existir el usuario.	El curso.	Id del usuario y del curso.	Devuelve un error 404 y un mensaje de error.
Eliminar un usuario de un curso que no existe	El usuario.	Id del usuario y del curso.	Devuelve un error 404 y un mensaje de error.
Eliminar un usuario de un curso pasándole un id de usuario inválido.	Nada.	Id del usuario inválido.	Devuelve un error 500 y un mensaje de error.
Eliminar un usuario de un curso pasándole un id de curso inválido.	Nada.	Id del curso inválido.	Devuelve un error 500 y un mensaje de error.

### 7.5.1.1.2 Controlador de cursos

Procedimiento	Base de datos	Entrada	Resultado esperado
Obtener todos los cursos	Dos cursos.		Devuelve todos los cursos existentes.
Obtener todos los cursos.	Nada		Devuelve un array vacío.
Obtener un curso existente por su id.	El curso que se quiere obtener.	El id del curso.	Devuelve el curso especificado.
Obtener un curso inexistente por su id.	Nada.	El id del curso.	Devuelve un error 404 y un mensaje de error.
Crear un curso con datos válido.	Nada.	El nombre y la descripción del curso.	El curso se crea correctamente y devuelve un mensaje.



Crear un curso ya existente con datos válidos.	El curso.	El nombre y la descripción del curso.	Devuelve un error 400 y un mensaje de error.
Crear un curso pasándole datos vacíos.	Nada.	Sin datos.	Devuelve un error 500 y un mensaje de error.
Crear un curso pasándole datos de diferente tipo.	Nada.	Datos con tipos erróneos.	Devuelve un error 500 y un mensaje de error.
Actualizar un curso existente.	El curso.	Id del curso, nuevo nombre y descripción.	El curso se actualiza correctamente y devuelve un mensaje.
Actualizar un curso inexistente.	Nada.	Id de curso, nuevo nombre y descripción.	Devuelve un error 404 y un mensaje de error.
Actualizar un curso sin pasarle un id.	Nada.	Nombre y descripción.	Devuelve un error 400 y un mensaje de error.
Actualizar un curso pasándole un id inválido.	Nada.	Id inválido, nombre y descripción.	Devuelve un error 400 y un mensaje de error.
Importar cursos con un fichero csv válido	Nada.	Fichero csv.	Los cursos se crean correctamente y devuelve un mensaje.
Importar cursos con un fichero vacío.	Nada.	Fichero vacío.	Devuelve un 500 y un mensaje de error.
Importar cursos con un fichero con formato inválido.	Nada.	Fichero con formato inválido.	Devuelve un 500 y un mensaje de error.
Importar cursos con un fichero csv al que le falten campos.	Nada.	Fichero csv con campos vacíos.	Devuelve un 500 y un mensaje de error.
Añadir cuestionarios a un curso correctamente.	Los cuestionarios y el curso.	Array de ids de cuestionarios e id del curso.	La relación se crea correctamente y devuelve un mensaje.
Añadir cuestionarios a un curso sin pasarle cuestionarios.	Nada.	Array de ids de cuestionarios vacío e id del curso.	Devuelve un error 400 y un mensaje de error.
Añadir cuestionarios a un curso inexistente.	Nada.	Array de ids de cuestionarios e id del curso.	Devuelve un error 404 y un mensaje de error.
Añadir cuestionarios a un curso y al menos un cuestionario no existe.	Varios cuestionarios y el curso.	Array de ids de cuestionarios, exceptuando uno de los existentes e id del curso.	Devuelve un error 404 y un mensaje de error.
Añadir usuarios a un curso correctamente.	Los usuarios y el curso.	Array de ids de usuarios e id del curso.	La relación se crea correctamente y devuelve un mensaje.
Añadir usuarios a un curso sin pasarle cuestionarios.	Nada.	Array de ids de usuarios vacío e id del curso.	Devuelve un error 400 y un mensaje de error.
Añadir usuarios a un curso inexistente.	Nada.	Array de ids de usuarios e id del curso.	Devuelve un error 404 y un mensaje de error.
Añadir usuarios a un curso y al menos un cuestionario no existe.	Varios usuarios y el curso.	Array de ids de usuarios, exceptuando uno de los existentes e id del curso.	Devuelve un error 404 y un mensaje de error.



Añadir preguntas a un curso correctamente.	Las preguntas y el curso.	Array de ids de preguntas e id del curso.	La relación se crea correctamente y devuelve un mensaje.
Añadir preguntas a un curso sin pasarle cuestionarios.	Nada.	Array de ids de preguntas vacío e id del curso.	Devuelve un error 400 y un mensaje de error.
Añadir preguntas a un curso inexistente.	Nada.	Array de ids de preguntas e id del curso.	Devuelve un error 404 y un mensaje de error.
Añadir preguntas a un curso y al menos un cuestionario no existe.	Varias preguntas y el curso.	Array de ids de preguntas, exceptuando uno de los existentes e id del curso.	Devuelve un error 404 y un mensaje de error.
Obtener correctamente todos los cursos a los que pertenece un usuario, perteneciendo al menos a uno.	El usuario, los cursos y las relaciones entre estos.	Id del usuario.	Devuelve un array todos los cursos a los que pertenece.
Obtener correctamente todos los cursos a los que pertenece un usuario, pero no pertenece a ninguno.	El usuario y los cursos.	Id del usuario.	Devuelve un array vacío.
Obtener todos los cursos a los que pertenece un usuario pasándole un id inválido del usuario.	Nada.	Id inválido del usuario.	Devuelve un error 400 y un mensaje de error.
Obtener todos los cursos a los que pertenece un usuario sin pasarle un id.	Nada.	Nada.	Devuelve un error 400 y un mensaje de error.



### 7.5.1.1.3 Controlador de preguntas

Procedimiento	Base de datos	Entrada	Resultado esperado
Obtener todas las preguntas, existiendo al menos una.	Varias preguntas.	Nada.	Devuelve todas las preguntas existentes.
Obtener todas las preguntas, sin existir ninguna.	Nada.	Nada.	Devuelve un array vacío
Obtener una pregunta dado su id.	La pregunta.	El id de la pregunta.	Devuelve la pregunta especificada.
Obtener una pregunta inexistente dado su id.	Nada.	El id de la pregunta.	Devuelve un error 404 y un mensaje de error.
Obtener las preguntas creadas por un usuario dado, existiendo al menos una.	Varias preguntas y su creador.	El id del usuario.	Devuelve un array con todas las preguntas creadas por ese usuario.
Obtener las preguntas creadas por un usuario dado, sin haber creado ninguna.	El usuario.	El id del usuario.	Devuelve un array vacío.
Crear una pregunta con datos válidos.	El usuario que la crea.	Datos de la pregunta válidos.	El sistema la crea correctamente.
Crear una pregunta con datos válidos, cuyo creador no existe.	Nada.	Datos de la pregunta válidos.	Devuelve un error 404 y un mensaje de error.
Intentar crear una pregunta sin respuestas.	Nada.	Datos de la pregunta, exceptuando respuestas.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta sin pasarle todos los datos necesarios.	Nada.	Algunos datos de la pregunta.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta con un tiempo inferior a 5 segundos.	Nada.	Todos los datos necesarios, con un tiempo menor a 5 segundos.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta multi opción si tiene más de 4 respuestas.	Nada.	Todos los datos necesarios con más de 4 respuestas.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta multi opción si tiene menos de 4 respuestas.	Nada.	Todos los datos necesarios con menos de 4 respuestas.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta multi opción con más de una respuesta correcta.	Nada.	Todos los datos necesarios, pero con más de una respuesta correcta.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta multi opción sin ninguna respuesta correcta	Nada.	Todos los datos necesarios, pero ninguna respuesta es correcta.	Devuelve un error 500 y un mensaje de error.



Intentar crear una pregunta verdadero o falso si tiene más de 2 respuestas.	Nada.	Todos los datos necesarios con más de 4 respuestas.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta verdadero o falso si tiene menos de 2 respuestas.	Nada.	Todos los datos necesarios con menos de 4 respuestas.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta verdadero o falso con más de una respuesta correcta.	Nada.	Todos los datos necesarios, pero con más de una respuesta correcta.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta verdadero o falso sin ninguna respuesta correcta	Nada.	Todos los datos necesarios, pero ninguna respuesta es correcta.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta de respuesta corta si tiene más de 4 respuestas.	Nada.	Todos los datos necesarios con más de 4 respuestas.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta de respuesta corta si todas sus respuestas no son correctas.	Nada.	Todos los datos necesarios, pero alguna respuesta incorrecta.	Devuelve un error 500 y un mensaje de error.
Intentar crear una pregunta con algún dato con tipo erróneo.	Nada.	Algún dato no tiene el tipo correcto.	Devuelve un error 500 y un mensaje de error.
Eliminar una pregunta existente.	La pregunta que se quiere eliminar.	El id de la pregunta.	El sistema elimina la pregunta y todas las respuestas asociadas a ella y devolver un mensaje.
Eliminar una pregunta inexistente.	Nada.	El id de la pregunta.	Devuelve un error 404 y un mensaje de error.
Eliminar una pregunta pasándole un id inválido.	Nada.	El id inválido de la pregunta.	Devuelve un error 400 y un mensaje de error.
Eliminar una pregunta sin pasarle un id.	Nada.	Nada.	Devuelve un error 400 y un mensaje de error.
Importar preguntas con un fichero csv válido	Nada.	Fichero csv.	Las preguntas se crean correctamente y devuelve un mensaje.
Importar preguntas con un fichero vacío.	Nada.	Fichero vacío.	Devuelve un 500 y un mensaje de error.
Importar preguntas con un fichero con formato inválido.	Nada.	Fichero con formato inválido.	Devuelve un 500 y un mensaje de error.
Importar preguntas con un fichero csv al que le falten campos.	Nada.	Fichero csv con campos vacíos.	Devuelve un 500 y un mensaje de error.



Exportar preguntas de un usuario inexistente.	Nada.	Id del usuario.	Devuelve un error 404 y un mensaje de error.
Exportar preguntas de un usuario existente que no haya creado ninguna pregunta.	Usuario.	Id del usuario.	Devuelve un csv vacío.
Exportar preguntas de un usuario existente que haya creado alguna pregunta.	El usuario y alguna pregunta creada por él.	Id del usuario.	Devuelve un csv con las preguntas.
Actualizar una pregunta que existe.	La pregunta que se quiere actualizar.	Id y datos de la pregunta.	El sistema actualiza todos los datos y devuelve un mensaje.
Actualizar una pregunta pasándole datos vacíos.	La pregunta que se quiere actualizar.	Id y datos vacíos.	El sistema deja la pregunta con los datos que tenía anteriormente y devuelve un mensaje.
Actualizar una pregunta pasándole datos que hagan que tenga un mal formato.	La pregunta que se quiere actualizar.	Id y datos de la pregunta.	El sistema devuelve un error 400 y un mensaje de error.
Actualizar una pregunta pasándole un array vacío de respuestas.	La pregunta que se quiere actualizar.	Id y datos de la pregunta, con la lista de respuestas vacías.	El sistema devuelve un error 400 y un mensaje de error.
Actualizar una pregunta inexistente.	Nada.	Id y datos de la pregunta.	El sistema devuelve un error 404 y un mensaje de error.
Obtener todas las preguntas de un curso, teniendo el curso varias.	Las preguntas, el curso y sus relaciones.	Id del curso.	El sistema devuelve una lista con las preguntas del curso.
Obtener todas las preguntas de un curso al que todavía no se le ha asignado ninguna pregunta.	Las preguntas y el curso.	Id del curso.	El sistema devuelve una lista vacía de preguntas.
Obtener todas las preguntas de un curso pasándole un id del curso invalido.	Nada.	Id inválido del curso.	Devuelve un error 400 y un mensaje de error.
Obtener todas las preguntas de un curso sin pasarle un id.	Nada.	Nada.	Devuelve un error 400 y un mensaje de error.
Eliminar una pregunta de un curso al que pertenece.	La pregunta , el curso y su relación.	Id de la pregunta y del curso.	Se elimina correctamente la relación y devuelve un mensaje de error.
Eliminar una pregunta de un curso al que no pertenece.	La pregunta y el curso.	Id de la pregunta y del curso.	Devuelve un error 500 y un mensaje de error.



Eliminar una pregunta de un curso sin existir la pregunta.	El curso.	Id de la pregunta y del curso.	Devuelve un error 404 y un mensaje de error.
Eliminar una pregunta de un curso que no existe	La pregunta.	Id de la pregunta y del curso.	Devuelve un error 404 y un mensaje de error.
Eliminar una pregunta de un curso pasándole un id de pregunta inválido.	Nada.	Id de la pregunta inválido.	Devuelve un error 500 y un mensaje de error.
Eliminar una pregunta de un curso pasándole un id de curso inválido.	Nada.	Id del curso inválido.	Devuelve un error 500 y un mensaje de error.

#### 7.5.1.1.4 Controlador de cuestionarios

Procedimiento	Base de datos	Entrada	Resultado esperado
Obtener todos los cuestionarios, habiendo al menos uno.	Varios cuestionarios.	Nada.	El sistema devuelve una lista con todos los cuestionarios.
Obtener todos los cuestionarios, no habiendo ninguno.	Nada.	Nada.	El sistema devuelve una lista vacía.
Obtener todos los cuestionarios creados por un usuario, habiendo creado alguno.	Los cuestionarios y el usuario.	Id del usuario.	El sistema devuelve una lista con los cuestionarios creados por el usuario.
Obtener todos los cuestionarios creados por un usuario que no ha creado ninguno.	El usuario.	Id del usuario.	El sistema devuelve una lista vacía.
Obtener los cuestionarios creados por un usuario, pasando un id de usuario no válido.	Nada.	Id no válido del usuario.	El sistema devuelve un error 400 y un mensaje de error.
Obtener los cuestionarios creados por un usuario, sin pasarle el id del mismo.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Obtener un cuestionario existente dado un id válido.	El cuestionario.	El id del cuestionario.	El sistema devuelve el cuestionario correspondiente al id dado.
Obtener un cuestionario inexistente dado un id válido.	Nada.	El id del cuestionario.	El sistema devuelve null.
Obtener un cuestionario dado un id inválido.	Nada.	El id inválido del cuestionario.	El sistema devuelve un error 400 y un mensaje de error.





Obtener un cuestionario sin proporcionar un id.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Obtener todos los cuestionarios de un curso, teniendo el curso varios cuestionarios.	Los cuestionarios, el curso y sus relaciones.	Id del curso.	El sistema devuelve una lista con los cuestionarios del curso.
Obtener todos los cuestionarios de un curso al que todavía no se le ha asignado ninguna pregunta.	Los cuestionarios y el curso.	Id del curso.	El sistema devuelve una lista vacía de cuestionarios.
Obtener todos los cuestionarios de un curso pasándole un id del curso invalido.	Nada.	Id inválido del curso.	Devuelve un error 400 y un mensaje de error.
Obtener todos los cuestionarios de un curso sin pasarle un id.	Nada.	Nada.	Devuelve un error 400 y un mensaje de error.
Crear un cuestionario con datos válidos y su creador existe.	Usuario que crea la pregunta.	Datos del cuestionario.	El sistema crea el cuestionario correctamente y devuelve un mensaje.
Crear un cuestionario con datos válidos y su creador no existe.	Nada.	Datos del cuestionario.	El sistema devuelve un error 404 y un mensaje de error.
Crear un cuestionario omitiendo algún dato.	Nada.	Datos incompletos del cuestionario.	El sistema devuelve un error 500 y un mensaje de error.
Crear un cuestionario con algún dato de tipo erróneo.	Nada.	Datos erróneos del cuestionario.	El sistema devuelve un error 500 y un mensaje de error.
Crear un cuestionario sin preguntas asociadas.	Nada.	Datos válidos, pero sin preguntas asociadas.	El sistema devuelve un error 400 y un mensaje de error.
Actualizar un cuestionario existente con datos válidos.	El cuestionario que se quiere actualizar.	Datos válidos para actualizar el cuestionario.	El sistema actualiza el cuestionario correctamente y devuelve un mensaje.
Actualizar un cuestionario existente sin preguntas asociadas.	El cuestionario que se quiere actualizar.	Datos válidos, pero sin preguntas.	El sistema devuelve un error 400 y un mensaje de error.
Actualizar un cuestionario inexistente.	Nada.	Datos válidos.	El sistema devuelve un error 404 y un mensaje de error.
Eliminar un cuestionario existente pasando un id válido.	El cuestionario que se quiere eliminar.	Id del cuestionario.	El sistema elimina correctamente el cuestionario y devuelve un mensaje.



Eliminar un cuestionario que no existe.	Nada.	Id del cuestionario.	El sistema devuelve un mensaje de error 404 y un mensaje de error.
Eliminar un cuestionario sin pasarle un id.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Eliminar un cuestionario pasándole un id inválido.	Nada.	Un id no válido del cuestionario.	El sistema devuelve un error 400 y un mensaje de error.
Eliminar un cuestionario de un curso al que pertenece.	El cuestionario, el curso y su relación.	Id del cuestionario y del curso.	Se elimina correctamente la relación y devuelve un mensaje de error.
Eliminar un cuestionario de un curso al que no pertenece.	El cuestionario y el curso.	Id del cuestionario y del curso.	Devuelve un error 500 y un mensaje de error.
Eliminar un cuestionario de un curso sin existir el cuestionario.	El curso.	Id del cuestionario y del curso.	Devuelve un error 404 y un mensaje de error.
Eliminar un cuestionario de un curso que no existe	El cuestionario.	Id del cuestionario y del curso.	Devuelve un error 404 y un mensaje de error.
Eliminar un cuestionario de un curso pasándole un id de cuestionario inválido.	Nada.	Id del cuestionario inválido.	Devuelve un error 500 y un mensaje de error.
Eliminar un cuestionario de un curso pasándole un id de curso inválido.	Nada.	Id del curso inválido.	Devuelve un error 500 y un mensaje de error.

#### 7.5.1.1.5 Controlador de juegos

Procedimiento	Base de datos	Entrada	Resultado esperado
Crear un juego cuyo creador y cuestionario existen.	Usuario y cuestionario asociados.	Datos para crear el juego.	El sistema crea el juego correctamente y lo devuelve.
Crear un juego cuyo creador no existe.	El cuestionario asociado.	Datos para crear el juego.	El sistema devuelve un error 404 y un mensaje de error.
Crear un juego cuyo cuestionario no existe.	El usuario asociado.	Datos para crear el juego.	El sistema devuelve un error 404 y un mensaje de error.
Crear un juego con datos necesarios omitidos.	El usuario y el cuestionario asociados.	Algunos datos para crear el juego.	El sistema devuelve un error 500 y un mensaje de error.
Crear un juego con datos con tipos erróneos.	El usuario y el cuestionario asociado.	Datos erróneos para crear el juego.	El sistema devuelve un error 500 y un mensaje de error.
Actualizar el estado de un juego existente.	El juego que se quiere actualizar.	El id y el nuevo estado del juego.	El sistema actualiza correctamente el juego y lo devuelve.



Actualizar un juego inexistente.	Nada.	Id y nuevo estado del juego.	El sistema devuelve un error 404 y un mensaje de error.
Actualizar un juego omitiendo el id o el estado.	El juego que se quiere actualizar.	Id o nuevo estado del juego.	El sistema devuelve un error 400 y un mensaje de error.
Actualizar un juego pasándole un id erróneo.	El juego que se quiere actualizar.	Id erróneo del juego.	El sistema devuelve un error 400 y un mensaje de error.
Eliminar un juego existente pasando un id válido.	El juego que se quiere eliminar.	Id del juego.	El sistema elimina correctamente el juego y devuelve un mensaje.
Eliminar un juego inexistente pasando un id válido.	Nada	Id del juego.	El sistema devuelve un error 404 y un mensaje de error.
Eliminar un juego pasando un id inválido.	Nada.	Id inválido del juego.	El sistema devuelve un error 400 y un mensaje de error.
Eliminar un juego sin pasarle un id.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Obtener los juegos abiertos o empezados dado un curso	Tres cursos, uno abierto, uno empezado y otro cerrado y dos cursos.	Lista de ids de cursos.	El sistema devuelve solo los juegos abiertos o empezados y no el cerrado.
Obtener los juegos abiertos o empezados sin pasarle nada.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Obtener los juegos abiertos o empezados pasando una lista de ids de cursos vacía.	Nada.	Una lista vacía de ids de cursos.	El sistema devuelve un error 400 y un mensaje de error.
Crear los resultados con datos correctos.	El juego asociado.	Una lista con los resultados que se quieren crear.	El sistema crea todos los resultados correctamente y devuelve un mensaje.
Crear los resultados pasándole datos con mal formato.	Nada.	Datos con un mal formato.	El sistema devuelve un error 400 y un mensaje de error.
Crear los resultados pasándole una lista de resultados vacía.	Nada.	Una lista de resultados vacía.	El sistema devuelve un error 400 y un mensaje de error.
Obtener todos los resultados de un usuario, existiendo al menos uno.	El usuario y los resultados.	El id del usuario.	El sistema devuelve todos los resultados existentes.
Obtener todos los resultados de un usuario, no existiendo ninguno.	El usuario.	El id del usuario.	El sistema devuelve una lista vacía de resultados.



Obtener todos los resultados de un usuario inexistente.	Nada.	El id del usuario.	El sistema devuelve un error 404 y un mensaje de error.
Obtener todos los resultados de un usuario sin proporcionar el id.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Obtener todos los resultados de un usuario proporcionando un id no válido.	Nada.	El id no válido del usuario.	El sistema devuelve un error 400 y un mensaje de error.
Obtener los juegos de un curso, existiendo el curso y al menos un juego.	Varios juegos y un curso.	El id del curso.	El sistema devuelve una lista con todos los juegos del curso.
Obtener los juegos de un curso, existiendo el curso, pero no ningún juego.	El curso.	El id del curso.	El sistema devuelve una lista de juegos vacía.
Obtener los juegos de un curso, no existiendo el curso.	Nada.	El id del curso.	El sistema devuelve un error 404 y un mensaje de error.
Obtener los juegos de un curso, sin proporcionar el id del curso.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Obtener los juegos de un curso, proporcionando un id inválido.	Nada.	El id inválido del curso.	El sistema devuelve un error 400 y un mensaje de error.
Obtener el resultado de un juego dado el usuario, existiendo el juego, el usuario y el resultado.	El resultado, el usuario y el juego.	El id del usuario y del juego.	El sistema devuelve el resultado para el juego y el usuario especificados.
Obtener el resultado de un juego dado el usuario, pero no existiendo el resultado.	El usuario y el juego.	El id del usuario y del juego.	El sistema devuelve un null.
Obtener el resultado de un juego dado el usuario, sin existir el juego.	El usuario.	El id del usuario y del juego.	El sistema devuelve un error 404 y un mensaje de error.
Obtener el resultado de un juego dado el usuario sin pasarle el id del juego.	Nada.	El id del usuario.	El sistema devuelve un error 400 y un mensaje de error.
Obtener el resultado de un juego dado el usuario pasándole un id del juego no válido.	Nada.	El id del usuario válido y el id del juego no válido.	El sistema devuelve un error 400 y un mensaje de error.
Obtener el resultado de un juego dado el usuario, sin existir el usuario.	El juego.	El id del usuario y del juego.	El sistema devuelve un error 404 y un mensaje de error.
Obtener el resultado de un juego dado el usuario sin pasarle el id del juego.	Nada.	El id del usuario.	El sistema devuelve un error 400 y un mensaje de error.



Obtener el resultado de un juego dado el usuario pasándole un id de usuario no válido.	Nada.	El id del usuario no válido y el id del juego no válido.	El sistema devuelve un error 400 y un mensaje de error.
Obtener los resultados de un juego, existiendo al menos uno.	El juego y sus resultados.	El id del juego.	El sistema devuelve los resultados de dicho juego.
Obtener los resultados de un juego, sin existir ningún resultado.	El juego.	El id del juego.	El sistema devuelve una lista vacía de resultados.
Obtener los resultados de un juego que no existe.	Nada.	El id del juego.	El sistema devuelve un error 404 y un mensaje de error.
Obtener los resultados de un juego pasando un id de juego no válido.	Nada.	El id inválido del juego.	El sistema devuelve un error 400 y un mensaje de error.
Obtener los resultados de un juego sin pasarle un id de juego.	Nada.	Nada.	El sistema devuelve un error 400 y un mensaje de error.
Obtener todos los resultados de los juegos de un curso dado un usuario, existiendo al menos un resultado, el usuario y el curso.	El usuario, el curso y algún resultado.	El id del usuario y del curso.	El sistema devuelve una lista de resultados.
Obtener todos los resultados de los juegos de un curso dado un usuario, sin existir ningún resultado.	El usuario y el curso.	El id del usuario y del curso.	El sistema devuelve una lista vacía de resultados.
Obtener todos los resultados de los juegos de un curso dado un usuario, sin existir el curso.	El usuario.	El id del usuario y del curso.	El sistema devuelve un error 404 y un mensaje de error.
Obtener todos los resultados de los juegos de un curso dado un usuario, pasando un id de curso inválido.	Nada	El id válido del usuario y el id inválido del curso.	El sistema devuelve un error 400 y un mensaje de error.
Obtener todos los resultados de los juegos de un curso dado un usuario, sin pasarle el id del curso.	Nada.	El id del usuario.	El sistema devuelve un error 400 y un mensaje de error.
Obtener todos los resultados de los juegos de un curso dado un usuario, sin existir el usuario.	El curso.	El id del usuario y del curso.	El sistema devuelve un error 404 y un mensaje de error.



Obtener todos los resultados de los juegos de un curso dado un usuario, pasando un id de usuario inválido.	Nada.	El id inválido del usuario y el id válido del curso.	El sistema devuelve un error 400 y un mensaje de error.
Obtener todos los resultados de los juegos de un curso dado un usuario, sin pasarle el id del usuario.	Nada.	El id del curso.	El sistema devuelve un error 400 y un mensaje de error.

### 7.5.1.2 Componentes y servicios de la aplicación web

AppComponent	
Proceso	Resultado esperado
La aplicación se inicia.	Se debería crear el componente raíz.

ErrorPageComponent	
Proceso	Resultado esperado
La aplicación se inicia.	Se debería crear el componente.

LoginFormComponent	
Proceso	Resultado esperado
La aplicación se inicia.	Se debería crear el componente.
El usuario no rellena ningún campo del formulario.	El formulario es inválido.
El usuario rellena los dos campos del formulario.	El formulario es válido.
El usuario rellena los dos campos del formulario con el usuario de tipo administrador.	El sistema permite iniciar sesión, navegando a la página de inicio del administrador.
El usuario rellena los dos campos del formulario con el usuario de tipo profesor.	El sistema permite iniciar sesión, navegando a la página de inicio del profesor.
El usuario rellena los dos campos del formulario con el usuario de tipo estudiante.	El sistema permite iniciar sesión, navegando a la página de inicio del estudiante.

CourseListComponent	
Proceso	Resultado esperado
La aplicación se inicia.	Se debería crear el componente, además de obtener los cursos del servidor y mostrarlos.
El usuario presiona en el botón de crear nuevo curso.	El sistema navega hasta la página de crear curso.
El usuario presiona en el botón de eliminar de uno de los cursos.	El sistema elimina el curso y ya no lo muestra.
El usuario sube un fichero para importar cursos.	El sistema lo detecta y lo maneja correctamente.



<b>CreateCourseFormComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente.
El usuario no rellena ningún campo del formulario.	El sistema indica que el formulario es inválido.
El usuario rellena el campo del nombre y de la descripción.	El sistema indica que el formulario es válido.
El usuario le da a crear el curso.	El sistema permite enviar la petición para crear el curso.

<b>CreateUserFormComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente.
El usuario no rellena ningún campo del formulario.	El sistema indica que el formulario es inválido.
El usuario rellena el campo del nombre de usuario y del rol.	El sistema indica que el formulario es válido.
El usuario le da a crear el usuario.	El sistema permite enviar la petición para crear el usuario.

<b>UpdateUserFormComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente y obtener el usuario al inicializarse.
El usuario no rellena ningún campo del formulario.	El sistema indica que el formulario es inválido.
El usuario rellena el campo del nombre de usuario y del rol.	El sistema indica que el formulario es válido.
El usuario le da a actualizar el usuario.	El sistema permite enviar la petición para actualizar el usuario.

<b>UpdateCourseFormComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente y obtener el usuario al inicializarse.
El usuario no rellena ningún campo del formulario.	El sistema indica que el formulario es inválido.
El usuario rellena el campo del nombre y de la descripción.	El sistema indica que el formulario es válido.
El usuario le da a actualizar el curso.	El sistema permite enviar la petición para actualizar el curso.

<b>UserListComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, además de obtener los usuarios del servidor y mostrarlos.
El usuario presiona en el botón de crear nuevo usuario.	El sistema navega hasta la página de crear usuario.
El usuario presiona en el botón de eliminar de uno de los usuarios.	El sistema elimina el usuario y ya no lo muestra.
El usuario sube un fichero para importar usuarios.	El sistema lo detecta y lo maneja correctamente.



<b>CourseProfessorDetailsComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, además de obtener y mostrar el curso y los usuarios, preguntas y juegos asociados al curso.

<b>CreateQuestionComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, así como AnswersComponent.
El usuario deja alguno de los campos del formulario en blanco.	El sistema indica que el formulario es inválido y muestra un error.
El usuario no marca ninguna respuesta como correcta.	El sistema indica que el formulario no es válido y muestra un error.
El usuario no introduce un tiempo de pregunta superior a 5 segundos.	El sistema indica que el formulario no es válido y muestra un error.
El usuario carga una imagen.	El sistema lo detecta y lo maneja correctamente.

<b>CreateSurveyComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente.
El usuario no introduce el título del cuestionario.	El sistema indica que el formulario no es válido.
El usuario no añade ninguna pregunta al cuestionario.	El sistema indica que el formulario no es válido y muestra un error.
El usuario introduce el título y al menos una pregunta.	El sistema indica que el formulario es válido y permite crear el cuestionario.
El usuario sube una imagen.	El sistema lo detecta y lo maneja correctamente.

<b>HostGameComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente.
Existen alumnos que han entrado al juego.	El sistema los muestra en la pantalla.
El juego se encuentra en la fase de previsualizar la pregunta.	El sistema debe mostrar un contador que indica el tiempo restante, así como el enunciado de la pregunta.
El juego se encuentra en la fase de visualizar la pregunta.	El sistema muestra el enunciado de la pregunta, así como sus posibles respuestas.
El juego se encuentra en la fase de visualizar la respuesta correcta.	El sistema muestra la respuesta correcta para la pregunta anterior.
El juego se encuentra en la fase de visualizar el podio.	El sistema muestra a los tres jugadores con más puntuación.
Cuando el juego termina, el profesor le da al botón de abandonar el juego.	El sistema navega a la página principal del profesor.





<b>LibraryComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear tanto este componente como SurveyListComponent, QuestionListComponent y CourseListComponent.

<b>QuestionListComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, además de obtener las preguntas del servidor y mostrarlas.
El usuario presiona en el botón de crear nueva pregunta.	El sistema navega hasta la página de crear pregunta.
El usuario presiona en el botón de eliminar de una de las preguntas	El sistema elimina el usuario y ya no lo muestra.
El usuario sube un fichero para importar preguntas.	El sistema lo detecta y lo maneja correctamente.

<b>SurveyListComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, además de obtener los cuestionarios del servidor y mostrarlos.
El usuario presiona en el botón de crear nuevo usuario.	El sistema navega hasta la página de crear cuestionario.
El usuario presiona en el botón de eliminar de uno de los cuestionarios.	El sistema elimina el cuestionario y ya no lo muestra.

<b>UpdateQuestionComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear tanto este componente como CreateQuestionComponent.

<b>UpdateSurveyComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear tanto este componente como CreateSurveyComponent.

<b>FooterComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente y mostrar el nombre del autor del trabajo.



<b>HeaderComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear este componente.
Si el usuario es el administrador.	El sistema debe mostrar solo los links del administrador.
Si el usuario es un profesor.	El sistema debe mostrar solo los links del profesor.
Si el usuario es un estudiante.	El sistema debe mostrar solo los links del estudiante.
Independientemente el tipo de usuario.	El sistema debe mostrar el botón para cerrar sesión.

<b>CourseStudentDetailsComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente y obtener y mostrar los resultados del servidor.
El usuario presiona en alguno de los resultados.	El sistema navega hasta la página de revisión de un juego.

<b>StudentGameComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear este componente.
El juego no ha empezado.	El sistema debe indicarle al usuario que busque su nombre en la pantalla del profesor.
El juego está en la fase de previsualización de la pregunta.	El sistema solo muestra un contador.
El juego está en la fase de visualización de la pregunta.	El sistema muestra el enunciado de la pregunta, así como la imagen si la tuviera y las respuestas posibles.
El juego está en la fase de mostrar el resultado. El usuario ha contestado bien.	El sistema muestra que el alumno ha respondido correctamente.
El juego está en la fase de mostrar el resultado. El usuario ha contestado mal.	El sistema muestra que el alumno ha respondido incorrectamente.
El juego está en la fase de mostrar el resultado. El usuario no ha contestado.	El sistema muestra que el alumno ha respondido incorrectamente.
El juego está en la fase de mostrar el podio.	El sistema muestra al alumno la puntuación total que lleva hasta el momento.
El juego ha finalizado. El usuario le da al botón de salir.	El sistema debe navegar hasta la página principal del alumno.

<b>StudentHomeComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, así como obtener y mostrar los cursos a los que pertenece el usuario y los juegos abiertos.
El usuario hace clic en alguno de los cursos.	El sistema navega hasta la pantalla de detalles de un curso.
El usuario hace clic en alguno de los juegos.	El sistema navega hasta la pantalla de juego del alumno.



<b>StudentProfileComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente y obtener y mostrar los últimos resultados de los juegos jugados por el alumno.
El jugador no rellena ningún campo del formulario de actualizar contraseña.	El sistema indica que el formulario es inválido.
El jugador rellena todos los campos del formulario.	El sistema indica que el formulario es válido.
El jugador le da a actualizar contraseña habiendo rellenado los campos.	El sistema permite enviar la petición.

<b>GameDetailsComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente y obtener y mostrar los resultados de todos los usuarios que hayan participado en el juego.

<b>AnswerComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente.

<b>CreateGameDialogComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente.
El usuario no rellena todos los campos del formulario de crear juego.	El sistema indica que el formulario es inválido.
El usuario rellena el campo de tipo de puntuación.	El sistema indica que el formulario es válido.
El usuario le da a crear juego.	El sistema permite llevar a cabo la petición de crear juego.

<b>LinkQuestionCourseComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, así como QuestionListComponent.

<b>LinkSurveyCourseComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, así como SurveyListComponent.

<b>LinkUserCourseComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente, así como UserListComponent.



<b>GameRevisionComponent</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el componente y obtener y mostrar el resultado del usuario, así como las preguntas del cuestionario.

<b>ApiAdminService</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el servicio.
Se llama a createUser() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a createUser() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a createCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a createCourse() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getUsers() y la api devuelve un resultado correcto.	El servicio devuelve todos los usuarios disponibles.
Se llama a getUsers() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getUser() y la api devuelve un resultado correcto.	El servicio devuelve el usuario con el id especificado.
Se llama a getUser() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getCourses() y la api devuelve un resultado correcto.	El servicio devuelve todos los cursos disponibles.
Se llama a getCourses () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getCourse() y la api devuelve un resultado correcto.	El servicio devuelve el curso con el id especificado.
Se llama a getCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteUser() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteUser () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a updateUser() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a updateUser () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a updateCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.



Se llama a updateCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a uploadUserFile () y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a uploadUserFile() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a uploadCourseFile () y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a uploadCourseFile() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a addUserToCourse () y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a addUserToCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a addSurveyToCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a addSurveyToCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a addQuestionToCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a addQuestionToCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.

<b>ApiAuthService</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el servicio.
Se llama a login() y la api devuelve un resultado correcto.	El sistema debe devolver el usuario si el inicio de sesión fue correcto.
Se llama a login() y la api devuelve un resultado incorrecto.	El sistema debe devolver un mensaje de error.
Se llama a logout().	El sistema debe navegar a la pantalla de inicio de sesión.

<b>ApiProfessorService</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el servicio.
Se llama a createQuestion() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a createQuestion () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getQuestionsByUser() y la api devuelve un resultado correcto.	El servicio devuelve todas las preguntas de un usuario.
Se llama a getQuestionsByUser () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getQuestionById() y la api devuelve un resultado correcto.	El servicio devuelve la pregunta correspondiente al id especificado.
Se llama a getQuestionsById () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.



Se llama a createSruvey() y la api devuelve un resultado correcto.	El servicio devuelve el usuario con el id especificado.
Se llama a createSruvey () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getSurveysByUser() y la api devuelve un resultado correcto.	El servicio devuelve todos los cuestionarios de un usuario.
Se llama a getSurveysByUser () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getSurveyById() y la api devuelve un resultado correcto.	El servicio devuelve el cuestionario con el id especificado.
Se llama a getSurveyById () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getUsersByCourse() y la api devuelve un resultado correcto.	El servicio devuelve todos los usuarios asociados a un curso.
Se llama a getUsersByCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getSurveysByCourse() y la api devuelve un resultado correcto.	El servicio devuelve todos los cuestionarios asociados a un curso.
Se llama a getSurveysByCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a createGame() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a createGame () y la api devuelve un resultado incorrecto.	El servicio devuelve el juego creado.
Se llama a updateGame() y la api devuelve un resultado correcto.	El servicio devuelve el juego actualizado.
Se llama a updateGame () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getGameById () y la api devuelve un resultado correcto.	El servicio devuelve el juego correspondiente al id especificado.
Se llama a getGameById () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a createUserResults () y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a createUserResults () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a importQuestions () y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a importQuestions () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a updateQuestion() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a updateQuestion () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteQuestion() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteQuestion () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.



Se llama a updateSurvey() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a updateSurvey () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteSurvey() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteSurvey() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteSurveyFromCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteSurveyFromCourse() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteQuestionFromCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteQuestionCourse() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteUserFromCourse() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteUserFromCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a deleteGame() y la api devuelve un resultado correcto.	El servicio devuelve un mensaje de éxito.
Se llama a deleteGame() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getGameResultsByGame() y la api devuelve un resultado correcto.	El servicio devuelve todos los resultados del juego especificado.
Se llama a getGameResultsByGame() y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getGamesByCourse() y la api devuelve un resultado correcto.	El servicio devuelve todos los juegos del curso especificado.
Se llama a getGamesByCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.

<b>ApiUserService</b>	
<b>Proceso</b>	<b>Resultado esperado</b>
La aplicación se inicia.	Se debería crear el servicio.
Se llama a getCoursesByUser() y la api devuelve un resultado correcto.	El servicio devuelve todos los cursos a los que pertenece el usuario especificado.
Se llama a getCoursesByUser () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getOpenOrStartedGamesByCourses() y la api devuelve un resultado correcto.	El servicio devuelve todos los juegos creados o empezados de una lista de cursos dada.
Se llama a getOpenOrStartedGamesByCourses () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a changePassword() y la api devuelve un resultado correcto.	El servicio devuelve la pregunta correspondiente al id especificado.



Se llama a changePassword () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getGamesResultsByUser() y la api devuelve un resultado correcto.	El servicio devuelve todos los resultados de los cuestionarios jugados por un usuario.
Se llama a getGamesResultsByUser () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getGamesResultsByUserAndCourse() y la api devuelve un resultado correcto.	El servicio devuelve todos los resultados de los cuestionarios jugados por un usuario en un curso determinado.
Se llama a getGamesResultsByUserAndCourse () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.
Se llama a getResultByUserAndGame() y la api devuelve un resultado correcto.	El servicio devuelve el resultado de un juego y un usuario concretos.
Se llama a getResultByUserAndGame () y la api devuelve un resultado incorrecto.	El servicio devuelve un mensaje de error.

## 7.5.2 Pruebas de Integración

Habrán dos tipos de pruebas de integración, las manuales que serán realizadas por el autor del trabajo y las automatizadas. Estas últimas se tratan de pruebas e2e que automaticen algunos de los casos de uso que se especifiquen a continuación. Estas pruebas se dividirán en cuatro partes, cada una correspondiente al tipo de usuario dentro de la aplicación, además de del caso común de iniciar sesión, y cada una tendrá una serie de casos de uso y un resultado esperado.

### 7.5.2.1 Pruebas manuales

#### 7.5.2.1.1 Caso de uso común

Inicio de sesión	
Caso de prueba	Resultado esperado
Iniciar sesión con un usuario de tipo administrador existente y la contraseña coincide.	El sistema permite entrar al usuario y lo redirige a la pantalla principal del usuario administrador.
Iniciar sesión con un usuario de tipo profesor existente y la contraseña coincide.	El sistema permite entrar al usuario y lo redirige a la pantalla principal del usuario profesor.
Iniciar sesión con un usuario de tipo alumno existente y la contraseña coincide.	El sistema permite entrar al usuario y lo redirige a la pantalla principal del usuario alumno.
Se dejan en blanco el formulario de iniciar sesión.	No permite iniciar sesión.
Se introduce el nombre de un usuario existente pero la contraseña no coincide.	El sistema muestra un error diciendo que las credenciales no son correctas.
Se introduce el nombre de un usuario no existente.	El sistema muestra un error diciendo que las credenciales no son correctas.

Tabla 7 Casos de prueba Inicio de sesión





### 7.5.2.1.2 Casos de uso del administrador

Creación de usuarios	
Caso de prueba	Resultado esperado
Se introduce un nombre que no coincida con ningún otro y cualquier rol.	El sistema crea el usuario y muestra un mensaje. Además, generará una contraseña aleatoria y le enviará un correo al nombre de usuario añadiéndole "@uniovi.es" con sus credenciales.
Se introduce un nombre que coincide con el de otro usuario existente.	El sistema no crea el usuario y muestra un mensaje de error.
Se dejan el campo del nombre y el rol en blancos.	El sistema no permite crear el usuario.
Se introduce el nombre, pero no el rol.	El sistema no permite crear el usuario.

Modificación de usuarios	
Caso de prueba	Resultado esperado
Se introduce solo el nombre del usuario.	El sistema actualiza el nombre del usuario y muestra un mensaje.
Se introduce solo una nueva contraseña.	El sistema actualiza la contraseña del usuario y muestra un mensaje.
Se introduce solo el rol.	El sistema actualiza el rol del usuario y muestra un mensaje.
No se introduce ningún campo.	El sistema no actualiza ningún dato del usuario.

Eliminación de usuarios	
Caso de prueba	Resultado esperado
El usuario existe, no tiene ninguna entidad asociada y se intenta eliminar.	El sistema elimina el usuario y muestra un mensaje.
El usuario es un profesor con alguna pregunta, cuestionario creado.	El sistema no permite eliminarlo.
El usuario es un alumno con algún resultado existente.	El sistema no permite eliminarlo.

Creación de cursos	
Caso de prueba	Resultado esperado
Se introduce un nombre que no coincida con ningún otro y cualquier descripción.	El sistema crea el curso y muestra un mensaje.
Se introduce un nombre que coincide con el de otro curso existente.	El sistema no crea el curso y muestra un mensaje de error.
Se dejan el campo del nombre y la descripción en blancos.	El sistema no permite crear el curso.
Se introduce el nombre, pero no la descripción.	El sistema permite crear el curso.



Se introduce la descripción, pero no el nombre.	El sistema no permite crear el curso.
---	---------------------------------------

<b>Modificación de cursos</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
Se introduce solo el nombre del curso.	El sistema actualiza el nombre del curso y muestra un mensaje.
Se introduce solo una nueva descripción.	El sistema actualiza la descripción del curso y muestra un mensaje.
No se introduce ningún campo.	El sistema no actualiza ningún dato del curso.

<b>Eliminación de cursos</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El curso existe, no tiene ninguna entidad asociada y se intenta eliminar.	El sistema elimina el curso y muestra un mensaje.
El curso tiene asociada alguna entidad y se intenta eliminar.	El sistema no permite eliminarlo.

### 7.5.2.1.3 Casos de uso del profesor

<b>Creación de preguntas</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
Se introduce un título, una duración de 5 segundos o más, se selecciona el tipo multi opción, se rellenan las 4 respuestas y se marca una como correcta.	El sistema crear la pregunta correctamente y muestra un mensaje.
Se introduce un título, una duración de 5 segundos o más, se selecciona el tipo verdadero o falso, se rellenan las 2 respuestas y se marca una como correcta.	El sistema crear la pregunta correctamente y muestra un mensaje.
Se introduce un título, una duración de 5 segundos o más, se selecciona el tipo respuesta corta y se rellenan al menos una respuesta.	El sistema crear la pregunta correctamente y muestra un mensaje.
Se dejan en blanco el formulario todos los campos.	El sistema muestra un mensaje de error diciendo que se deben rellanar todos los campos.
Se rellenan todos los campos, pero el tiempo es menor de 5 segundos.	El sistema muestra un error diciendo que el tiempo debe ser al menos de 5 segundos.
Se rellenan todos los campos, se introduce un tiempo superior a 5, se selecciona el tipo de pregunta multi opción o verdadero o falso y no se marca ninguna respuesta como correcta.	El sistema muestra un error diciendo que se debe seleccionar una respuesta como correcta.

<b>Modificación de preguntas</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El usuario selecciona una pregunta para editarla.	Al entrar a la pantalla, los campos deben estar completos con los datos de la pregunta.



El usuario intenta modificar el tipo de la pregunta e intenta actualizar.	El sistema no permite actualizar el tipo de la pregunta y muestra un mensaje de error al usuario indicando el motivo.
El usuario intenta modificar la respuesta correcta.	El sistema no permite actualizar la respuesta correcta la pregunta y muestra un mensaje de error al usuario indicando el motivo.
El usuario edita una pregunta y modifica cualquiera de los campos, estando todos rellenos y el tiempo superior a 5.	El sistema actualiza la pregunta y muestra un mensaje al usuario.
Se modifican los campos dejando algunos vacíos.	No permite actualizar la pregunta y muestra un mensaje de error indicando el motivo.
Se decreta el tiempo de la pregunta por debajo de 5 segundos.	No permite actualizar la pregunta y muestra un mensaje de error indicando el motivo.
Se intenta actualizar con los mismos datos.	El sistema actualiza, pero no modifica ningún dato.

<b>Eliminación de preguntas</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El usuario intenta eliminar una pregunta existente que no está asociada a ningún cuestionario.	El sistema elimina la pregunta y muestra un mensaje al usuario.
El usuario intenta eliminar una pregunta asociada a algún cuestionario.	El sistema no permite eliminarla y muestra un mensaje de error explicando el motivo.

<b>Creación de cuestionarios</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
Se introduce un título y se añade al menos una pregunta.	El sistema crea el cuestionario y muestra un mensaje.
Se introduce un título y no se añade ninguna pregunta.	El sistema no permite crear el cuestionario y muestra un mensaje de error indicando el motivo.
No se introduce el título, pero si se añade al menos una pregunta.	El sistema no permite crear el cuestionario y muestra un mensaje de error indicando el motivo.
Se añaden al menos 2 preguntas y se elimina una.	El cuestionario debería quedar una pregunta menos.
Se introducen varias preguntas y el usuario intenta cambiarlas de orden.	El sistema debería dejar arrastrar las preguntas para cambiarlas de orden.
El usuario hace clic en alguna pregunta para ver sus respuestas.	El sistema muestra las respuestas de la pregunta seleccionada.

<b>Modificación de cuestionarios</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El usuario selecciona un cuestionario para editarlo.	Al entrar a la pantalla, los campos deben estar completos con los datos del cuestionario.
El usuario intenta añadir o eliminar preguntas.	El sistema no debería dejar hacerlo.



El usuario edita el título y orden de las preguntas del cuestionario.	El sistema actualiza el cuestionario y muestra un mensaje.
---	--

<b>Eliminación de cuestionarios</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El usuario intenta eliminar un cuestionario que no tenga resultados asociados.	El sistema permite eliminar el cuestionario y muestra un mensaje al usuario..
El usuario intenta eliminar un cuestionario que tiene resultados asociados.	El sistema no permite eliminar el cuestionario y muestra un mensaje de error indicando el motivo.

<b>Gestión de cursos</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El usuario intenta añadir al curso cuestionarios que no han sido asociados todavía.	El sistema añade los cuestionarios y muestra un mensaje.
El usuario intenta añadir al curso cuestionarios que ya han sido asociados.	El sistema no debe dejar añadir los cuestionarios que ya hayan sido asociados.
El usuario intenta añadir al curso preguntas que no han sido asociadas todavía.	El sistema añade las preguntas al curso y muestra un mensaje.
El usuario intenta añadir al curso preguntas que ya han sido asociadas.	El sistema no debe dejar añadir las preguntas que ya hayan sido asociadas.
El usuario intenta añadir al curso alumnos que no han sido asociados todavía.	El sistema añade los alumnos y muestra un mensaje.
El usuario intenta añadir al curso alumnos que ya han sido asociados.	El sistema no debe dejar añadir los alumnos que ya hayan sido asociados al curso.
El usuario intenta añadir al curso usuarios de tipo profesor o al administrador.	El sistema no debe mostrar los usuarios que no son alumnos.

<b>Lanzamiento de cuestionarios en línea</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El usuario selecciona un tipo de puntuación y marca que quiere que se vean las preguntas en los dispositivos del alumno.	El sistema crea el juego y pasa a la pantalla de juego.
El usuario selecciona un tipo de puntuación y marca que no quiere que se vean las preguntas en los dispositivos del alumno.	El sistema crea el juego y pasa a la pantalla de juego.
El usuario no selecciona un tipo de puntuación e intenta crear el juego.	El sistema no permite crear el juego.



<b>Jugar un cuestionario en línea</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El profesor está a la espera de que entren los alumnos.	El sistema debe mostrar los alumnos que vayan entrando y no debe dejar empezar hasta que haya al menos un alumno en la sala.
El profesor le da a empezar la partida.	Automáticamente, la pantalla del profesor empieza un contador y se muestra la previsualización de la descripción de la pregunta.
El contador de la previsualización se acaba y la pregunta es de tipo multi opción o verdadero o falso.	El sistema muestra la pregunta y todas sus posibles respuestas, así como el tiempo restante y el número de respuestas que van llegando.
El contador de la previsualización se acaba y la pregunta es de tipo respuesta corta.	El sistema muestra la pregunta e indica que el alumno debe escribir su respuesta, así como el tiempo restante y el número de respuestas que van llegando.
El tiempo de la pregunta finaliza.	El sistema muestra la respuesta correcta a la pregunta y empieza otro contador.
El contador de la pantalla del resultado finaliza.	El sistema muestra el podio con los tres jugadores con mejor puntuación.
El jugador le da al botón de siguiente en la pantalla del podio y todavía quedan preguntas.	El sistema muestra la previsualización de la siguiente pregunta.
El jugador le da al botón de siguiente en la pantalla del podio y ya no quedan preguntas.	El sistema indica que el juego se ha terminado.

<b>Consulta de resultados de los cuestionarios</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El usuario selecciona la opción de “Juegos” en la pantalla del curso.	El sistema muestra todos los cuestionarios jugados en ese curso.
El usuario clicca en uno de los juegos.	El sistema debe mostrar el resultado del cuestionario para cada uno de los alumnos que hayan participado en él.

#### 7.5.2.1.4 Casos de uso del alumno

<b>Jugar un cuestionario en línea</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El alumno entra a la aplicación y ya existía un cuestionario abierto para un curso en el que esté asociado.	El sistema debe mostrar aquellos cuestionarios que hayan sido abiertos.
El alumno se encuentra ya en la aplicación y un profesor abre un cuestionario de un curso al que pertenece.	El sistema muestra en tiempo real el cuestionario que se acaba de abrir.
El alumno no pertenece a ningún curso y se abren cuestionarios.	El sistema no debería mostrarle ningún cuestionario abierto.



El alumno selecciona un juego para jugar.	El sistema lo redirige a la pantalla del juego y espera a que el profesor inicie la partida. Además, al entrar al juego, el alumno debería ver su nombre en la pantalla del profesor.
El profesor empieza el cuestionario.	El sistema muestra un contador que ira decreyendo.
El contador finaliza y la pregunta es de tipo multi opción o verdadero o falso y no tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta, así como un botón por cada una de las posibles respuestas.
El contador finaliza y la pregunta es de tipo multi opción o verdadero o falso y tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta, la imagen asociada, así como un botón por cada una de las posibles respuestas.
El contador finaliza y la pregunta es de tipo respuesta corta y no tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta y una entrada de texto y un botón de aceptar para introducir la respuesta.
El contador finaliza y la pregunta es de tipo respuesta corta y tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta, la imagen asociada y una entrada de texto y un botón de aceptar para introducir la respuesta.
El usuario selecciona la respuesta de una pregunta de tipo multi opción o verdadero o falso.	El sistema queda a la espera de que acabe el tiempo de la pregunta, y cuando finalice muestra si el alumno acertó o no.
El usuario introduce una respuesta de una pregunta de tipo respuesta corta.	El sistema queda a la espera de que acabe el tiempo de la pregunta, y cuando finalice muestra si el alumno acertó o no.
El tiempo para mostrar el resultado de la pregunta finaliza.	El sistema muestra la puntuación que lleva hasta el momento el alumno.
El profesor le da a siguiente y quedan preguntas.	El sistema muestra otra vez un contador antes de mostrar la pregunta.
El profesor le da a siguiente y no quedan preguntas.	El sistema muestra que el juego ha finalizado y muestra un botón para salir del juego.
El profesor abandona el juego en mitad de una partida.	El sistema finaliza el juego automáticamente.
El alumno abandona el juego en mitad de una partida.	El sistema permitirá al alumno volver a entrar.
El alumno vuelve a entrar a la partida en mitad de una pregunta.	El sistema no permitirá contestar al alumno hasta la siguiente pregunta.

<b>Consultar resultado de los cuestionarios</b>	
<b>Caso de prueba</b>	<b>Resultado esperado</b>
El jugador entra en algún curso al que pertenezca y haya participado en algún juego.	El sistema mostrará todos los cuestionarios jugados, así como sus resultados.
El jugador entre en algún curso al que pertenezca y no haya participado aún en algún juego.	El sistema no muestra ningún resultado.
El jugador hace clic en algún cuestionario que haya jugado.	El sistema lo redirige a una página donde podrá revisar el cuestionario, mostrando las preguntas con las respuestas y las respuestas que haya dado el alumno.



Modificar contraseña	
Caso de prueba	Resultado esperado
El alumno introduce su antigua contraseña, la nueva y su confirmación y la contraseña antigua coincide.	El sistema actualiza la contraseña del usuario y muestra un mensaje.
El alumno introduce la antigua contraseña y la nueva y su confirmación, pero la contraseña antigua no coincide.	El sistema no permite actualizar la contraseña y muestra un mensaje de error explicando el motivo.
El alumno no introduce algún campo.	El sistema no permite actualizar la contraseña.
Las dos contraseñas nuevas no coinciden.	El sistema no permite actualizar la contraseña.

### 7.5.2.2 Pruebas e2e automatizadas

Las pruebas automatizadas llevan a cabo los mismos casos de uso que las manuales, con la excepción de que solo se probarán los caminos felices, es decir, los caminos que no llevan a error y no se realizarán los casos de uso correspondientes a jugar juegos en línea. Al igual que en las pruebas de carga, realizar pruebas e2e para estos casos es demasiado complejo por requerir dos instancias de diferentes navegadores, por lo que se decidió que dichos casos solo serán probados manualmente. Por consiguiente, tampoco se realizarán las pruebas correspondientes a la consulta de resultados.

Para realizar se utilizó una herramienta, llamada TestCafe. Se trata de una herramienta de prueba funcional de aplicaciones web desarrollada por DevExpress. Se utiliza para automatizar las pruebas en navegadores web y verificar el funcionamiento de una aplicación web en diferentes escenarios.

### 7.5.3 Pruebas de Usabilidad

Para llevar a cabo las pruebas de usabilidad, se seleccionarán varios usuarios de diferentes niveles informáticos y edades para que realicen unas determinadas tareas en la aplicación. Posteriormente, se les pedirá a estos usuarios que realicen una encuesta en relación a estas tareas, lo que permitirá evaluar lo siguiente:

- **Facilidad de aprendizaje:** ¿Qué tan fácil es para los usuarios aprender a utilizar la aplicación web por primera vez?
- **Eficiencia:** ¿Qué tan rápidamente y con qué facilidad los usuarios pueden realizar tareas específicas en la aplicación?
- **Satisfacción del usuario:** ¿Los usuarios se sienten satisfechos al utilizar la aplicación web y logran sus objetivos de manera efectiva?
- **Navegación y estructura:** ¿La navegación dentro de la aplicación web es clara, intuitiva y coherente? ¿La estructura de la información está bien organizada?
- **Diseño visual:** ¿El diseño visual de la aplicación web es atractivo, legible y coherente con la identidad de la marca?



- **Retroalimentación y respuesta del sistema:** ¿La aplicación web proporciona retroalimentación clara y oportuna sobre las acciones del usuario? ¿Las respuestas del sistema son rápidas y apropiadas?
- **Manejo de errores:** ¿La aplicación web proporciona mensajes de error claros y útiles cuando los usuarios cometen errores? ¿Se les brinda una forma de corregir esos errores?
- **Sugerencias de mejora:** Los usuarios podrán dar su punto de vista de mejoras futuras.

### 7.5.3.1 Diseño del cuestionario de evaluación

El cuestionario de evaluación constará de tres partes. En la primera, se realizarán preguntas de carácter general que no están relacionadas con el propio sistema, para determinar el nivel de destreza informática de los usuarios. En la segunda, los usuarios realizarán una serie de actividades dentro del sistema para, finalmente, realizar una serie de preguntas cortas sobre éstas.

### 7.5.3.2 Actividades de las Pruebas de Usabilidad

#### 7.5.3.2.1 Preguntas de carácter general

Edad
¿Cuál consideras que es tu nivel de experiencia en el uso de aplicaciones web?
<ol style="list-style-type: none"><li>1. Principiante</li><li>2. Intermedio</li><li>3. Experto</li></ol>
¿Con qué frecuencia utilizas aplicaciones web en tu día a día?
<ol style="list-style-type: none"><li>1. Diariamente</li><li>2. Varias veces por semana</li><li>3. Ocasionalmente</li></ol>
¿Qué tipo de dispositivo utilizas para acceder a aplicaciones web?
<ol style="list-style-type: none"><li>1. Ordenador de escritorio</li><li>2. Ordenador portátil</li><li>3. Teléfono inteligente</li><li>4. Tableta</li></ol>
¿Has usado alguna vez una aplicación para realizar cuestionarios en línea (Ej. Kahoot)?
<ol style="list-style-type: none"><li>1. Sí, he utilizado Kahoot</li><li>2. Sí, pero no Kahoot, si no otra aplicación similar.</li><li>3. No, pero me lo he planteado</li><li>4. No</li><li>5. No, y además no sé a qué se refiere con cuestionarios en línea.</li></ol>
¿De los siguientes aspectos, cuál valoras más en una aplicación web?
<ol style="list-style-type: none"><li>1. Que sea muy rápida.</li></ol>





2. Que sea intuitiva y fácil de usar.
3. Que tenga un diseño bonito.
4. Que tenga muchas funcionalidades

### 7.5.3.2.2 Tareas guiadas

Las tareas que tendrán que llevar a cabo los usuarios están divididas en función del tipo de usuario dentro de la aplicación.

#### Administrador

- Crear, actualizar, importar y eliminar usuarios.

#### Profesor

- Crear preguntas.
- Crear un cuestionario.
- Añadir las preguntas y cuestionarios a un curso.
- Lanzar un juego
- Jugar un juego (junto con otro usuario de tipo alumno que será el responsable de las pruebas)

#### Alumno

- Jugar un juego (junto con otro usuario de tipo profesor que será el responsable de las pruebas)
- Consultar resultados de un curso.
- Revisar un cuestionario.
- Modificar contraseña.

### 7.5.3.2.3 Preguntas cortas sobre la aplicación y observaciones

En una escala del 1 al 5, ¿qué tan intuitiva te pareció la interfaz de la aplicación para realizar las actividades?
<ol style="list-style-type: none"> <li>1. Nada intuitiva</li> <li>2. Poco intuitiva</li> <li>3. Neutral</li> <li>4. Bastante intuitiva</li> <li>5. Muy intuitiva</li> </ol>
¿Encuentras que las instrucciones proporcionadas eran claras y comprensibles para completar las actividades?
<ol style="list-style-type: none"> <li>1. Si</li> <li>2. No</li> <li>3. No estoy seguro</li> </ol>
¿Cuál fue la actividad que encontraste más fácil de realizar en la aplicación? ¿Por qué?



¿Cuál fue la actividad que encontraste más difícil de realizar en la aplicación? ¿Por qué?
En tu opinión, ¿qué aspectos de la aplicación podrían mejorarse para facilitar la realización de las actividades?
¿Hubo algún mensaje de error o problema técnico que encontraste durante la realización de las actividades?
1. Si 2. No
Si respondiste "Sí" en la pregunta anterior, ¿puedes describir brevemente el problema técnico que encontraste?
En términos generales, ¿te sentiste satisfecho/a con la experiencia de usar la aplicación para realizar las actividades?
1. Si, muy satisfecho/a 2. Satisfecho/a 3. Neutral 4. Insatisfecho/a 5. Muy insatisfecho/a
¿Tienes algún comentario adicional o sugerencia para mejorar la aplicación y hacerla más amigable para el usuario?

## 7.5.4 Pruebas de Accesibilidad

La aplicación será desarrollada siguiendo las pautas establecidas por el WCAG y se llevarán a cabo una revisión preliminar, tanto manual como automática, y una evaluación de conformidad. La primera servirá para detectar rápidamente problemas de accesibilidad de la aplicación, mientras que la evaluación indicará si la aplicación cumple con los estándares establecidos por el WCAG.

La revisión preliminar se enfocará en los siguientes aspectos:

- **Probar la aplicación en diferentes navegadores:** es necesario que la aplicación se ejecute bien independientemente del navegador que utilice el usuario. Es por esto que la aplicación se probará en los principales navegadores (Edge, Chrome, Firefox y Opera).
- **Evaluar el contraste de color:** Verifica que el texto y los elementos visuales en la aplicación web tengan suficiente contraste para que sean legibles. Debe haber una diferencia clara entre el color del texto y el fondo en todas las áreas de contenido.



- **Evaluar la compatibilidad con dispositivos móviles:** La aplicación fue construida desde un principio con los móviles en mente, pues hoy en día muchas personas lo utilizan para acceder a las aplicaciones web, por tanto, se evaluará si la aplicación se puede redimensionar y que no haya problemas de visualización en dispositivos con ese tamaño.
- **Comprobar la navegación y la estructura:** Se comprobará que la aplicación tenga una estructura clara y coherente, además de verificar que sea posible navegar por la aplicación utilizando solo el teclado.

## 7.5.5 Pruebas de Rendimiento

En un principio se había decidido realizar pruebas de rendimiento, concretamente pruebas de carga para ver cómo se comporta el sistema cuando se jugaba un cuestionario en línea con muchos jugadores simultáneos. Debido a la naturaleza de los cuestionarios en línea, que requieren de dos partes, la del profesor y la del alumno, para llevarse a cabo, así como al uso de *websockets*, la complejidad del uso de herramientas convencionales para crear pruebas automáticas, como Jmeter o Gatling, aumentaba considerablemente, por lo que, finalmente, se ha decidido no realizarlas. Debido a esto, solo se probarán el rendimiento de manera manual, es decir, con los dispositivos que se tengan disponibles.



# Capítulo 8 CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

**FASE DE DESARROLLO**

**CSI**



## 8.1 CSI 1: PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN

---

### 8.1.1 Estándares y normas seguidos

#### 8.1.1.1 IEEE 830-1998

Se trata de un estándar que especifica las prácticas recomendadas a la hora de escribir requisitos funcionales y no funcionales de un sistema informático. Para este proyecto, no se ha utilizado de manera estricta, si no que se ha utilizado como referencia para especificar los requisitos del sistema.

#### 8.1.1.2 JSON

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos que se utiliza comúnmente en aplicaciones web y servicios API. Fue diseñado como una alternativa más sencilla y legible para el intercambio de datos que el formato XML. En este proyecto es utilizado para el intercambio de datos entre el servidor web y la aplicación web.

#### 8.1.1.3 WebSocket

Cómo se ha comentado en los [aspectos teóricos](#), la aplicación usará este protocolo de comunicación para establecer una vía bidireccional entre el cliente el servidor y poder ejecutar cuestionarios en línea, sincronizando eventos entre los profesores y los alumnos.

### 8.1.2 Lenguajes de programación

La implementación de todo el proyecto se ha hecho con el mismo lenguaje, TypeScript. A continuación, se detallará este y otros lenguajes que han sido utilizados también, pero en menor medida.

#### 8.1.2.1 TypeScript

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. TypeScript extiende la sintaxis de JavaScript, por tanto, cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original [11].

Debido al framework Angular y de Nodejs , el uso de JavaScript era obligatorio, sin embargo, se ha decido usar TypeScript por las ventajas que ofrece sobre JavaScript. Estas ventajas son:



- Proporciona un nivel mayor de seguridad gracias al uso de tipado estático.
- Al tener tipos definidos explícitamente, facilita el mantenimiento del código debido a que es más legible y comprensible.
- Al proporcionar tipado estático también permite a los editores de código ofrece un autocompletado más preciso, resaltado de errores en tiempo real y herramientas de refactorización más potentes.

Anteriormente, se ha comentado que TypeScript es el principal lenguaje usado en este proyecto, concretamente en los siguientes frameworks y herramientas.

#### 8.1.2.1.1 Angular

Angular es una plataforma de desarrollo de código abierto, construida sobre TypeScript. Su principal objetivo es construir aplicaciones de una sola página (SPA) de manera eficiente y escalable. Se ha usado en este proyecto para construir la aplicación web del sistema, es decir, la parte Frontend. El framework utiliza un enfoque basado en componentes reutilizables que encapsulan una parte específica de la interfaz de usuario y su lógica asociada. Concretamente, cada componente aúna código TypeScript para la lógica, y HTML y CSS para definir la estructura de la página y los estilos de la misma. Además, se usa también el framework de CSS, Bootstrap, que ofrece estilos predefinidos que ayuda a aligerar el desarrollo de la aplicación.

#### 8.1.2.1.2 Node.js

Node.js es un entorno de ejecución de JavaScript que permite la ejecución de este en la parte del servidor, corriendo sobre el motor V8 de JavaScript. Presenta una arquitectura basada en eventos con un solo hilo que gestiona las operaciones de manera asíncrona [12]. En este proyecto se ha utilizado, junto con el framework Express.js, para realizar la parte del Backend del sistema.

#### 8.1.2.1.3 Express.js

Express.js es un framework web rápido y minimalista para Node.js, que permite crear aplicaciones web y APIs de manera sencilla y eficiente. Es uno de los frameworks más populares y ampliamente utilizados en el ecosistema de Node.js debido a su enfoque simple y flexible. Ha sido utilizado en el contexto de este proyecto, junto con Node.js, para construir la API Rest del sistema.

### 8.1.2.2 HTML

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web [13]. Se ha utilizado su quinta versión para la definición de las plantillas de cada uno de los componentes de Angular.



### 8.1.2.3 CSS

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML, describiendo como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios [14]. Se usa para definir los estilos de las plantillas de los componentes de Angular.

## 8.1.3 Herramientas y programas usados para el desarrollo

### 8.1.3.1 Visual Studio Code

Visual Studio Code (también conocido como VS Code) es un editor de código fuente desarrollado por Microsoft. Es una herramienta altamente popular y ampliamente utilizada por desarrolladores de software en diversas plataformas, incluyendo Windows, macOS y Linux. Una de sus principales características es la posibilidad de añadir extensiones que faciliten el desarrollo de software y ha sido utilizado para realizar todas las partes del proyecto.

### 8.1.3.2 GitHub (Git)

Git es un sistema de control de versiones distribuido ampliamente utilizado para el seguimiento de cambios en archivos y la colaboración en proyectos de desarrollo de software. Fue creado por Linus Torvalds en 2005 y se ha convertido en una herramienta fundamental en el desarrollo de software moderno. Se ha utilizado junto con la plataforma GitHub, donde se ha hospedado remotamente el repositorio del proyecto.

### 8.1.3.3 Postman

Es una herramienta que permite a los desarrolladores probar, documentar y compartir APIs de manera eficiente. Es una herramienta popular utilizada en el ciclo de desarrollo de APIs y es ampliamente utilizada por desarrolladores y equipos de desarrollo de software. Se ha utilizado para probar que todos puntos de acceso (endpoints) de la API Rest del sistema se comportaban según lo esperado y que, realizaban las funciones correctamente.

### 8.1.3.4 PgAdmin 4

PgAdmin 4 es una herramienta de administración y desarrollo de bases de datos para PostgreSQL, un sistema de gestión de bases de datos relacional de código abierto y altamente potente. Proporciona una interfaz gráfica de usuario para administrar, monitorear y desarrollar bases de datos PostgreSQL de manera eficiente.





### 8.1.3.5 Angular Material

Angular Material es una biblioteca de componentes UI (User Interface) desarrollada específicamente para Angular, un framework de JavaScript ampliamente utilizado para construir aplicaciones web de una sola página (SPA). Angular Material proporciona una serie de componentes predefinidos y estilizados que siguen las directrices de diseño de Material Design, creado por Google.

### 8.1.3.6 Socket.io

Socket.io [7] es una biblioteca de JavaScript de código abierto que permite la comunicación en tiempo real entre clientes y servidores web. Proporciona una capa de abstracción sobre las API de WebSocket y ofrece una forma sencilla de establecer conexiones bidireccionales entre un cliente y un servidor.

Socket.io es especialmente útil para aplicaciones que requieren actualizaciones en tiempo real, por lo que encaja a la perfección para el propósito de este trabajo.

## 8.2 CSI 2: GENERACIÓN DEL CÓDIGO DE LOS COMPONENTES Y PROCEDIMIENTOS

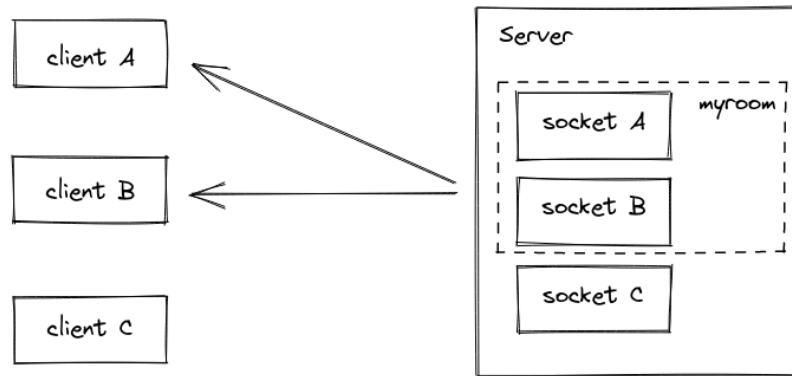
---

### 8.2.1 Problemas encontrados

#### 8.2.1.1 Uso de Socket.io

El primer problema encontrado fue la complejidad de diseñar un flujo de eventos que hagan posible la ejecución del cuestionario en línea, entre el cliente del profesor y de los alumnos. A base de realizar diagramas a papel y lápiz y hacer muchas pruebas fui capaz de hallar una solución que lo permitiese. A partir de ahí, se fue perfeccionando y puliendo para paliar varios errores que han ido surgiendo a lo largo del desarrollo. En la sección posterior, se explicarán los ficheros, tanto del servidor como de la aplicación, que se coordinan para llevar a cabo los cuestionarios en línea.

El siguiente reto que se me surgió fue hacer que dos cuestionarios no se solaparan, debido a que, dos profesores pueden lanzar dos cuestionarios diferentes, pero el código que lleva a cabo el juego es el mismo, es decir, escuchan y emiten los mismos eventos. Por suerte, la librería **socket.io** tiene una solución sencilla para esto, las habitaciones o *rooms*, que permiten a la parte del servidor emitir eventos solo a un subconjunto de clientes. El siguiente ejemplo sacado de la documentación oficial lo explica [15].



*Ilustración 57 Rooms en Socket.io*

Esto permite que un alumno envíe una respuesta al servidor, y este se la proporcione solo al profesor que corresponde. Cada habitación tiene un identificador único, que, para este caso en concreto, se trata del identificador de la entidad Juego de la base de datos. Gracias a esta aproximación, los cuestionarios en línea se pueden llevar a cabo de forma independiente unos de otros.

Más adelante, se me ocurrió probar qué pasaba si un usuario dentro de un juego salía o recargaba la página. Como cabe esperar, el sistema no estaba preparado para eso, haciendo que, si el profesor realizaba alguna de esas dos opciones, el juego se colgaba y no seguía, mientras que si lo hacía un alumno perdía la posibilidad de seguir participando en el cuestionario. Las soluciones propuestas fueron:

1. En el caso de que el profesor recargue la página o se vaya, emite un evento a todos los alumnos de que se ha terminado el cuestionario, y elimina el juego para que no quede rastro de él.
2. Por el contrario, si un alumno se va o recarga la página, se le permitirá recuperar la sesión y reincorporarse al cuestionario, con la única salvedad de que, si hay una pregunta en curso, no podrá responder hasta la siguiente.

Otro problema ocurrió a la hora de desplegar el sistema en un servidor, la comunicación con **socket.io** dio problemas, cayendo en uno de los riesgos que se habían identificado. Dicho problema fue que la librería, en entornos de producción, solo trabaja con el protocolo **HTTPS**, por lo que se ha tenido que configurar, tanto el servidor web, como la aplicación web, para utilizarlo, haciendo uso de un certificado ssl auto firmado.

Para finalizar, el último problema relacionado con **socket.io** fue realizar pruebas de carga y e2e para el caso de uso de jugar un cuestionario en línea. Esto está comentado en [Pruebas e2e automatizadas](#) y [Pruebas de Rendimiento](#).

### *8.2.1.2 Inconsistencias a la hora de borrar entidades*

En un momento del desarrollo, se llegó a la conclusión de que borrar ciertas entidades de la base de datos podía conllevar a inconsistencias en el sistema. Por ejemplo, los cuestionarios están



compuestos de al menos una pregunta, por lo que si se borra la única pregunta que componía un cuestionario, este quedaría vacío, lo que sería un estado erróneo. Por otro lado, los resultados de los alumnos a los cuestionarios hacen uso de estas preguntas, haciendo que si se eliminan preguntas de un cuestionario que haya sido jugado, esas preguntas no saldrían a la hora de hacer la revisión de un cuestionario. La solución propuesta es sencilla, negar la eliminación de entidades que estén relacionadas con otras, así no se llegará nunca a ningún estado inconsistente en el sistema.

## 8.2.2 Descripción detallada de las clases

En este apartado se recogerán y detallarán la funcionalidad de los archivos que implementan la librería **socket.io**, tanto del servidor web como de la aplicación, ya que son los que permiten realizar la funcionalidad principal del sistema, los cuestionarios en línea.

Además, se ha generado documentación para ambos subsistemas utilizando **Compodoc** para generar la de la aplicación web y **apiDoc** para generar la del servidor web, ambas ubicadas en el directorio **docs** de sus respectivos proyectos.

### 8.2.2.1 *game.socket.ts*

Esta clase, ubicada en el servidor web, es la encargada de hacer de intermediario entre los profesores y los alumnos a la hora de jugar un cuestionario en línea. Por poner un ejemplo, cada vez que un alumno selecciona una respuesta, la aplicación web la envía al servidor, este fichero lo procesa y se la devuelve al profesor.

El fichero está compuesto de una serie de *listeners* (escuchadores) que quedan a la espera de eventos emitidos desde un cliente para procesarlos y devolver una respuesta. Esto se lleva a cabo mediante **sockets**, que son una interfaz de programación que permite la comunicación entre procesos o aplicaciones a través de una red, intercambiando datos que se pueden enviar y recibir de manera bidireccional. Como se ha comentado en el apartado anterior, es necesario que los sockets estén en habitaciones para que no haya comunicaciones indeseadas.

A continuación, se detallarán cada uno de estos eventos:

- **Crear juego:** el servidor recibe del profesor el juego con el cuestionario que se va a jugar, crea una sesión y se la devuelve al profesor y emite un juego a todos los alumnos que estén a la escucha.
- **Unirse a un juego:** los alumnos envían su usuario y el id del juego. El servidor añadirá al usuario a la sesión correspondiente, devolverá la sesión al alumno y enviará al profesor correspondiente el usuario que acaba de entrar.
- **Previsualización de la pregunta:** el evento consiste simplemente en que el profesor envía la señal de que se previsualice la pregunta y el servidor se encarga de comunicárselo a todos los alumnos.



- **Empezar el tiempo de la pregunta:** igual que el anterior. El cliente del profesor comunica a los alumnos que se debe mostrar la pregunta.
- **Finalizar pregunta:** el cliente del profesor comunica que se ha acabado el tiempo de la pregunta para que el cliente de los alumnos muestre el resultado.
- **Mostrar puntuación:** el cliente del profesor comunica que se muestre en el cliente de los alumnos la puntuación que llevan hasta el momento.
- **Enviar respuesta:** el cliente del alumno envía la respuesta al cliente del profesor.
- **Desconectar:** maneja los usuarios que se desconectan de un socket.
- **Unirse al socket de un curso:** permite a los sockets de los alumnos entrar en la habitación correspondiente a los cursos a los que pertenece.
- **Finalizar el juego:** el cliente del profesor comunica al cliente de los alumnos que el juego a finalizado.
- **Salir del juego:** el socket del cliente que envía el evento sale de la habitación con el identificador correspondiente.

### 8.2.2.2 *host-game.component.ts*

Este es el componente donde se desarrolla un cuestionario en línea por parte del profesor. Mediante los eventos definidos, va ejecutando el flujo del juego, haciendo que un juego pase por los siguientes estados:

- **Esperando alumnos:** en la pantalla del profesor van apareciendo los nombres de los alumnos que se vayan uniendo. Una vez que haya al menos uno, el profesor podrá empezar el juego.
- **Previsualización de la pregunta:** Al comenzar el juego, se mostrará un contador y el enunciado de la siguiente pregunta.
- **Visualización de la pregunta:** Cuando termina la previsualización de la pregunta, se muestra la pregunta entera, con el enunciado, la imagen si la tuviera y cada una de sus respuestas.
- **Resultado de la pregunta:** Cuando finalice el tiempo de la pregunta, se muestra la respuesta correcta.
- **Visualización del podio:** Una vez enseñada la respuesta correcta, la pantalla muestra a los tres alumnos con mejor puntuación. Si quedan preguntas en el cuestionario, vuelve a la fase de Previsualización de la pregunta.
- **Juego finalizado:** Si no quedan preguntas, el juego se termina.

Cabe decir que cada vez que el cuestionario cambia de estado, se envía un evento al servidor para comunicárselo a cada uno de los alumnos que participan y actualizar el cliente de estos.

### 8.2.2.3 *student-game.component.ts*

Este componente permite a los alumnos conectarse a un cuestionario y jugarlo. Es un componente que básicamente solo tiene eventos de escucha, pues ejecuta el juego de manera pasiva, según la

información que le llega de la parte del cliente del profesor. El juego, en este componente pasa por las mismas fases que en el componente del profesor, salvando dos:

- Cuando en el cliente del profesor se muestra la respuesta correcta, al alumno se le muestra si ha respondido correctamente o no.
- Cuando en el cliente del profesor se muestra el podio, el alumno puede ver la puntuación que lleva hasta ese momento.

## 8.3 CSI 3: EJECUCIÓN DE LAS PRUEBAS UNITARIAS

### 8.3.1 Herramientas utilizadas

#### 8.3.1.1 Jest

Jest es un framework de testing de JavaScript diseñado para asegurar el correcto funcionamiento de cualquier código JavaScript. Está construido sobre Jasmine, que se comentará más adelante, y está mantenido por Meta. Fue diseñado por Christoph Nakazawa con un enfoque en la simplicidad y el soporte para grandes aplicaciones web [16]. Se utilizará para realizar y automatizar los tests unitarios de los controladores de la API Rest.

#### 8.3.1.2 Jasmine

Jasmine es un marco de prueba de código abierto para JavaScript. Su objetivo es ejecutarse en cualquier plataforma habilitada para JavaScript, no entrometerse en la aplicación ni en el entorno de desarrollo, y tener una sintaxis fácil de leer [17]. Para aplicaciones Angular se suele utilizar junto con el *test-runner* Karma, que es un módulo que permite automatizar algunas de las tareas de las suites de testing como Jasmine [18].

### 8.3.2 Resultado de las pruebas

#### 8.3.2.1 Controladores

```
Test Suites: 5 passed, 5 total
Tests:      209 passed, 209 total
Snapshots:  0 total
Time:       7.677 s, estimated 12 s
```

*Ilustración 58 Total de tests unitarios de la API Rest*

### 8.3.2.1.1 Controlador de usuarios

```
PASS ./userController.test.ts
Users
  Get users
    Exist users
      ✓ should retrieve all users (77 ms)
    Not exist users
      ✓ should retrieve an empty array (1 ms)
  Get user by id
    there are at least one user
      ✓ should retrieve the user with the id given (5 ms)
    There are no users
      ✓ should retrieve null (2 ms)
  Create user
    With valid input data
      ✓ User created successfully (256 ms)
      ✓ should assign role to the user correctly (177 ms)
    Password must be encrypted
      ✓ password shouldn't be test (170 ms)
    The user already exists
      ✓ should return a 400 (3 ms)
  Try to create a user with wrong fields
    With empty field
      ✓ should return 500 (1 ms)
    With wrong data types
      ✓ should return 500 (1 ms)
  Login
    Success
      ✓ when valid credentials (53 ms)
    Failure
      ✓ when invalid credentials (51 ms)
      ✓ when user does not exists (1 ms)
      ✓ when invalid input (1 ms)
  Update user
    User exists
      ✓ should update username, password and role (105 ms)
      ✓ should update with existing values with empty values (104 ms)
    User not exists
      ✓ should return 404
```

*Ilustración 59 Tests unitarios del controlador de usuarios 1*

```
Delete user
  Exists user
    ✓ should delete with valid id (3 ms)
  Failure
    User not exists
      ✓ should return 404 (1 ms)
    When providing an invalid id
      Should return 400
        ✓ with no id
        ✓ with invalid id
  Import users
    Valid csv file
      ✓ File is uploaded and users are successfully created (63 ms)
    Invalid file
      ✓ should return 500 if empty file (52 ms)
      ✓ should return 500 if file has an invalid format (65 ms)
      ✓ should return 500 if file has missing fields (52 ms)
  Get all course's users
    Valid course id
      Course has no users
        ✓ should return an empty array (7 ms)
      Course has users
        ✓ should retrieve all course's users giving course id (6 ms)
    Invalid course id
      ✓ should return 400 when course id is not a number (1 ms)
      ✓ should return 400 when course id is not provided
  Change user password
    User exists
      The password is successfully changed
        ✓ should update the password if the users exists and the actual password matches (154 ms)
      Password does not match
        ✓ should return 400 (53 ms)
    Invalid parameters
      ✓ should returns an error when parameters are missing (11 ms)
      ✓ should return an error if body is missing or incomplete (4 ms)
  User does not exists
    ✓ should return 404
```

*Ilustración 60 Tests unitarios del controlador de usuarios 2*

```
Delete user from course
  With valid parameters
    User belongs to course
      ✓ should delete the user form course (5 ms)
    User does not belong to course
      ✓ should return an error (3 ms)
  With invalid parameters
    ✓ should return 500 if user id is not a number (1 ms)
    ✓ should return 404 if user does not exists (1 ms)
  User exists
    ✓ should return 500 if course id is not a number (2 ms)
    ✓ should return 404 if course does not exists (1 ms)
```

*Ilustración 61 Tests del controlador de usuarios 3*

### 8.3.2.1.2 Controlador de cursos

```
PASS ./courseController.test.ts
Courses
  Get courses
    Exist courses
      ✓ should retrieve all courses (72 ms)
    Not exist courses
      ✓ should retrieve an empty array (1 ms)
  Get course by id
    there are at least one course
      ✓ should retrieve the course with the id given (8 ms)
    There are no courses
      ✓ should retrieve null (1 ms)
  Create course
    With valid input data
      ✓ Course created successfully (5 ms)
    The course already exists
      ✓ should return a 400 (3 ms)
    Try to create a course with wrong fields
      With empty field
        ✓ should return 500 (11 ms)
      With wrong data types
        ✓ should return 500 (7 ms)
  Update course
    Course exists
      ✓ should update name and description (5 ms)
      ✓ should update with existing values with empty values (4 ms)
    Course not exists
      ✓ should return 404 (1 ms)
  Delete course
    Exists course
      ✓ should delete with valid id (2 ms)
  Failure
    Course not exists
      ✓ should return 404
    When providing an invalid id
      Should return 400
        ✓ with no id
        ✓ with invalid id
```

*Ilustración 62 Tests del controlador de cursos 1*



```
Import courses
Valid csv file
  ✓ File is uploaded and courses are successfully created (13 ms)
Invalid file
  ✓ should return 500 if empty file (3 ms)
  ✓ should return 500 if file has an invalid format (5 ms)
  ✓ should return 500 if file has missing fields (2 ms)
Add users to course
Users can be added
  ✓ users should be added to the course (3 ms)
Users can't be added
  Users array is empty
    ✓ should return an error (1 ms)
  Course does not exist
    ✓ should return an error (2 ms)
  At least one user does not exist
    ✓ should return an error (1 ms)
Add surveys to course
Surveys can be added
  ✓ surveys should be added to the course (3 ms)
Surveys can't be added
  Surveys array is empty
    ✓ should return an error (1 ms)
  Course does not exist
    ✓ should return an error (1 ms)
  At least one survey does not exist
    ✓ should return an error (1 ms)
```

**Ilustración 63 Tests del controlador de cursos 2**

```
Add questions to course
Questions can be added
  ✓ questions should be added to the course (3 ms)
Questions can't be added
  Questions array is empty
    ✓ should return an error (1 ms)
  Course does not exist
    ✓ should return an error
  At least one question does not exist
    ✓ should return an error (1 ms)
Get all users's course
Valid user id
  User has no courses
    ✓ should return an empty array (2 ms)
  User has courses
    ✓ should retrieve all user's courses giving user id (5 ms)
Invalid user id
  ✓ should return 400 when user id is not a number
  ✓ should return 400 when user id is not provided (1 ms)
```

**Ilustración 64 Tests del controlador de cursos 3**



### 8.3.2.1.3 Controlador de preguntas

```
PASS ./questionController.test.ts
Questions
  Get questions
    Exist questions
      ✓ should retrieve all questions (84 ms)
    Not exist questions
      ✓ should retrieve an empty array (2 ms)
  Get question by id
    there are at least one question
      ✓ should retrieve the question with the id given (7 ms)
    There are no questions
      ✓ should retrieve null (2 ms)
  Get question by user
    there are at least one question
      ✓ should retrieve the questions with the id given (8 ms)
    There are no questions
      ✓ should retrieve an empty array (2 ms)
  Create question
    With valid input data
      User creator exists
        ✓ Question created successfully (48 ms)
      User creator does not exist
        ✓ should return 404 (5 ms)
    Try to create a question with wrong fields
      ✓ should return 500 if any field is undefined (2 ms)
      ✓ should return 500 if questions has no answers (4 ms)
      ✓ should return 500 if answer time is less than 5 seconds (4 ms)
    Multioption questions
      ✓ should return 500 if question has more than 4 answers (4 ms)
      ✓ should return 500 if question has less than 4 answers (3 ms)
      ✓ should return 500 if more than one answer is correct (4 ms)
      ✓ should return 500 if any answer is correct (4 ms)
    True/false questions
      ✓ should return 500 if question has more than 2 answers (5 ms)
      ✓ should return 500 if question has less than 2 answers (3 ms)
      ✓ should return 500 if more than one answer is correct (3 ms)
      ✓ should return 500 if question has no correct answers (3 ms)
    Short questions
      ✓ should return 500 if question has more than 4 answers (9 ms)
      ✓ should return 500 if not all answers are correct (3 ms)
    With wrong data types
      ✓ should return 500 (30 ms)
```

*Ilustración 65 Tests del controlador de preguntas 1*



```
Delete question
Exists question
  ✓ should delete the question and all the answers with valid question id (5 ms)
Failure
Question not exists
  ✓ should return 404 (1 ms)
When providing an invalid id
  Should return 400
    ✓ with no id (1 ms)
    ✓ with invalid id
Import questions
Valid csv file
  ✓ File is uploaded and questions are successfully created (20 ms)
Invalid file
  ✓ should return 500 if empty file (4 ms)
  ✓ should return 500 if file has an invalid format (3 ms)
  ✓ should return 500 if file has missing fields (4 ms)
Export questions
  ✓ should return error if user does not exist (1 ms)
User exists
  User has no questions
    ✓ should return a empty csv if user does not have questions (4 ms)
  User has at least one question
    ✓ should return a csv if users and questions exists (5 ms)
Update question
Question exists
  ✓ should update description, type, time, and answers (10 ms)
  ✓ should update with existing values with empty values (10 ms)
  ✓ should return error if question has wrong format (5 ms)
  ✓ should return error if answers is empty (3 ms)
Question not exists
  ✓ should return 404
```

### *Ilustración 66 Tests del controlador de preguntas 2*

```
Get all course's questions
Valid course id
  Course has no questions
    ✓ should return an empty array (4 ms)
  Course has questions
    ✓ should retrieve all course's questions giving course id (8 ms)
Invalid course id
  ✓ should return 400 when course id is not a number
  ✓ should return 400 when course id is not provided
Delete question from course
With valid parameters
  Question belongs to course
    ✓ should delete the question form course (3 ms)
  Question does not belong to course
    ✓ should return an error (2 ms)
With invalid parameters
  ✓ should return 500 if question id is not a number (7 ms)
  ✓ should return 404 if question does not exist (1 ms)
Question exists
  ✓ should return 500 if course id is not a number (1 ms)
  ✓ should return 404 if course does not exist (1 ms)
```

### *Ilustración 67 Tests del controlador de preguntas 3*

### 8.3.2.1.4 Controlador de cuestionarios

```
PASS ./surveyController.test.ts
Surveys
  Get surveys
  Exist surveys
    ✓ should retrieve all surveys (79 ms)
  Not exist surveys
    ✓ should retrieve an empty array (1 ms)
  Get surveys by user
    there are at least one survey
      ✓ should retrieve the surveys with the user id given (12 ms)
    There are no surveys
      ✓ should retrieve an empty array (2 ms)
  When providing an invalid id
    Should return 400
      ✓ with no id (1 ms)
      ✓ with invalid id (1 ms)
  Get survey by id
    there are at least one survey
      ✓ should retrieve the survey with the id given (6 ms)
    There are no surveys
      ✓ should retrieve null (1 ms)
  When providing an invalid id
    Should return 400
      ✓ with no id (1 ms)
      ✓ with invalid id (1 ms)
  Get all course's surveys
  Valid course id
    Course has no surveys
      ✓ should return an empty array (5 ms)
    Course has surveys
      ✓ should retrieve all course's surveys giving course id (8 ms)
  Invalid course id
    ✓ should return 400 when course id is not a number
    ✓ should return 400 when course id is not provided (1 ms)
```

*Ilustración 68 Tests del controlador de cuestionarios 1*

```
Create survey
  With valid input data
    User creator exists
      ✓ Survey created successfully (5 ms)
    User creator does not exist
      ✓ should return error (1 ms)
  Try to create a user with wrong fields
  With missing field
    ✓ should return 500 (1 ms)
  With wrong data types
    ✓ should return 500 (31 ms)
  Without any question attached
    ✓ should return an error (1 ms)
Update survey
  Survey exists
    ✓ should update title and questions (13 ms)
    ✓ should return error if there is no questions (9 ms)
  Survey not exists
    ✓ should return 404 (1 ms)
Delete survey
  Exists survey
    ✓ should delete the survey (3 ms)
  Failure
  Survey not exists
    ✓ should return 404 (12 ms)
  When providing an invalid id
    Should return 400
      ✓ with no id
      ✓ with invalid id (1 ms)
```

*Ilustración 69 Tests del controlador de cuestionarios 2*

```
Delete surveys from course
With valid parameters
  User belongs to course
    ✓ should delete the question form course (4 ms)
  Survey does not belong to course
    ✓ should return an error (6 ms)
With invalid parameters
  ✓ should return 500 if course id is not a number (15 ms)
  ✓ should return 404 if course does not exist (1 ms)
Course does exists
  ✓ should return 404 if survey does not exist (1 ms)
  ✓ should return 500 if survey id is not a number (5 ms)
```

### *Ilustración 70 Tests del controlador de cuestionarios 3*

#### 8.3.2.1.5 Controlador de juegos

```
PASS ./gameController.test.ts
Game
  Create game
    With valid input data
      User host and survey attached exists
        ✓ game created successfully (29 ms)
      User creator does not exist
        ✓ should return error (2 ms)
      Survey attached does not exist
        ✓ should return error (2 ms)
    Try to create a game with wrong fields
      With missing field
        ✓ should return 500 (1 ms)
      With wrong data types
        ✓ should return 500 (207 ms)
  Update state
    Game exists
      ✓ should update state (20 ms)
    Failure
      ✓ should return error if empty params (3 ms)
      ✓ should return error if wrong id (2 ms)
    Game not exists
      ✓ should return 404 (6 ms)
  Delete game
    Exists game
      ✓ should delete the game (2 ms)
    Failure
      Survey not exists
        ✓ should return 404 (1 ms)
      When providing an invalid id
        Should return 400
          ✓ with no id
          ✓ with invalid id (1 ms)
```

### *Ilustración 71 Tests del controlador de juegos 1*

```
Get game by id
there are at least one game
  ✓ should retrieve the game with the id given (2 ms)
There are no games
  ✓ should retrieve null (1 ms)
When providing an invalid id
  Should return 400
    ✓ with no id
    ✓ with invalid id
Get open or started games by courses
Success
  ✓ should return all the games open or started from the courses given and not the closed one (6 ms)
Failure
  ✓ should return error if courses is missing
  ✓ should return error if courses is empty
Create results
Success
  ✓ should create the results correctly (8 ms)
Failure
  ✓ should return an error if body has wrong format (3 ms)
  ✓ should return error if body is empty
Get game results by user
there are at least one result
  ✓ should retrieve the results with the user id given (4 ms)
There are no results
  ✓ should retrieve an empty array (3 ms)
When providing an invalid user id
  Should return error
    ✓ 400 with no id
    ✓ 400 with invalid id (1 ms)
    ✓ 404 user not exists (1 ms)
```

### *Ilustración 72 Tests del controlador de juegos 2*

```
Get games by course
there are at least one game
  ✓ should retrieve the games with the course id given (2 ms)
There are no games
  ✓ should retrieve an empty array (3 ms)
When providing an invalid id
  ✓ should return 404 if course does not exists (1 ms)
  Should return 400
    ✓ with no id
    ✓ with invalid id
Get game result by user and game
The result exists
  ✓ should retrieve the results with the game and user id given (6 ms)
The result does not exist
  ✓ should retrieve null (5 ms)
When providing an invalid game id
  Should return error
    ✓ 400 with no id (1 ms)
    ✓ 400 with invalid id (1 ms)
  User exists
    ✓ 404 if game not exists (2 ms)
When providing an invalid user id
  Should return error
    ✓ 400 with no id
    ✓ 400 with invalid id (1 ms)
    ✓ 404 if user not exists (1 ms)
Get game results by game
there are at least one result
  ✓ should retrieve the results with the game id given (3 ms)
There are no results
  ✓ should retrieve an empty array (1 ms)
When providing an invalid game id
  Should return error
    ✓ 400 with no id (1 ms)
    ✓ 400 with invalid id
    ✓ 404 if game not exists (1 ms)
```

### *Ilustración 73 Tests del controlador de juegos 3*



```
Get game results by user and course
The are at least one result
  ✓ should retrieve the results with the course and user id given (5 ms)
There is no results
  ✓ should retrieve an empty array (3 ms)
When providing an invalid course id
Should return error
  ✓ 400 with no id
  ✓ 400 with invalid id (1 ms)
User exists
  ✓ 404 if course not exists (1 ms)
When providing an invalid user id
Should return error
  ✓ 400 with no id
  ✓ 400 with invalid id (1 ms)
  ✓ 404 if user not exists
```

*Ilustración 74 Tests del controlador de juegos 4*

### *8.3.2.2 Componentes y servicios de la aplicación web*

En total se han ejecutado 213 pruebas, ninguna fallida y finalizado en 1.362 segundos. A continuación, se muestra una captura de la ejecución de los tests. Que los tests estén en verde significa que la prueba ha sido satisfactoria.



```
AppComponent
  • should create the app
ErrorPageComponent
  • should create
LoginFormComponent
  • should create
  • form invalid when empty
  • username and password fields validity
  • submitting admin user emits correctly and navigates to the correct page
  • submitting professor user emits correctly and navigates to the correct page
  • submitting student user emits correctly and navigates to the correct page
CourseListComponent
  • should fetch courses on initialization
  • should show the courses
  • onCreateCourse should navigate to /courses/create
  • should delete course
  • should detect file input change
  • file change event should arrive in handler
CreateCourseFormComponent
  • should create
  • form invalid when empty
  • name and description fields validity
  • submitting a form emits correctly
SocketIoService
  • should be created
CreateUserFormComponent
  • should create
  • form invalid when empty
  • username field validity
  • submitting a form emits correctly
UpdateCourseFormComponent
  • should create
  • should retrieve course
  • submitting a form emits correctly
UpdateUserFormComponent
  • should create
  • should retrieve user
  • submitting a form emits correctly
UserListComponent
  • should fetch users on initialization
  • should show the users
  • onCreateUser should navigate to /user/create
  • should delete user
  • should detect file input change
  • file change event should arrive in handler
CourseProfessorDetailsComponent
  • should create
  • should fetch course, users, questions, students and games played
  • should show the entities fetches
CreateQuestionComponent
  • should create
  • should show error if any field required is missing
  • should show error if there is no correct answer
  • should show error if limit time is less than 5 seconds
  • file change event should arrive in handler
```

**Ilustración 75 Tests unitarios de la aplicación web 1**



```
CreateSurveyComponent
• should create
• title invalid when empty
• survey invalid when no questions added
• submitting a survey emits correctly
• file change event should arrive in handler

HostGameComponent
• should create
• should show user list
• should show question preview
• should show question screen
• should show question result
• should show leaderboard screen
• leave game should navigate to /professor/home

LibraryComponent
• should create

QuestionListComponent
• should create
• should fetch questions on initialization
• should show the questions
• createNewQuestion should navigate to /questions/create
• should delete question
• should detect file input change
• file change event should arrive in handler

SurveyListComponent
• should create
• should fetch surveys on initialization
• should show the surveys
• createNewSurvey should navigate to survey/create
• should delete survey

UpdateQuestionComponent
• should create

UpdateSurveyComponent
• should create

FooterComponent
• should create
• should contain author's name

HeaderComponent
• should create
• should show admin links
• should show professor links
• should show student links

CourseStudentDetailsComponent
• should create
• should fetch results on initialization
• should show the results
• When survey click should navigate to /game/revision/:game_id/:user_id
```

### *Ilustración 76 Tests unitarios de la aplicación web 2*

```
StudentGameComponent
• should create
• should show wait room
• should show preview
• should show question screen
• should show correct if answer is correct
• should show incorrect if answer is incorrect
• should show incorrect if user does not have answered
• should show score
• leave game should navigate to /student/home

StudentHomeComponent
• should create
• should fetch courses on initialization
• should show the courses
• should fetch open games on initialization
• should show the opened games
• When course click should navigate to /student/course/details/:id
• When game click should navigate to /game/join/:id

StudentProfileComponent
• should create
• should fetch user results on initialization
• should show the results
• form invalid when empty
• username and password fields validity
• submitting correctly with both password
```

### *Ilustración 77 Tests unitarios de la aplicación web 3*





```
ApiAdminService
• should be created
createUser()
• should return success message
• should return an error message if something wrong occurred
createCourse()
• should return success message
• should return an error message if something wrong occurred
getUsers()
• should return all users
• should return an error message if something wrong occurred
getUser()
• should return user giving id
• should return an error message if something wrong occurred
getCourses()
• should return all courses
• should return an error message if something wrong occurred
getCourse()
• should return course giving id
• should return an error message if something wrong occurred
deleteUser()
• should return success message
• should return an error message if something wrong occurred
deleteCourse()
• should return success message
• should return an error message if something wrong occurred
updateUser()
• should return success message
• should return an error message if something wrong occurred
updateCourse()
• should return success message
• should return an error message if something wrong occurred
uploadUserFile()
• should return success message
• should return an error message if something wrong occurred
uploadCourseFile()
• should return success message
• should return an error message if something wrong occurred
addUserToCourse()
• should return success message
• should return an error message if something wrong occurred
addSurveyToCourse()
• should return success message
• should return an error message if something wrong occurred
addQuestionToCourse()
• should return success message
• should return an error message if something wrong occurred
```

*Ilustración 78 Tests unitarios de la aplicación web 4*



```
ApiAuthService
• should be created
login()
• should return user if login was successful
logout()
• should navigate to /login
ApiProfessorService
• should be created
createQuestion()
• should return success message
• should return an error message if something wrong occurred
getQuestions()
• should return all questions
• should return an error message if something wrong occurred
getQuestionsByUser()
• should return user's questions
• should return an error message if something wrong occurred
getQuestionById()
• should return user's courses
• should return an error message if something wrong occurred
createSurvey()
• should return success message
• should return an error message if something wrong occurred
getSurveysByUser()
• should return user's surveys
• should return an error message if something wrong occurred
getSurveyById()
• should return survey giving its id
• should return an error message if something wrong occurred
getUsersByCourse()
• should return course's users
• should return an error message if something wrong occurred
getSurveysByCourse()
• should return course's surveys
• should return an error message if something wrong occurred
getQuestionsByCourse()
• should return course's questions
• should return an error message if something wrong occurred
createGame()
• should return game created
• should return an error message if something wrong occurred
updateGame()
• should return game updated
• should return an error message if something wrong occurred
getGameById()
• should return game giving id
• should return an error message if something wrong occurred
```

*Ilustración 79 Tests unitarios de la aplicación web 5*

```
createUserResults()
  • should return success message
  • should return an error message if something wrong occurred
importQuestions()
  • should return success message
  • should return an error message if something wrong occurred
updateQuestion()
  • should return success message
  • should return an error message if something wrong occurred
deleteQuestion()
  • should return success message
  • should return an error message if something wrong occurred
updateSurvey()
  • should return success message
  • should return an error message if something wrong occurred
deleteSurvey()
  • should return success message
  • should return an error message if something wrong occurred
deleteSurveyFromCourse()
  • should return success message
  • should return an error message if something wrong occurred
deleteQuestionFromCourse()
  • should return success message
  • should return an error message if something wrong occurred
deleteUserFromCourse()
  • should return success message
  • should return an error message if something wrong occurred
deleteGame()
  • should return success message
  • should return an error message if something wrong occurred
getGameResultsByGame()
  • should return game results
  • should return an error message if something wrong occurred
getGamesByCourse()
  • should return course's games
  • should return an error message if something wrong occurred
```

### *Ilustración 80 Tests unitarios de la aplicación web 6*

```
ApiUserService
  • should be created
getCoursesByUser()
  • should return user's courses
  • should return an error message if something wrong occurred
getOpenOrStartedGamesByCourses()
  • should return open or started games giving course
  • should return an error message if something wrong occurred
changePassword()
  • should return a success message
  • should return an error message if something wrong occurred
getGamesResultsByUser()
  • should return the game results giving user
  • should return an error message if something wrong occurred
getGamesResultsByUserAndCourse()
  • should return the game results giving user and course
  • should return an error message if something wrong occurred
getGamesResultByUserAndGame()
  • should return the game result giving user and game
  • should return an error message if something wrong occurred
GameDetailsComponent
  • should create
  • should fetch user results
  • should show user results
AnswersComponent
  • should create
LinkQuestionCourseComponent
  • should create with question list component
CreateGameDialogComponent
  • form invalid when empty
  • point type and questions visible fields validity
  • submitting a form emits correctly
LinkSurveyCourseComponent
  • should create with survey list component
LinkUserCourseComponent
  • should create with user list component
GameRevisionComponent
  • should create
  • should fetch result on initialization
  • should show the questions with the result retrieved
```

### *Ilustración 81 Tests unitarios de la aplicación web 7*



## 8.4 CSI 4: EJECUCIÓN DE LAS PRUEBAS DE INTEGRACIÓN

### 8.4.1 Herramientas utilizadas

Para realizar las pruebas automatizadas e2e se ha utilizado el software TestCafé. Este software es un framework de testing e2e para aplicaciones web que corre con Node.js [12]. Soporta los tres sistemas operativos principales, Linux, Windows y macOS y sirve para simular escenarios de nuestra aplicación de una manera rápida y sencilla [19].

Una de las características destacadas de TestCafé es que no requiere la instalación de complementos o extensiones en el navegador. En su lugar, utiliza una arquitectura basada en proxy para interactuar con las páginas web y realizar las pruebas. Esto significa que las pruebas se ejecutan de forma aislada y no se ven afectadas por las extensiones del navegador o el estado de inicio de sesión del usuario.

### 8.4.2 Resultado de las pruebas

#### 8.4.2.1 Caso de uso común

Inicio de sesión		
Caso de prueba	Resultado esperado	Resultado final
Iniciar sesión con un usuario de tipo administrador existente y la contraseña coincide.	El sistema permite entrar al usuario y lo redirige a la pantalla principal del usuario administrador.	Correcto.
Iniciar sesión con un usuario de tipo profesor existente y la contraseña coincide.	El sistema permite entrar al usuario y lo redirige a la pantalla principal del usuario profesor.	Correcto.
Iniciar sesión con un usuario de tipo alumno existente y la contraseña coincide.	El sistema permite entrar al usuario y lo redirige a la pantalla principal del usuario alumno.	Correcto.
Se dejan en blanco el formulario de iniciar sesión.	No permite iniciar sesión.	Correcto.
Se introduce el nombre de un usuario existente pero la contraseña no coincide.	El sistema muestra un error diciendo que las credenciales no son correctas.	Correcto.
Se introduce el nombre de un usuario no existente.	El sistema muestra un error diciendo que las credenciales no son correctas.	Correcto.

Tabla 8 Casos de prueba Inicio de sesión

### 8.4.2.2 Casos de uso del administrador

Creación de usuarios		
Caso de prueba	Resultado esperado	Resultado final
Se introduce un nombre que no coincida con ningún otro y cualquier rol.	El sistema crea el usuario y muestra un mensaje. Además, generará una contraseña aleatoria y le enviará un correo al nombre de usuario añadiéndole "@uniovi.es" con sus credenciales.	Correcto, aunque el mensaje enviado no se puede verificar.
Se introduce un nombre que coincide con el de otro usuario existente.	El sistema no crea el usuario y muestra un mensaje de error.	Se detectó que no se mostraba el mensaje.
Se dejan el campo del nombre y el rol en blancos.	El sistema no permite crear el usuario.	Correcto.
Se introduce el nombre, pero no el rol.	El sistema no permite crear el usuario.	Correcto.

Modificación de usuarios		
Caso de prueba	Resultado esperado	Resultado final
Se introduce solo el nombre del usuario.	El sistema actualiza el nombre del usuario y muestra un mensaje.	Correcto.
Se introduce solo una nueva contraseña.	El sistema actualiza la contraseña del usuario y muestra un mensaje.	Correcto.
Se introduce solo el rol.	El sistema actualiza el rol del usuario y muestra un mensaje.	Correcto.
No se introduce ningún campo.	El sistema no actualiza ningún dato del usuario.	Correcto.

Eliminación de usuarios		
Caso de prueba	Resultado esperado	Resultado final
El usuario existe, no tiene ninguna entidad asociada y se intenta eliminar.	El sistema elimina el usuario y muestra un mensaje.	Correcto.
El usuario es un profesor con alguna pregunta, cuestionario creado.	El sistema no permite eliminarlo.	Correcto.
El usuario es un alumno con algún resultado existente.	El sistema no permite eliminarlo.	Correcto.



Creación de cursos		
Caso de prueba	Resultado esperado	Resultado final
Se introduce un nombre que no coincida con ningún otro y cualquier descripción.	El sistema crea el curso y muestra un mensaje.	Correcto.
Se introduce un nombre que coincide con el de otro curso existente.	El sistema no crea el curso y muestra un mensaje de error.	Correcto.
Se dejan el campo del nombre y la descripción en blancos.	El sistema no permite crear el curso.	Se detectó que dejaba crear cursos con parámetros en blanco.
Se introduce el nombre, pero no la descripción.	El sistema permite crear el curso.	Se detectó que dejaba crear cursos con parámetros en blanco.
Se introduce la descripción, pero no el nombre.	El sistema no permite crear el curso.	Se detectó que dejaba crear cursos con parámetros en blanco.

Modificación de cursos		
Caso de prueba	Resultado esperado	Resultado final
Se introduce solo el nombre del curso.	El sistema actualiza el nombre del curso y muestra un mensaje.	Correcto.
Se introduce solo una nueva descripción.	El sistema actualiza la descripción del curso y muestra un mensaje.	Correcto.
No se introduce ningún campo.	El sistema no actualiza ningún dato del curso.	Correcto.

Eliminación de cursos		
Caso de prueba	Resultado esperado	Resultado final
El curso existe, no tiene ninguna entidad asociada y se intenta eliminar.	El sistema elimina el curso y muestra un mensaje.	Correcto.
El curso tiene asociada alguna entidad y se intenta eliminar.	El sistema no permite eliminarlo.	Correcto.

### 8.4.2.3 Casos de uso del profesor

Creación de preguntas		
Caso de prueba	Resultado esperado	Resultado final
Se introduce un título, una duración de 5 segundos o más, se selecciona el tipo multi opción, se rellenan las 4 respuestas y se marca una como correcta.	El sistema crear la pregunta correctamente y muestra un mensaje.	Correcto.
Se introduce un título, una duración de 5 segundos o más, se selecciona el tipo	El sistema crear la pregunta correctamente y muestra un mensaje.	Correcto.



verdadero o falso, se rellenan las 2 respuestas y se marca una como correcta.		
Se introduce un título, una duración de 5 segundos o más, se selecciona el tipo respuesta corta y se rellenan al menos una respuesta.	El sistema crear la pregunta correctamente y muestra un mensaje.	Correcto.
Se dejan en blanco el formulario todos los campos.	El sistema muestra un mensaje de error diciendo que se deben rellenar todos los campos.	Correcto.
Se rellenan todos los campos, pero el tiempo es menor de 5 segundos.	El sistema muestra un error diciendo que el tiempo debe ser al menos de 5 segundos.	Se detecta que el sistema permitía crear preguntas con un tiempo inferior a 5 segundos.
Se rellenan todos los campos, se introduce un tiempo superior a 5, se selecciona el tipo de pregunta multi opción o verdadero o falso y no se marca ninguna respuesta como correcta.	El sistema muestra un error diciendo que se debe seleccionar una respuesta como correcta.	Correcto.

<b>Modificación de preguntas</b>		
<b>Caso de prueba</b>	<b>Resultado esperado</b>	<b>Resultado final</b>
El usuario selecciona una pregunta para editarla.	Al entrar a la pantalla, los campos deben estar completos con los datos de la pregunta.	Correcto.
El usuario intenta modificar el tipo de la pregunta e intenta actualizar.	El sistema no permite actualizar el tipo de la pregunta y muestra un mensaje de error al usuario indicando el motivo.	Correcto.
El usuario intenta modificar la respuesta correcta.	El sistema no permite actualizar la respuesta correcta la pregunta y muestra un mensaje de error al usuario indicando el motivo.	Correcto.
El usuario edita una pregunta y modifica cualquiera de los campos, estando todos rellenos y el tiempo superior a 5.	El sistema actualiza la pregunta y muestra un mensaje al usuario.	Correcto.
Se modifican los campos dejando algunos vacíos.	No permite actualizar la pregunta y muestra un mensaje de error indicando el motivo.	Correcto.
Se decreta el tiempo de la pregunta por debajo de 5 segundos.	No permite actualizar la pregunta y muestra un mensaje de error indicando el motivo.	Correcto.
Se intenta actualizar con los mismos datos.	El sistema actualiza, pero no modifica ningún dato.	Correcto.





Eliminación de preguntas		
Caso de prueba	Resultado esperado	Resultado final
El usuario intenta eliminar una pregunta existente que no está asociada a ningún cuestionario.	El sistema elimina la pregunta y muestra un mensaje al usuario.	Correcto.
El usuario intenta eliminar una pregunta asociada a algún cuestionario.	El sistema no permite eliminarla y muestra un mensaje de error explicando el motivo.	Se detecta que el sistema permite eliminarla, quedando en cuestionario sin dicha pregunta.

Creación de cuestionarios		
Caso de prueba	Resultado esperado	Resultado final
Se introduce un título y se añade al menos una pregunta.	El sistema crea el cuestionario y muestra un mensaje.	Correcto.
Se introduce un título y no se añade ninguna pregunta.	El sistema no permite crear el cuestionario y muestra un mensaje de error indicando el motivo.	Correcto.
No se introduce el título, pero si se añade al menos una pregunta.	El sistema no permite crear el cuestionario y muestra un mensaje de error indicando el motivo.	Correcto.
Se añaden al menos 2 preguntas y se elimina una.	El cuestionario debería quedar una pregunta menos.	Correcto.
Se introducen varias preguntas y el usuario intenta cambiarlas de orden.	El sistema debería dejar arrastrar las preguntas para cambiarlas de orden.	Correcto.
El usuario hace clic en alguna pregunta para ver sus respuestas.	El sistema muestra las respuestas de la pregunta seleccionada.	Correcto.

Modificación de cuestionarios		
Caso de prueba	Resultado esperado	Resultado final
El usuario selecciona un cuestionario para editarlo.	Al entrar a la pantalla, los campos deben estar completos con los datos del cuestionario.	Correcto.
El usuario intenta añadir o eliminar preguntas.	El sistema no debería dejar hacerlo.	Correcto.
El usuario edita el título y orden de las preguntas del cuestionario.	El sistema actualiza el cuestionario y muestra un mensaje.	Correcto.

Eliminación de cuestionarios		
Caso de prueba	Resultado esperado	Resultado final
El usuario intenta eliminar un cuestionario que no tenga resultados asociados.	El sistema permite eliminar el cuestionario y muestra un mensaje al usuario..	Correcto.





El usuario intenta eliminar un cuestionario que tiene resultados asociados.	El sistema no permite eliminar el cuestionario y muestra un mensaje de error indicando el motivo.	Se detecta que el sistema no mostraba el mensaje de error.

<b>Gestión de cursos</b>		
<b>Caso de prueba</b>	<b>Resultado esperado</b>	<b>Resultado final</b>
El usuario intenta añadir al curso cuestionarios que no han sido asociados todavía.	El sistema añade los cuestionarios y muestra un mensaje.	Correcto.
El usuario intenta añadir al curso cuestionarios que ya han sido asociados.	El sistema no debe dejar añadir los cuestionarios que ya hayan sido asociados.	Correcto.
El usuario intenta añadir al curso preguntas que no han sido asociadas todavía.	El sistema añade las preguntas al curso y muestra un mensaje.	Correcto.
El usuario intenta añadir al curso preguntas que ya han sido asociadas.	El sistema no debe dejar añadir las preguntas que ya hayan sido asociadas.	Correcto.
El usuario intenta añadir al curso alumnos que no han sido asociados todavía.	El sistema añade los alumnos y muestra un mensaje.	Correcto.
El usuario intenta añadir al curso alumnos que ya han sido asociados.	El sistema no debe dejar añadir los alumnos que ya hayan sido asociados al curso.	Correcto.
El usuario intenta añadir al curso usuarios de tipo profesor o al administrador.	El sistema no debe mostrar los usuarios que no son alumnos.	Correcto.

<b>Lanzamiento de cuestionarios en línea</b>		
<b>Caso de prueba</b>	<b>Resultado esperado</b>	<b>Resultado final</b>
El usuario selecciona un tipo de puntuación y marca que quiere que se vean las preguntas en los dispositivos del alumno.	El sistema crea el juego y pasa a la pantalla de juego.	Correcto.
El usuario selecciona un tipo de puntuación y marca que no quiere que se vean las preguntas en los dispositivos del alumno.	El sistema crea el juego y pasa a la pantalla de juego.	Correcto.
El usuario no selecciona un tipo de puntuación e intenta crear el juego.	El sistema no permite crear el juego.	Correcto.



<b>Jugar un cuestionario en línea</b>		
<b>Caso de prueba</b>	<b>Resultado esperado</b>	<b>Resultado final</b>
El profesor está a la espera de que entren los alumnos.	El sistema debe mostrar los alumnos que vayan entrando y no debe dejar empezar hasta que haya al menos un alumno en la sala.	Correcto.
El profesor le da a empezar la partida.	Automáticamente, la pantalla del profesor empieza un contador y se muestra la previsualización de la descripción de la pregunta.	Correcto.
El contador de la previsualización se acaba y la pregunta es de tipo multi opción o verdadero o falso.	El sistema muestra la pregunta y todas sus posibles respuestas, así como el tiempo restante y el número de respuestas que van llegando.	Correcto.
El contador de la previsualización se acaba y la pregunta es de tipo respuesta corta.	El sistema muestra la pregunta e indica que el alumno debe escribir su respuesta, así como el tiempo restante y el número de respuestas que van llegando.	Correcto.
El tiempo de la pregunta finaliza.	El sistema muestra la respuesta correcta a la pregunta y empieza otro contador.	Correcto.
El contador de la pantalla del resultado finaliza.	El sistema muestra el podio con los tres jugadores con mejor puntuación.	Correcto.
El jugador le da al botón de siguiente en la pantalla del podio y todavía quedan preguntas.	El sistema muestra la previsualización de la siguiente pregunta.	Correcto.
El jugador le da al botón de siguiente en la pantalla del podio y ya no quedan preguntas.	El sistema indica que el juego se ha terminado.	Correcto.

<b>Consulta de resultados de los cuestionarios</b>		
<b>Caso de prueba</b>	<b>Resultado esperado</b>	<b>Resultado final</b>
El usuario selecciona la opción de "Juegos" en la pantalla del curso.	El sistema muestra todos los cuestionarios jugados en ese curso.	Correcto.
El usuario clic en uno de los juegos.	El sistema debe mostrar el resultado del cuestionario para cada uno de los alumnos que hayan participado en él.	Correcto.



#### 8.4.2.4 Casos de uso del alumno

Jugar un cuestionario en línea		
Caso de prueba	Resultado esperado	Resultado final
El alumno entra a la aplicación y ya existía un cuestionario abierto para un curso en el que esté asociado.	El sistema debe mostrar aquellos cuestionarios que hayan sido abiertos.	Correcto.
El alumno se encuentra ya en la aplicación y un profesor abre un cuestionario de un curso al que pertenece.	El sistema muestra en tiempo real el cuestionario que se acaba de abrir.	Correcto.
El alumno no pertenece a ningún curso y se abren cuestionarios.	El sistema no debería mostrarle ningún cuestionario abierto.	Correcto.
El alumno selecciona un juego para jugar.	El sistema lo redirige a la pantalla del juego y espera a que el profesor inicie la partida. Además, al entrar al juego, el alumno debería ver su nombre en la pantalla del profesor.	Correcto.
El profesor empieza el cuestionario.	El sistema muestra un contador que ira decrementando.	Correcto.
El contador finaliza y la pregunta es de tipo multi opción o verdadero o falso y no tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta, así como un botón por cada una de las posibles respuestas.	Correcto.
El contador finaliza y la pregunta es de tipo multi opción o verdadero o falso y tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta, la imagen asociada, así como un botón por cada una de las posibles respuestas.	Correcto.
El contador finaliza y la pregunta es de tipo respuesta corta y no tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta y una entrada de texto y un botón de aceptar para introducir la respuesta.	Correcto.
El contador finaliza y la pregunta es de tipo respuesta corta y tiene imagen asociada.	El sistema debe mostrar el enunciado de la pregunta, la imagen asociada y una entrada de texto y un botón de aceptar para introducir la respuesta.	Correcto.
El usuario selecciona la respuesta de una pregunta de tipo multi opción o verdadero o falso.	El sistema queda a la espera de que acabe el tiempo de la pregunta, y cuando finalice muestra si el alumno acertó o no.	Correcto.
El usuario introduce una respuesta de una pregunta de tipo respuesta corta.	El sistema queda a la espera de que acabe el tiempo de la pregunta, y cuando finalice muestra si el alumno acertó o no.	Correcto.
El tiempo para mostrar el resultado de la pregunta finaliza.	El sistema muestra la puntuación que lleva hasta el momento el alumno.	Correcto.
El profesor le da a siguiente y quedan preguntas.	El sistema muestra otra vez un contador antes de mostrar la pregunta.	Correcto.



El profesor le da a siguiente y no quedan preguntas.	El sistema muestra que el juego ha finalizado y muestra un botón para salir del juego.	Correcto.
El profesor abandona el juego en mitad de una partida.	El sistema finaliza el juego automáticamente.	Correcto.
El alumno abandona el juego en mitad de una partida.	El sistema permitirá al alumno volver a entrar.	Correcto.
El alumno vuelve a entrar a la partida en mitad de una pregunta.	El sistema no permitirá contestar al alumno hasta la siguiente pregunta.	Correcto.

<b>Consultar resultado de los cuestionarios</b>		
<b>Caso de prueba</b>	<b>Resultado esperado</b>	<b>Resultado final</b>
El jugador entra en algún curso al que pertenezca y haya participado en algún juego.	El sistema mostrará todos los cuestionarios jugados, así como sus resultados.	Correcto.
El jugador entre en algún curso al que pertenezca y no haya participado aún en algún juego.	El sistema no muestra ningún resultado.	Correcto.
El jugador hace clic en algún cuestionario que haya jugado.	El sistema lo redirige a una página donde podrá revisar el cuestionario, mostrando las preguntas con las respuestas y las respuestas que haya dado el alumno.	Correcto.

<b>Modificar contraseña</b>		
<b>Caso de prueba</b>	<b>Resultado esperado</b>	<b>Resultado final</b>
El alumno introduce su antigua contraseña, la nueva y su confirmación y la contraseña antigua coincide.	El sistema actualiza la contraseña del usuario y muestra un mensaje.	Se detecta que faltaba el campo de confirmación de contraseña.
El alumno introduce la antigua contraseña y la nueva y su confirmación, pero la contraseña antigua no coincide.	El sistema no permite actualizar la contraseña y muestra un mensaje de error explicando el motivo.	Se detecta que faltaba el campo de confirmación de contraseña.
El alumno no introduce algún campo.	El sistema no permite actualizar la contraseña.	Correcto.
Las dos contraseñas nuevas no coinciden.	El sistema no permite actualizar la contraseña.	Correcto.

## 8.5 CSI 5: EJECUCIÓN DE LAS PRUEBAS DEL SISTEMA

### 8.5.1 Pruebas de Usabilidad

En la sección 7.5.3 Pruebas de Usabilidad se definieron los cuestionarios y las actividades para realizar las pruebas de usabilidad de la aplicación web. A continuación, se recogen los datos del cuestionario preliminar.

#### 8.5.1.1 Resultados del cuestionario de evaluación

##### Sujeto 1

Edad
21
¿Cuál consideras que es tu nivel de experiencia en el uso de aplicaciones web?
<ol style="list-style-type: none"><li>1. Principiante</li><li><b>2. Intermedio</b></li><li>3. Experto</li></ol>
¿Con qué frecuencia utilizas aplicaciones web en tu día a día?
<ol style="list-style-type: none"><li>1. Diariamente</li><li><b>2. Varias veces por semana</b></li><li>3. Ocasionalmente</li></ol>
¿Qué tipo de dispositivo utilizas para acceder a aplicaciones web?
<ol style="list-style-type: none"><li>1. Ordenador de escritorio</li><li><b>2. Ordenador portátil</b></li><li><b>3. Teléfono inteligente</b></li><li>4. Tableta</li></ol>
¿Has usado alguna vez una aplicación para realizar cuestionarios en línea (Ej. Kahoot)?
<ol style="list-style-type: none"><li><b>1. Sí, he utilizado Kahoot</b></li><li>2. Sí, pero no Kahoot, si no otra aplicación similar.</li><li>3. No, pero me lo he planteado</li><li>4. No</li><li>5. No, y además no sé a qué se refiere con cuestionarios en línea.</li></ol>
¿De los siguientes aspectos, cuál valoras más en una aplicación web?
<ol style="list-style-type: none"><li>1. Que sea muy rápida.</li><li><b>2. Que sea intuitiva y fácil de usar.</b></li><li><b>3. Que tenga un diseño bonito.</b></li><li>4. Que tenga muchas funcionalidades</li></ol>

Ilustración 82 Cuestionario de evaluación del sujeto 1



## Sujeto 2

Edad
50 años
¿Cuál consideras que es tu nivel de experiencia en el uso de aplicaciones web?
<ol style="list-style-type: none"><li>1. Principiante</li><li><b>2. Intermedio</b></li><li>3. Experto</li></ol>
¿Con qué frecuencia utilizas aplicaciones web en tu día a día?
<ol style="list-style-type: none"><li><b>1. Diariamente</b></li><li>2. Varias veces por semana</li><li>3. Ocasionalmente</li></ol>
¿Qué tipo de dispositivo utilizas para acceder a aplicaciones web?
<ol style="list-style-type: none"><li>1. Ordenador de escritorio</li><li><b>2. Ordenador portátil</b></li><li>3. Teléfono inteligente</li><li>4. Tableta</li></ol>
¿Has usado alguna vez una aplicación para realizar cuestionarios en línea (Ej. Kahoot)?
<ol style="list-style-type: none"><li>1. Sí, he utilizado Kahoot</li><li>2. Sí, pero no Kahoot, si no otra aplicación similar.</li><li>3. No, pero me lo he planteado</li><li><b>4. No</b></li><li>5. No, y además no sé a qué se refiere con cuestionarios en línea.</li></ol>
¿De los siguientes aspectos, cuál valoras más en una aplicación web?
<ol style="list-style-type: none"><li><b>1. Que sea muy rápida.</b></li><li><b>2. Que sea intuitiva y fácil de usar.</b></li><li>3. Que tenga un diseño bonito.</li><li>4. Que tenga muchas funcionalidades</li></ol>

*Ilustración 83 Cuestionario de evaluación del sujeto 2*

## Sujeto 3

Edad
26
¿Cuál consideras que es tu nivel de experiencia en el uso de aplicaciones web?
<ol style="list-style-type: none"><li>1. Principiante</li><li><b>2. Intermedio</b></li><li>3. Experto</li></ol>
¿Con qué frecuencia utilizas aplicaciones web en tu día a día?
<ol style="list-style-type: none"><li><b>1. Diariamente</b></li><li>2. Varias veces por semana</li><li>3. Ocasionalmente</li></ol>



¿Qué tipo de dispositivo utilizas para acceder a aplicaciones web?
<ol style="list-style-type: none"> <li>1. Ordenador de escritorio</li> <li>2. Ordenador portátil</li> <li><b>3. Teléfono inteligente</b></li> <li><b>4. Tableta</b></li> </ol>
¿Has usado alguna vez una aplicación para realizar cuestionarios en línea (Ej. Kahoot)?
<ol style="list-style-type: none"> <li><b>1. Sí, he utilizado Kahoot</b></li> <li>2. Sí, pero no Kahoot, si no otra aplicación similar.</li> <li>3. No, pero me lo he planteado</li> <li>4. No</li> <li>5. No, y además no sé a qué se refiere con cuestionarios en línea.</li> </ol>
¿De los siguientes aspectos, cuál valoras más en una aplicación web?
<ol style="list-style-type: none"> <li>1. Que sea muy rápida.</li> <li><b>2. Que sea intuitiva y fácil de usar.</b></li> <li>3. Que tenga un diseño bonito.</li> <li><b>4. Que tenga muchas funcionalidades</b></li> </ol>

Ilustración 84 Cuestionario de evaluación del sujeto 3

### 8.5.1.2 Resultados del cuestionario de preguntas cortas sobre la aplicación

#### Sujeto 1

En una escala del 1 al 5, ¿qué tan intuitiva te pareció la interfaz de la aplicación para realizar las actividades?
<ol style="list-style-type: none"> <li>1. Nada intuitiva</li> <li>2. Poco intuitiva</li> <li>3. Neutral</li> <li>4. Bastante intuitiva</li> <li><b>5. Muy intuitiva</b></li> </ol>
¿Encuentras que las instrucciones proporcionadas eran claras y comprensibles para completar las actividades?
<ol style="list-style-type: none"> <li><b>1. Si</b></li> <li>2. No</li> <li>3. No estoy seguro</li> </ol>
¿Cuál fue la actividad que encontraste más fácil de realizar en la aplicación? ¿Por qué?
Eliminar cualquier tipo de entidad porque solo había que darle a un botón
¿Cuál fue la actividad que encontraste más difícil de realizar en la aplicación? ¿Por qué?
Añadir preguntas al cuestionario porque el botón de la flecha era menos intuitivo que el resto
En tu opinión, ¿qué aspectos de la aplicación podrían mejorarse para facilitar la realización de las actividades?
Ninguno



¿Hubo algún mensaje de error o problema técnico que encontraste durante la realización de las actividades?
<ol style="list-style-type: none"><li>1. <b>Si</b></li><li>2. No</li></ol>
Si respondiste "Si" en la pregunta anterior, ¿puedes describir brevemente el problema técnico que encontraste?
Me faltaba por añadir el tiempo al crear una pregunta
En términos generales, ¿te sentiste satisfecho/a con la experiencia de usar la aplicación para realizar las actividades?
<ol style="list-style-type: none"><li>1. <b>Si, muy satisfecho/a</b></li><li>2. Satisfecho/a</li><li>3. Neutral</li><li>4. Insatisfecho/a</li><li>5. Muy insatisfecho/a</li></ol>
¿Tienes algún comentario adicional o sugerencia para mejorar la aplicación y hacerla más amigable para el usuario?
No

Tabla 9 Respuestas a las preguntas sobre la app del sujeto 1

## Sujeto 2

En una escala del 1 al 5, ¿qué tan intuitiva te pareció la interfaz de la aplicación para realizar las actividades?
<ol style="list-style-type: none"><li>1. Nada intuitiva</li><li>2. Poco intuitiva</li><li>3. Neutral</li><li>4. <b>Bastante intuitiva</b></li><li>5. Muy intuitiva</li></ol>
¿Encuentras que las instrucciones proporcionadas eran claras y comprensibles para completar las actividades?
<ol style="list-style-type: none"><li>1. <b>Si</b></li><li>2. No</li><li>3. No estoy seguro</li></ol>
¿Cuál fue la actividad que encontraste más fácil de realizar en la aplicación? ¿Por qué?
Eliminar
¿Cuál fue la actividad que encontraste más difícil de realizar en la aplicación? ¿Por qué?
Añadir preguntas al cuestionario porque el botón desplegable conlleva a error por parecer que es para ir hacia atrás
En tu opinión, ¿qué aspectos de la aplicación podrían mejorarse para facilitar la realización de las actividades



La pantalla de alerta no se ve bien. El botón de opciones al crear una pregunta no está en una posición adecuada
¿Hubo algún mensaje de error o problema técnico que encontraste durante la realización de las actividades?
<ol style="list-style-type: none"><li>1. Si</li><li>2. No</li></ol>
Si respondiste "Sí" en la pregunta anterior, ¿puedes describir brevemente el problema técnico que encontraste?
Sí, pero fue porque no poner una respuesta correcta Además, cuando cree un usuario le di a aceptar y luego quiso volver a crearlo y la aplicación se quedó cargando
En términos generales, ¿te sentiste satisfecho/a con la experiencia de usar la aplicación para realizar las actividades?
<ol style="list-style-type: none"><li>1. Si, muy satisfecho/a</li><li>2. <b>Satisfecho/a</b></li><li>3. Neutral</li><li>4. Insatisfecho/a</li><li>5. Muy insatisfecho/a</li></ol>
¿Tienes algún comentario adicional o sugerencia para mejorar la aplicación y hacerla más amigable para el usuario?
No

Tabla 10 Respuestas a las preguntas sobre la app del sujeto 2

### Sujeto 3

En una escala del 1 al 5, ¿qué tan intuitiva te pareció la interfaz de la aplicación para realizar las actividades?
<ol style="list-style-type: none"><li>1. Nada intuitiva</li><li>2. Poco intuitiva</li><li>3. Neutral</li><li>4. <b>Bastante intuitiva</b></li><li>5. Muy intuitiva</li></ol>
¿Encuentras que las instrucciones proporcionadas eran claras y comprensibles para completar las actividades?
<ol style="list-style-type: none"><li>1. Si</li><li>2. No</li><li>3. No estoy seguro</li></ol>
¿Cuál fue la actividad que encontraste más fácil de realizar en la aplicación? ¿Por qué?
Editar las entidades porque era muy intuitivo.
¿Cuál fue la actividad que encontraste más difícil de realizar en la aplicación? ¿Por qué?



En tu opinión, ¿qué aspectos de la aplicación podrían mejorarse para facilitar la realización de las actividades?
<ol style="list-style-type: none"><li>1. Las alertas las pondría de otra forma para que se vieses porque no te das cuenta cuando salen.</li><li>2. Ayudaría que apareciese una nota informativa en algunos casos, como poner en el apartado de preguntas que ahí es el único sitio en el que se crean.</li></ol>
¿Hubo algún mensaje de error o problema técnico que encontraste durante la realización de las actividades?
<ol style="list-style-type: none"><li>1. <b>Si</b></li><li>2. No</li></ol>
Si respondiste "Sí" en la pregunta anterior, ¿puedes describir brevemente el problema técnico que encontraste?
No gestiona errores la aplicación.
En términos generales, ¿te sentiste satisfecho/a con la experiencia de usar la aplicación para realizar las actividades?
<ol style="list-style-type: none"><li>1. Si, muy satisfecho/a</li><li>2. <b>Satisfecho/a</b></li><li>3. Neutral</li><li>4. Insatisfecho/a</li><li>5. Muy insatisfecho/a</li></ol>
¿Tienes algún comentario adicional o sugerencia para mejorar la aplicación y hacerla más amigable para el usuario?
No

Tabla 11 Respuestas a las preguntas sobre la app del sujeto 3

### 8.5.1.3 Análisis de los resultados

Una vez recogidas las respuestas de los cuestionarios y estudiadas las notas obtenidas durante el proceso de las pruebas, podemos sacar varias conclusiones.

En primera instancia, con el cuestionario de evaluación, se puede ver que los tres sujetos tienen un nivel intermedio en el uso de aplicaciones web y las utilizan a diario. Esto hace que la probabilidad de saber usar la aplicación aumente, aunque hubiera sido interesante tener sujetos con un nivel principiante para ver el comportamiento. A su vez, también utilizan variedad de dispositivos, lo que nos indica que la aplicación web debería ser redimensionable para verse bien en todos los dispositivos. Para finalizar con el cuestionario de evaluación, es interesante recalcar que los tres coinciden en que una de las cosas más importantes en una aplicación es que sea intuitiva y fácil de usar, por lo que la usabilidad tiene mucho peso en prácticamente todas las aplicaciones.

Por otro lado, recopilando las respuestas de las preguntas cortas sobre la aplicación una vez se han hechos el conjunto de actividades propuestas, en líneas generales las valoraciones son favorables y

la aplicación resulta fácil de usar. En cuanto a las observaciones aportadas por los sujetos y los datos recogidos de la observación de las actividades, se pueden sacar las siguientes conclusiones:

- Con el usuario administrador, cuando se crean o actualizan usuarios o cursos, después de aceptar la alerta y que todo haya salido bien, la aplicación se mantenía en la misma pantalla, lo que hace que desoriente al usuario pensando que no se ha creado. Esto se ha resuelto simplemente redirigiendo a otra pantalla.
- Los usuarios echaron en falta botones para volver a la página anterior. Siempre pueden utilizar el propio botón del navegador, pero añadirlos ayuda al usuario.
- En el formulario para actualizar la contraseña como alumno, no hay un campo extra para confirmar la nueva contraseña.
- A la hora de crear preguntas o cuestionarios, hubo cierta confusión con el texto “Incorporar recurso”. No queda claro qué se puede incorporar, cuando solo permite imágenes. Cambiar por el texto “Incorporar imagen”.
- Los sujetos echaron en falta un diálogo de confirmación al eliminar entidades, ya sean preguntas, cuestionarios como profesor, o usuarios o cursos como administrador e incluso al cerrar sesión. La solución es crear un diálogo que pregunta si de verdad se quiere llevar a cabo la acción.
- En la pantalla de creación de preguntas, el botón de opciones donde se configura el tipo de pregunta y el tiempo de la misma se encuentra en la esquina superior izquierda de la página, lo que, en ocasiones, puede llegar a despistar.
- Por otro lado, en la pantalla de creación de cuestionarios, para añadir preguntas hay un botón en el margen derecho con una flecha que se usa para abrir un desplegable. Después de realizar las pruebas los usuarios no lo ven con la suficiente claridad. La solución añadir un texto descriptivo al botón “Añadir preguntas” y situarlo en un lugar más accesible.
- En la pantalla principal del profesor, existe un conjunto de botones para seleccionar lo que se ve. El orden, de arriba abajo, de estos botones era: *Cursos*, *Cuestionarios* y *Preguntas*. La sugerencia fue cambiar el orden a: *Cursos*, *Preguntas* y *Cuestionarios*, pues para crear un cuestionario es necesario haber creado preguntas anteriormente.

## 8.5.2 Pruebas de Accesibilidad

### 8.5.2.1 Revisión Preliminar

#### **Paso 1. Grupo seleccionado de páginas de la aplicación**

Debido a que la aplicación no tiene una gran cantidad de páginas, se someterá la aplicación entera al proceso de revisión.

#### **Paso 2. Examinar las páginas usando un navegador gráfico**

En este paso se han llevado a cabo las siguientes operaciones:



- Se han desactivado las imágenes para probar cómo queda el aspecto de la aplicación sin ellas.
- Ver si la aplicación sigue siendo usable al aumentar el tamaño del texto con CTRL + Rueda del ratón.
- Probar la aplicación con diferentes resoluciones para verificar que no es necesario el scroll horizontal. Las resoluciones con las que se probó fueron 800x600, 1024x768, 1152x864, 1280x720, 1280x768, 1280x960, 1280x1024 y 1600x1200.
- Probar la aplicación sin las hojas de estilo CSS. La aplicación debe de seguir siendo legible y usable.
- Ver cómo se visualiza la aplicación utilizando una escala de grises. Para ello se hará uso de la extensión de Chrome **Grayscale the Web – Save Sites**.
- Usar el teclado para navegar por la aplicación.

### **Paso 3. Pruebas automáticas de evaluación de accesibilidad**

Para realizar dicha evaluación, se utilizará la extensión para el navegador de **WAVE**.

#### Paso 4. Resumen de resultados

Las pruebas correspondientes que utilizan un navegador gráfico han sido superadas. La aplicación es perfectamente a excepción de dos puntos:

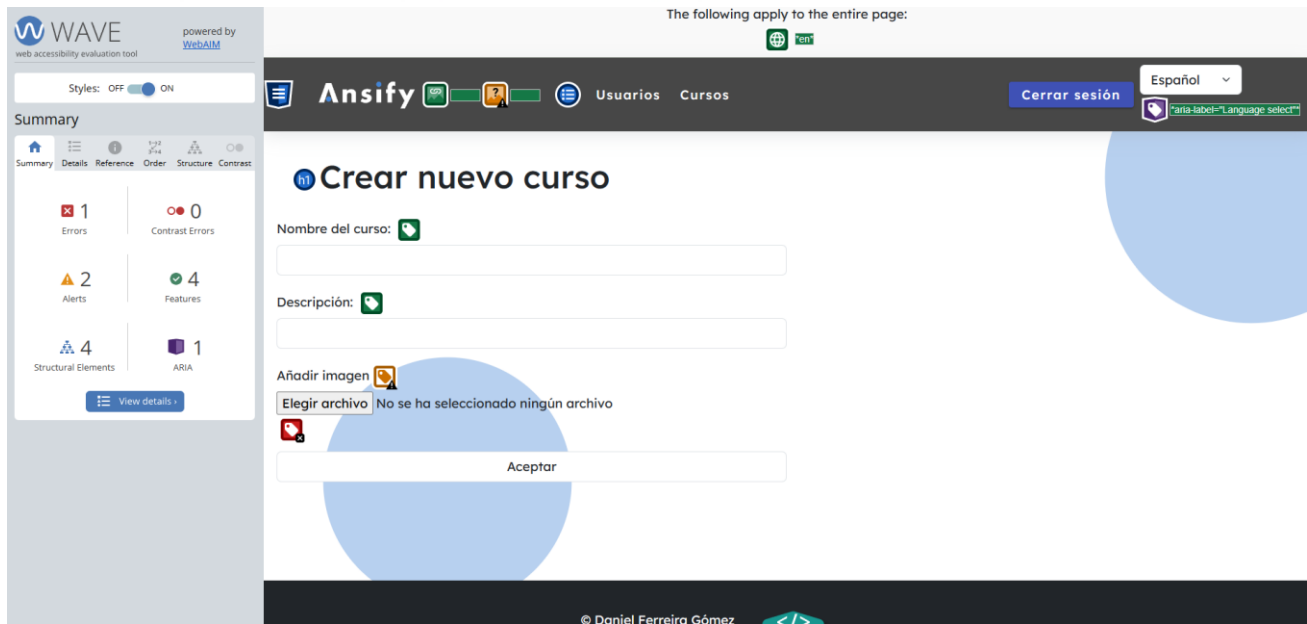
- Al esconder las imágenes, resultó en que los enlaces de la aplicación desaparecían debido a un estilo css que estaba siendo aplicado, por lo que ese aspecto tuvo que ser corregido.
- No es necesario scroll horizontal exceptuando en las partes de la aplicación donde se visualizan tablas. Se utilizan las tablas de la propia librería de componentes de Angular, **Angular Material**, por lo que el scroll horizontal es lo máximo que puede ofrecer.
- En la pantalla de creación de cuestionarios, para ordenar las preguntas es necesario coger y arrastrar, por lo que este aspecto no se puede llevar a cabo solamente con teclado. El resto de la aplicación es perfectamente usable únicamente con el teclado.

Respecto a las pruebas automáticas, se mostrarán a continuación capturas con aquellas pantallas que hayan presentado errores.



*Ilustración 85 Informe de WAVE del listado de usuarios del administrador*

Presenta dos errores, un *input* sin etiqueta y errores de contraste de color. Los mismos errores están presentes en el listado de cursos.



*Ilustración 86 Informe de WAVE para la pantalla de creación de cursos*

Al igual que en las pantallas anteriores, es necesario añadir una etiqueta al input para subir la imagen. En el resto de la aplicación se siguieron encontrando este mismo fallo de las etiquetas, por lo demás la aplicación parece estar correctamente adaptada.

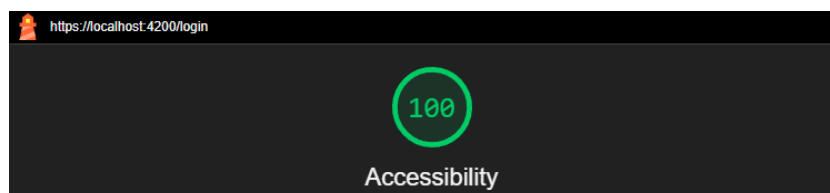
### 8.5.2.2 Evaluación de Conformidad

#### Paso 1. Determinar el alcance de la evaluación

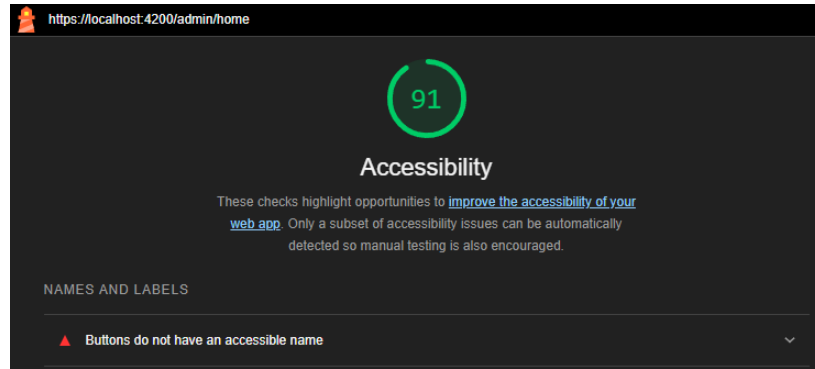
Como se ha explicado anteriormente, el nivel de conformidad según WCAG buscado es, como mínimo, AA, puesto que es el nivel mínimo exigido para las páginas de la administración pública, entendiéndolo como el nivel mínimo de calidad de la accesibilidad que debería tener la aplicación. Además, se dijo también que se han seleccionado todo el conjunto de páginas que componen la aplicación para realizar las pruebas.

#### Paso 2. Utilizar herramientas de evaluación automática de accesibilidad

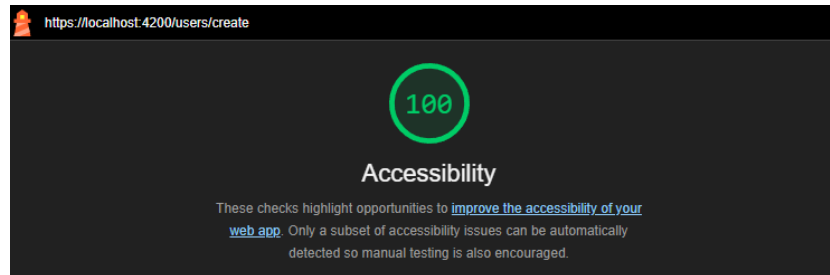
Se usará la herramienta **Lighthouse** del navegador Chrome para que ofrezca un informe de valoración de la totalidad de la aplicación. A continuación, se muestran las capturas de los informes de cada una de las páginas:



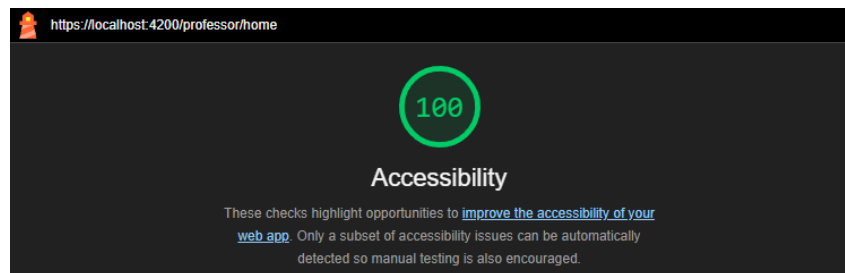
*Ilustración 87 Informe de lighthouse para la pantalla de inicio de sesión*



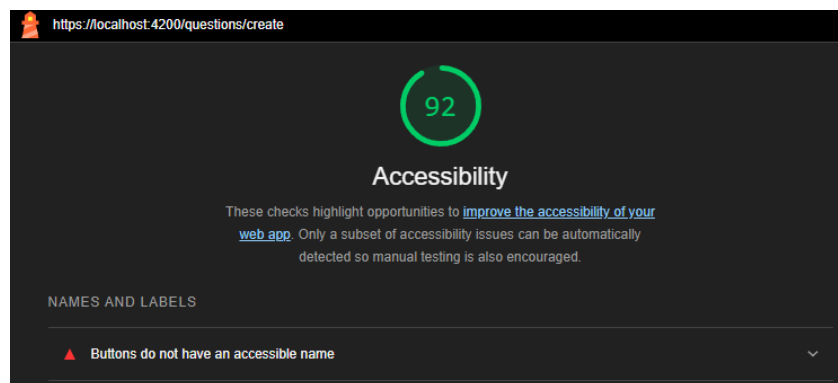
*Ilustración 88 Informe de lighthouse para la pantalla de listado de usuarios*



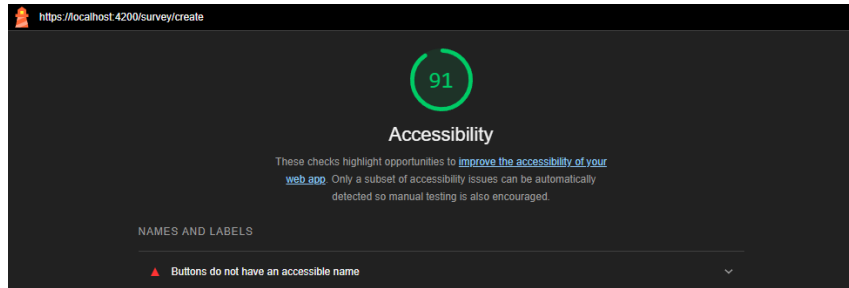
*Ilustración 89 Informe de lighthouse para la pantalla de creación de usuarios*



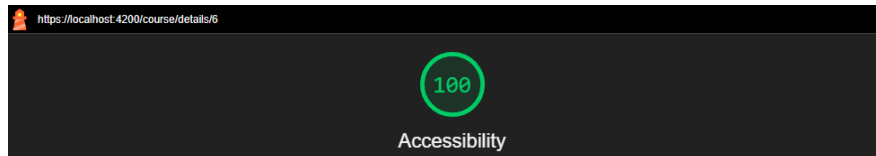
*Ilustración 90 Informe de lighthouse para la pantalla de inicio del profesor*



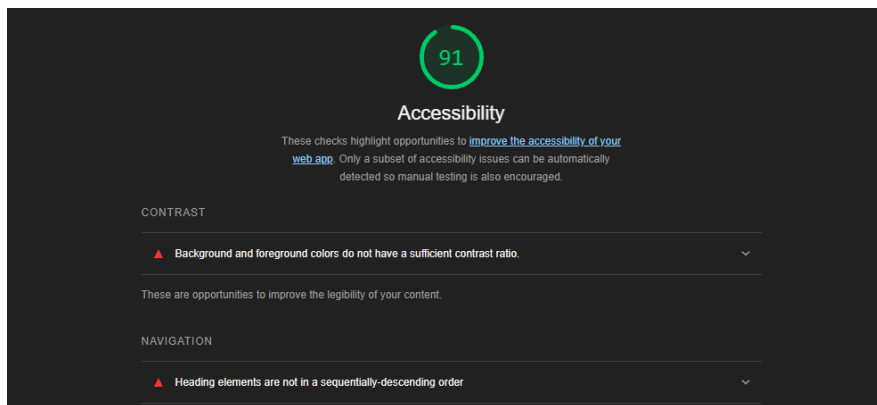
*Ilustración 91 Informe de lighthouse para la pantalla de creación de preguntas*



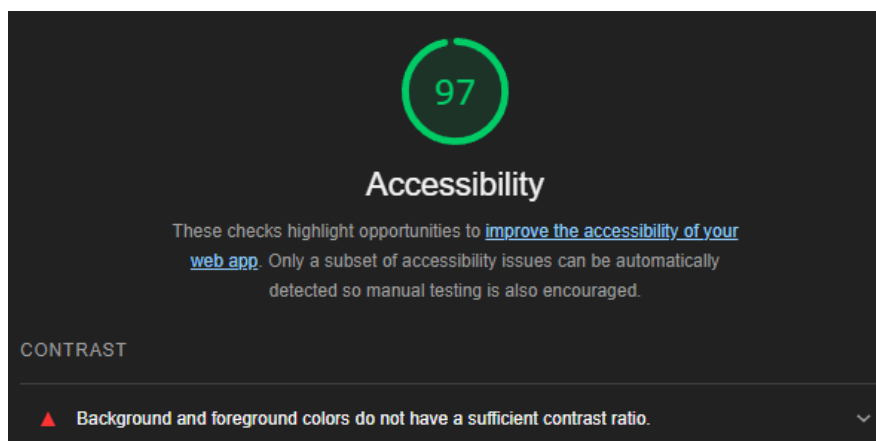
*Ilustración 92 Informe de lighthouse para la pantalla de creación de cuestionario*



*Ilustración 93 Informe de lighthouse para la pantalla de detalles de curso del profesor*

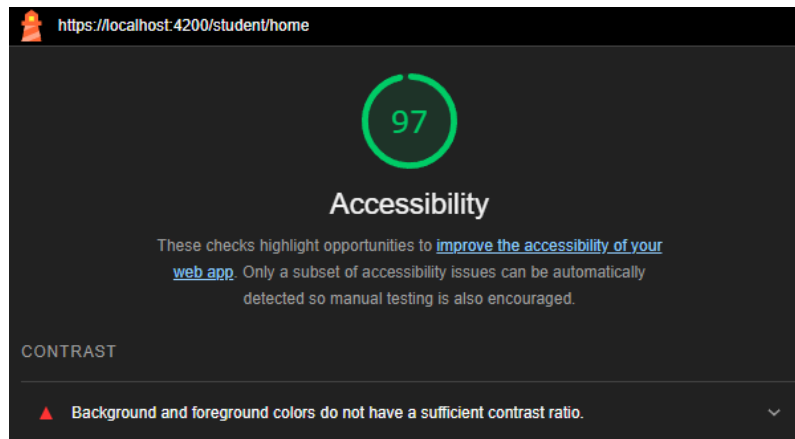


*Ilustración 94 Informe de lighthouse para la pantalla de juego del profesor*

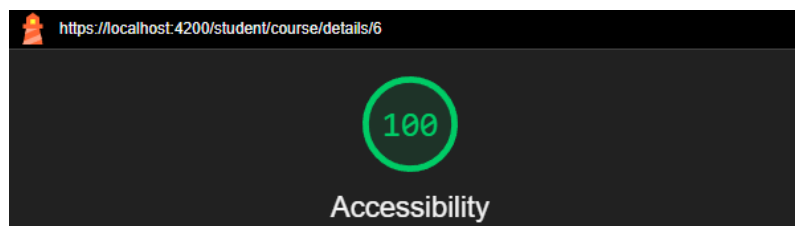


*Ilustración 95 Informe de lighthouse para la pantalla de juego del alumno*

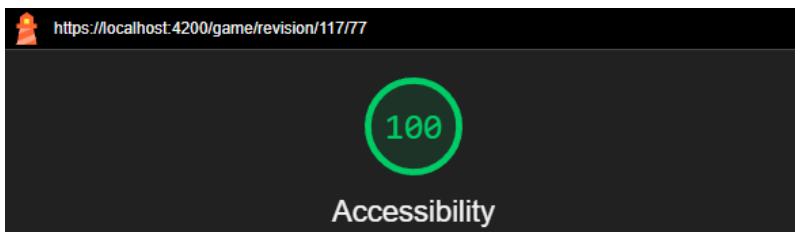




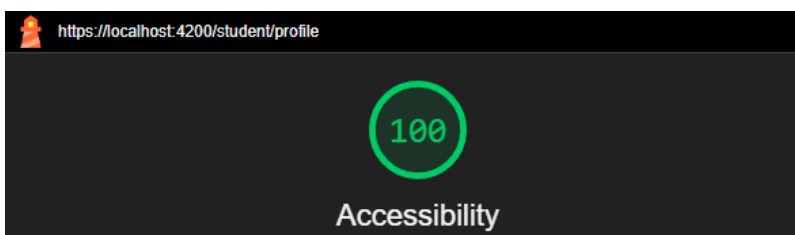
*Ilustración 96 Informe de lighthouse para la pantalla de inicio del alumno*



*Ilustración 97 Informe de lighthouse para la pantalla de detalles de curso del alumno*



*Ilustración 98 Informe de lighthouse para la pantalla de revisión de cuestionario*



*Ilustración 99 Informe de lighthouse para la pantalla de perfil*

### **Paso 3. Evaluar manualmente la muestra de las páginas**

Se utilizarán las checklist de la siguiente sección para evaluar el nivel de conformidad de cada una de las páginas de la aplicación utilizando los diferentes navegadores principales disponibles.

### **Paso 4. Elaborar el informe de conclusiones**

Una vez hechas las pruebas, se han tenido que cambiar un par de aspectos para corregir algunos de los informes de lighthouse. En total, han salido solo dos tipos de errores, de contraste y de secuencia



de los encabezados. Ambos han sido solucionados sin mucha complicación y lighthouse proporciona un informe de accesibilidad del 100% en todas y cada una de las páginas de la aplicación.

Finalmente, se han rellenado las checklist de los diferentes niveles de conformidad del WCAG 2.1, se puede decir que la aplicación es accesible como mínimo con un nivel AA según WCAG 2.1. Cabe destacar que se ha probado la accesibilidad únicamente con navegadores gráficos, por lo que, en una futura ampliación sería interesante realizar pruebas con navegadores textuales o de voz, pero esto se sale del alcance de este proyecto.

### 8.5.2.3 Checklist del WCAG 2.1

#### Puntos de verificación Nivel A:

	SI	NO	N/A
<b>1.1.1 Contenido no de texto:</b> Proporcionar alternativas de texto para contenido que no sea de texto	X		
<b>1.2.1 Solo audio y solo vídeo (pregrabado):</b> Proporcione una alternativa al contenido solo de vídeo y solo de audio			X
<b>1.2.2 Subtítulos (pregrabados):</b> Proporcione subtítulos para vídeos con audio			X
<b>1.2.3 Descripción de audio o alternativa multimedia (pregrabado):</b> El vídeo con audio tiene una segunda alternativa			X
<b>1.3.1 Información y relaciones:</b> Estructura lógica	X		
<b>1.3.2 Secuencia significativa:</b> Presente el contenido en un orden significativo	X		
<b>1.3.3 Características sensoriales:</b> Utilice más de un sentido para las instrucciones			X
<b>1.4.1 Uso del color:</b> No utilice una presentación que dependa únicamente del color	X		
<b>1.4.2 Control de audio:</b> No reproduzca audio automáticamente			X
<b>2.1.1 Teclado:</b> Accesible solo con teclado	X		
<b>2.1.2 Sin bloqueo de teclado:</b> No impida el uso de teclado a los usuarios	X		
<b>2.1.4 Accesos directos:</b> Proporcionar un mecanismo para configurar los métodos abreviados de teclado			X
<b>2.2.1 Temporización ajustable:</b> Los límites de tiempo tienen controles de usuario			X
<b>2.2.2 Pausa, Detener, Ocultar:</b> Proporcionar controles de usuario para mover contenido			X



<b>2.3.1 Tres destellos o menos:</b> Ningún contenido parpadea más de tres veces por segundo			<b>X</b>
<b>2.4.1 Bloques de derivación:</b> Proporcione un enlace "Skip to Content"			<b>X</b>
<b>2.4.2 Página titulada:</b> Utilice títulos de página útiles y claros	<b>X</b>		
<b>2.4.3 Orden de enfoque:</b> Orden lógico	<b>X</b>		
<b>2.4.4 Propósito del enlace (en contexto):</b> El propósito de cada enlace se desprende claramente de su contexto	<b>X</b>		
<b>2.5.1 Gestos de puntero:</b> todas las funciones que utilizan gestos multipunto o basados en rutas para hacer una operación se pueden hacer con un solo puntero sin gestos, a menos que sea esencial.			<b>X</b>
<b>2.5.2 Cancelación del puntero:</b> Para ayudar a evitar la activación involuntaria de los controles, evite la activación no esencial del evento descendente (por ejemplo, onmousedown) al hacer clic, tocar o presionar la pantalla durante mucho tiempo.	<b>X</b>		
<b>2.5.3 Etiqueta en Nombre:</b> Para los componentes de la interfaz de usuario con etiquetas que incluyen texto o imágenes de texto, el nombre contiene el texto que se presenta visualmente.	<b>X</b>		
<b>2.5.4 Motion Actuation:</b> La funcionalidad que se activa moviendo el dispositivo (como agitar o desplazar un dispositivo móvil) o por el movimiento del usuario (como saludar a una cámara) se puede desactivar y la funcionalidad equivalente se proporciona a través de controles estándar como botones			<b>X</b>
<b>3.1.1 Idioma de la página:</b> La página tiene un idioma asignado	<b>X</b>		
<b>3.2.1 Enfoque:</b> Los elementos no cambian cuando reciben enfoque	<b>X</b>		
<b>3.2.2 En la entrada:</b> Los elementos no cambian cuando reciben la entrada	<b>X</b>		
<b>3.3.1 Identificación de errores:</b> Identificar claramente los errores de entrada	<b>X</b>		
<b>3.3.2 Etiquetas o Instrucciones:</b> Etiquetar elementos y dar instrucciones	<b>X</b>		
<b>4.1.1 Análisis:</b> No hay errores de código importantes	<b>X</b>		
<b>4.1.2 Nombre, Rol, Valor:</b> Construir todos los elementos pensando en la accesibilidad	<b>X</b>		

### Checklist Nivel A



### Puntos de verificación Nivel AA:

	SI	NO	N/A
<b>1.2.4 Subtítulos (en vivo):</b> Los vídeos en directo tienen subtítulos			X
<b>1.2.5 Descripción de audio (pregrabada):</b> los usuarios tienen acceso a la descripción de audio para el contenido de vídeo			X
<b>1.3.4 Orientación:</b> El contenido no restringe su vista y operación a una sola orientación de visualización, como vertical u horizontal, a menos que una orientación de visualización específica sea esencial	X		
<b>1.3.5 Identificar propósito de entrada:</b> los campos de entrada que recopilan ciertos tipos de información de usuario tienen definido un atributo de autocompletar adecuado	X		
<b>1.4.3 Contraste (mínimo):</b> La relación de contraste entre texto y fondo es de al menos 4,5:1	X		
<b>1.4.4 Cambiar el tamaño del texto:</b> El texto se puede redimensionar al 200% sin pérdida de contenido o función	X		
<b>1.4.5 Imágenes de texto:</b> No utilice imágenes de texto	X		
<b>1.4.10 Reflujo:</b> No se produce ninguna pérdida de contenido o funcionalidad y se evita el desplazamiento horizontal cuando el contenido se presenta con un ancho de 320 píxeles	X		
<b>1.4.11 Contraste no de texto:</b> Existe una relación de contraste de al menos 3:1 para diferenciar objetos gráficos (como iconos y componentes de gráficos o gráficos) y componentes de interfaz personalizados por el autor (como botones, controles de formulario y enfoque indicadores/esquemas)	X		
<b>1.4.12 Espaciado de texto:</b> No se produce pérdida de contenido o funcionalidad cuando el usuario adapta la altura de la línea de texto/espaciado a 1,5 veces el tamaño de la fuente, el espaciado de párrafo a 2 veces el tamaño de la fuente, el espaciado de palabras a .16 veces el tamaño de la fuente y el espaciado de letras a .12 veces el tamaño de la fuente	X		
<b>1.4.13 Contenido en Hover o Focus:</b> Cuando se presenta contenido adicional en el hover del cursor o en el focus del teclado: el contenido que aparece se puede descartar (generalmente a través de la tecla Esc) sin mover el puntero o el foco del teclado, a menos que el contenido presente un error de entrada o no oculte ni interfiera con el contenido de otra	X		



página; el puntero se puede mover al nuevo contenido sin que el contenido desaparezca; el nuevo contenido debe permanecer visible hasta que el puntero o el foco del teclado se aleje del control desencadenador, se descarte el nuevo contenido o el nuevo contenido ya no sea relevante			
<b>2.4.5 Múltiples maneras:</b> Ofrecer varias maneras de encontrar páginas			<b>X</b>
<b>2.4.6 Encabezados y etiquetas:</b> Utilice encabezados y etiquetas transparentes	<b>X</b>		
<b>2.4.7 Enfoque visible:</b> Asegúrese de que el foco del teclado sea visible y claro	<b>X</b>		
<b>3.1.2 Idioma de las piezas:</b> Indique a los usuarios cuándo cambia el idioma de una página	<b>X</b>		
<b>3.2.3 Navegación consistente:</b> Utilice los menús de forma coherente	<b>X</b>		
<b>3.2.4 Identificación consistente:</b> Utilice iconos y botones de forma consistente	<b>X</b>		
<b>3.3.3 Sugerencia de error:</b> Sugerir correcciones cuando los usuarios cometen errores	<b>X</b>		
<b>3.3.4 Prevención de errores (Legal, Financiero, Datos):</b> Reducir el riesgo de errores de entrada de datos confidenciales			<b>X</b>
<b>4.1.3 Mensajes de estado:</b> los mensajes de estado se programar considerando roles o propiedades del usuario y presentarse al mismo mediante tecnologías de asistencia sin recibir el foco.	<b>X</b>		

### Checklist Nivel AA

#### Puntos de verificación Nivel AAA:

	SI	NO	N/A
<b>1.2.6 Lenguaje de señas (pregrabado):</b> Proporcione traducciones de lengua de signos para vídeos			<b>X</b>
<b>1.2.7 Descripción de audio ampliada (pregrabada):</b> Proporcione una descripción de audio ampliada para los vídeos			<b>X</b>
<b>1.2.8 Alternativa de medios (pregrabado):</b> Proporcione una alternativa de texto a los videos			<b>X</b>
<b>1.2.9 Solo audio (en vivo):</b> Proporcione alternativas para el audio en directo			<b>X</b>



<b>1.3.6 Identificar propósito:</b> el propósito de los componentes de interfaz de usuario, iconos y regiones se puede determinar programáticamente		<b>X</b>	
<b>1.4.6 Contraste (mejorado):</b> La relación de contraste entre texto y fondo es de al menos 7:1	<b>X</b>		
<b>1.4.7 Audio de fondo bajo o sin fondo:</b> El audio es claro para que los oyentes escuchen			<b>X</b>
<b>1.4.8 Presentación visual:</b> Ofrezca a los usuarios una gama de opciones de presentación			<b>X</b>
<b>1.4.9 Imágenes de texto (sin excepción):</b> No utilice imágenes de texto	<b>X</b>		
<b>2.1.3 Teclado (sin excepción):</b> Solo accesible con teclado, sin excepción	<b>X</b>		
<b>2.2.3 Sin tiempo:</b> Sin límites de tiempo		<b>X</b>	
<b>2.2.4 Interrupciones:</b> No interrumpa a los usuarios	<b>X</b>		
<b>2.2.5 Re-autenticación:</b> Guarde los datos del usuario al volver a autenticar	<b>X</b>		
<b>2.2.6 Tiempos de espera:</b> Se advierte a los usuarios de la duración de cualquier inactividad del usuario que pueda causar la pérdida de datos, a menos que los datos se conserven durante más de 20 horas cuando el usuario no realice ninguna acción		<b>X</b>	
<b>2.3.2 Tres destellos:</b> Ningún contenido parpadea más de tres veces por segundo	<b>X</b>		
<b>2.3.3 Animación de interacciones:</b> La animación de movimiento activada por interacción se puede desactivar, a menos que la animación sea esencial para la funcionalidad o la información que se transmite	<b>X</b>		
<b>2.4.8 Ubicación:</b> Haga saber a los usuarios dónde se encuentran			<b>X</b>
<b>2.4.9 Propósito del enlace (solo enlace):</b> El propósito de cada enlace se desprende claramente de su texto		<b>X</b>	
<b>2.4.10 Encabezados de sección:</b> Dividir el contenido con encabezados	<b>X</b>		
<b>2.5.5 Tamaño de destino:</b> Los objetivos en los que se puede hacer clic tienen al menos 44 por 44 píxeles de tamaño a menos que se proporcione un objetivo alternativo de ese tamaño, el objetivo esté en línea (por ejemplo, un vínculo dentro de una oración), el objetivo no se modifique por el autor (por ejemplo,		<b>X</b>	

una casilla de verificación predeterminada) o el tamaño del objetivo es esencial que sea pequeño para la funcionalidad			
<b>2.5.6 Mecanismos de entrada simultánea:</b> El contenido web no restringe el uso de las modalidades de entrada disponibles en una plataforma, excepto cuando la restricción sea esencial, necesaria para garantizar la seguridad del contenido o necesaria para respetar la configuración del usuario		X	
<b>3.1.3 Palabras inusuales:</b> Explicar cualquier palabra extraña			X
<b>3.1.4 Abreviaturas:</b> Explicar las abreviaturas			X
<b>3.1.5 Nivel de lectura:</b> Los usuarios con nueve años de escuela pueden leer su contenido	X		
<b>3.1.6 Pronunciación:</b> Explique las palabras que sean difíciles de pronunciar			X
<b>3.2.5 Cambio a petición:</b> No cambie elementos en su sitio web hasta que los usuarios pregunten	X		
<b>3.3.5 Ayuda:</b> Proporcione ayuda e instrucciones detalladas		X	
<b>3.3.6 Prevención de errores (Todos):</b> Reducir el riesgo de todos los errores de entrada	X		

#### Checklist Nivel AAA

##### 8.5.2.4 Accesibilidad con Dispositivos Móviles

Por último, es necesario comentar la accesibilidad para los dispositivos móviles. Anteriormente se ha comentado que la aplicación había sido diseñada con estos dispositivos en mente, por lo que cada una de las páginas que componen a la aplicación son totalmente redimensionables a las pantallas de los móviles, y, por tanto, toda la información es mostrada correctamente. La única excepción es la anteriormente citada, las tablas de los resultados de la aplicación, que, en dispositivos con pantalla reducida, es necesario hacer scroll horizontal únicamente en la tabla.



## 8.6 CSI 6: ELABORACIÓN DE LOS MANUALES DE USUARIO

---

En este capítulo se verán los diferentes manuales del sistema que servirán para entenderlo desde diferentes perspectivas, ya sea para instalar y desplegar, entender el uso de la aplicación web o para llevar a cabo ampliaciones sobre éste.

### 8.6.1 Manual de Instalación

Para este sistema no es necesario la instalación de ninguna aplicación para proceder a usarla, debido a que se trata de una aplicación web desplegada en un servidor. Si se quisiera instalar y ejecutar en local, el siguiente manual servirá para entender cómo hacerlo. Se puede acceder a la aplicación utilizando el siguiente enlace: [Ansify](#).

### 8.6.2 Manual de Ejecución y Despliegue

En este manual se verá cómo poder ejecutar paso a paso, tanto la aplicación web, como el servidor web en un entorno local una vez se tiene el código fuente del proyecto.

#### 8.6.2.1 Instalar Node.js, PostgreSQL y Angular

Lo primero de todo es tener instalado Nodejs y Angular para poder ejecutar el servidor web y la aplicación web y PostgreSQL para poder ejecutar la base de datos que utiliza el sistema.

Para instalar Nodejs podemos ir a [Descarga | Node.js \(nodejs.org\)](#) para descargar versión recomendada o la última. El proyecto se ha desarrollado con dos versiones de node diferentes, una versión para el ordenador personal del autor y otra versión en el servidor. Estas versiones son la **v18.14.0** y la **v18.16.1**, por lo que en caso de que se quiera descargar la misma se puede hacer en [Versiones Anteriores | Node.js \(nodejs.org\)](#).

Par instalar la base de datos PostgreSQL, se puede hacer desde la página oficial [PostgreSQL: Downloads](#). El proyecto utiliza versión 15, la última a fecha de realización de este trabajo. Descargar el ejecutable correspondiente al sistema operativo donde se quiera instalar. En un momento concreto de la instalación, pedirá la contraseña para el usuario **postgres**, es muy importante recordarla, puesto que la necesitaremos más adelante. Lo siguiente sería crear la base de datos, aunque podemos utilizar la base de datos postgres que viene por defecto.

Finalmente, es necesario instalar Angular para poder ejecutar la aplicación web. En la documentación oficial [Angular - Deploying an application](#) nos dice paso a paso cómo hacerlo. Básicamente necesitamos haber instalado npm (que viene por defecto al instalar Nodejs) y ejecutar el siguiente comando: **npm install -g @angular/cli**.



Con esto ya tendríamos todo lo necesario para ejecutar ambos sistemas.

### 8.6.2.2 Ejecutar el servidor web

Una vez tengamos todo lo necesario instalado, podemos empezar a ejecutar cada uno de los subsistemas. Antes de nada, es necesario crear un fichero `.env` en la raíz del subsistema `../restapi`, con los secretos que utiliza la aplicación. A continuación, se muestra un ejemplo de este fichero:

```
TEST_DATABASE_URL = "postgres://postgres:password@localhost:5432/test_data_base_name"  
PRODUCTION_DATABASE_URL = "postgres://postgres:password@localhost:5432/data_base_name"  
SECRET = "some_random_secret"  
EMAIL="ferreiragomezdaniel@gmail.com"  
REFRESH_TOKEN="refresh_token_google"  
CLIENT_ID="client_id_google"  
CLIENT_SECRET="client_secret_google"  
REDIRECT_URI="https://developers.google.com/oauthplayground"
```

#### *Ilustración 100 Ejemplo de .env*

En el ejemplo anterior vemos que hay dos bases de datos, una para tests y otra para los datos en producción, aunque es posible que solo esté la de `PRODUCTION_DATABASE_URL` e ir cambiando la url según la base de datos que queramos utilizar. Para poner la base de datos, será necesario cambiar la contraseña y el nombre de la base de datos, además del usuario y el puerto en el caso de que se hayan cambiado los que vienen por defecto. Por otro lado, tenemos `SECRET`, que es la cadena de texto que se utiliza para hacer un hash de las contraseñas pudiendo poner cualquiera. `REFRESH_TOKEN`, `CLIENT_ID`, `CLIENT_SECRET` y `REDIRECT_URI` se utilizan para enviar un correo con las credenciales cuando el administrador crea un usuario. No son esenciales, pues si no se introducen, simplemente no se envía el correo y explicar cómo se obtienen se escapa de los límites de este manual.

Para lanzar el servidor web, abrimos una consola de comando y nos ponemos en la raíz del subsistema, y ejecutamos los siguientes comandos:

1. `npm install`: esto hará que se instalen todas las dependencias necesarias para ejecutar el proyecto.
2. `npm run migrateDb`: este comando hace que el ORM Prisma actualice y le dé forma a la base de datos según el esquema de `../restapi/prisma/schema.prisma`.
3. `npm start`: finalmente, con este comando se inicia el servidor.

### 8.6.2.3 Ejecutar la aplicación web

En el caso de la aplicación web, nos situamos también en su directorio raíz, `../webapp`, y en el caso de que queramos ejecutarlo en local, introducimos los siguientes comandos:

1. `npm install`: instala todas las dependencias necesarias para ejecutar el proyecto.



2. `ng serve --ssl --ssl-cert './restapi/certificate/certificate.crt.pem' --ssl-key './restapi/certificate/privateKey.key.pem'`: esto hará que se inicie la aplicación en local. Como se puede ver, utiliza un certificado y una clave ubicadas en restapi, dado que utilizan el mismo para poder conectarse entre ellas. Se ha puesto en el proyecto para facilitar el desarrollo.

Con esto podríamos acceder a <https://localhost:4200> y entrar a la página.

En el caso de que se quiera desplegar en el servidor para poder acceder desde otros dispositivos, primero habría que configurar un servidor web. Actualmente, la aplicación se encuentra desplegada en un servidor web Apache, por lo que en este manual se enseñará a hacer un uno idéntico. Para descargar Apache, lo podemos hacer desde [Download - The Apache HTTP Server Project](#). En mi caso he descargado XAMPP, que también descarga Apache, por lo que es indiferente. Una vez instalado Apache, necesitamos modificar dos ficheros, **httpd.conf** y **httpd-ssl.conf**. En el primero simplemente tenemos que añadir la línea **Listen 0.0.0.0:80** y descomentar o añadir si no estuviera la línea **LoadModule ssl\_module modules/mod\_ssl.so**. En el segundo fichero, correspondiente a la configuración del servidor para usar https, solo tenemos que modificar el virtual host que viene por defecto. Este es:

```
<VirtualHost _default_:443>

# General setup for the virtual host
DocumentRoot "C:/xampp/htdocs"
ServerName 156.35.163.164:443
ServerAdmin admin@example.com
ErrorLog "C:/xampp/apache/logs/error.log"
TransferLog "C:/xampp/apache/logs/access.log"

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on
```

### *Ilustración 101 httpd-ssl.conf*

Aquí solo es necesario cambiar, en el caso de que no venga así por defecto, **ServerName** por la ip o nombre de dominio de nuestro servidor y dejar la línea **SSLEngine** a **on**. Con esto ya tendríamos configurado nuestro servidor web, solo nos resta compilar la aplicación web y desplegarla.

Volvemos a la raíz de la aplicación web y ejecutamos el comando `ng build`. Esto hará que se compile nuestra aplicación y la deja en el directorio de salida, que en este caso es **../webapp/dist/webapp**. Copiamos todos los ficheros que contenga la carpeta y lo pegamos en la carpeta **htdocs** de nuestro servidor Apache y con esto ya tendríamos desplegada nuestra aplicación web.

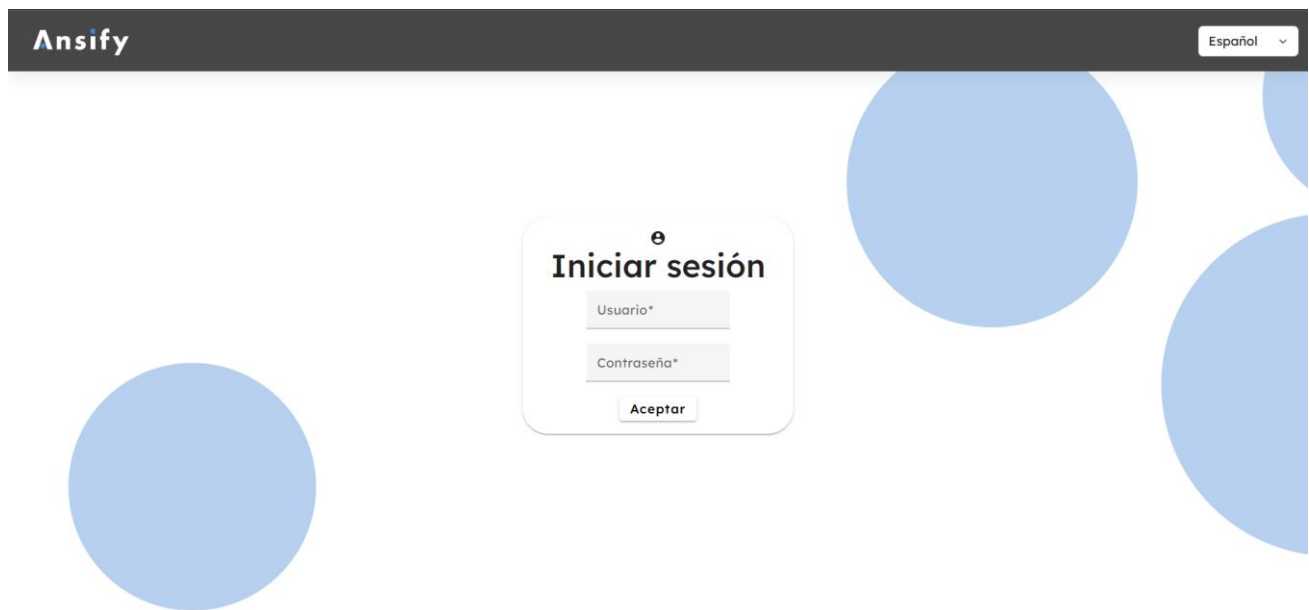
## 8.6.3 Manual de Usuario

El objetivo de este manual es que cualquier persona que lo lea, sea capaz de utilizar la aplicación sin muchas complicaciones. A continuación, se irán exponiendo cada uno de los casos de uso y su explicación para llevarlos a cabo.

### 8.6.3.1 Iniciar sesión

Mediante la siguiente pantalla el usuario podrá iniciar sesión con un usuario y su contraseña. Si las credenciales son correctas, el sistema redirigirá al usuario a la pantalla de inicio correspondiente con el rol del usuario. Si las credenciales no fueran válidas, saldría un mensaje de error indicándolo.

Además, se puede ver en la barra superior, que el usuario podrá cambiar el idioma de la aplicación en cualquier momento.



*Ilustración 102 Pantalla de inicio de sesión del manual de usuario*

### 8.6.3.2 Gestionar usuarios como administrador

El administrador empieza en la lista de usuarios del sistema. En ella podrá ver todos los usuarios del sistema, además de importar un fichero csv con usuarios donde indica “Importar usuarios” y eliminar los usuarios pulsando el botón de eliminar de cada uno de ellos. Para crear un usuario nuevo, tendrá que pulsar el botón de “Nuevo usuario”, y la aplicación lo redirigirá hacia el formulario de creación de usuario. Para editar un usuario, bastará con hacer clic encima del nombre de cualquiera de ellos. El formulario para editar un usuario es igual que el de crear, con la única diferencia que el primero tiene un campo para editar la contraseña.



Ansify Usuarios Cursos Cerrar sesión Español

## Lista de usuarios

Nuevo usuario Importar usuarios:

<b>PROFESOR</b> professor	Eliminar
<b>UO277371</b> student	Eliminar
<b>UO277603</b> student	Eliminar

*Ilustración 103 Lista de usuarios del administrador*

Para crear un nuevo usuario, simplemente es necesario rellenar el nombre que queramos que tenga e indicar el rol que deseamos que tenga, ya sea estudiante o profesor.

Ansify Usuarios Cursos Cerrar sesión Español

## Crear nuevo usuario

Nombre del usuario:

Rol:

*Ilustración 104 Formulario de creación de nuevo usuario del administrador*

Ansify Usuarios Cursos Cerrar sesión Español

**Usuario: UO277603**  
Rol: student

Nuevo nombre de usuario:

Nueva contraseña:

Nuevo rol:

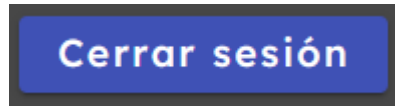
*Ilustración 105 Formulario de actualización de un usuario del administrador*

### 8.6.3.3 Gestionar cursos como administrador

El funcionamiento para los cursos es análogo al de los usuarios, por lo que se omite la explicación.

### 8.6.3.4 Cerrar sesión

Para cerrar la sesión, simplemente hay que pulsar el siguiente botón de la barra superior de navegación:

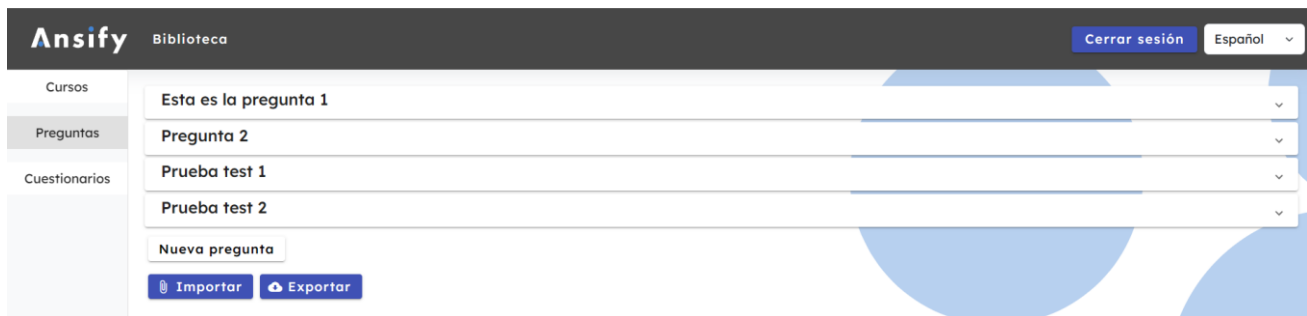


*Ilustración 106 Botón para cerrar sesión*

### 8.6.3.5 Gestionar preguntas como profesor

Una de las cosas que puede hacer un profesor es crear preguntas, para ello simplemente tiene que ir a la sección de Preguntas con los botones en el margen izquierdo de la pantalla. Esto hará que aparezcan las preguntas que haya creado el usuario. Para ir al formulario de crear pregunta habrá que pulsar el botón de Nueva pregunta.

En esta pantalla además se podrá importar preguntas en un fichero csv y exportar las preguntas en un fichero con el mismo formato. Si el usuario clic encima de una pregunta, se despliega un menú indicando las respuestas a la pregunta.



*Ilustración 107 Pantalla de librería del profesor*

Una vez en la pantalla de crear pregunta, vemos un formulario donde podemos insertar un título, una imagen y cada una de las cuatro respuestas. Además, arriba a la izquierda hay un botón de Opciones donde podremos seleccionar el tipo de pregunta, así como el tiempo que queremos que dure la misma. Dependiendo del tipo de pregunta, las respuestas cambiarán.

**Ilustración 108** Formulario de creación de preguntas

Para editarlas y eliminarlas es necesario clicar sobre ellas, haciendo que aparezcan el botón de editar y el de eliminar. El formulario para editar una pregunta es idéntico al de crearla.

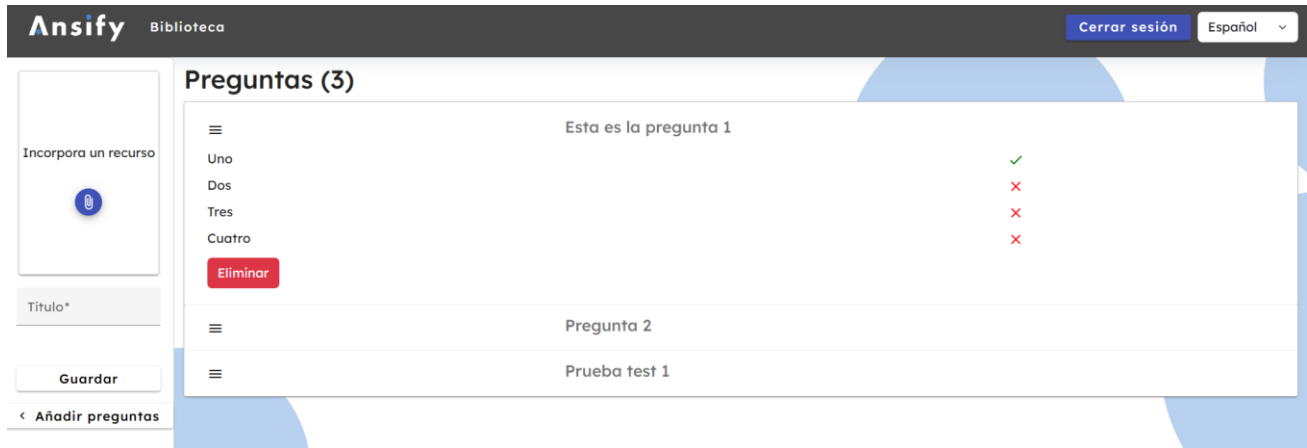
**Ilustración 109** Pregunta desplegada

### 8.6.3.6 Gestionar cuestionarios como profesor

El profesor también podrá crear cuestionarios con las preguntas que haya creado con anterioridad. Para ello tendrá que ir a la pantalla de Cuestionarios y darle a Nuevo cuestionario. Además, desde aquí también se podrá editar y eliminar los cuestionarios.

**Ilustración 110** Lista de cuestionarios de la librería del profesor

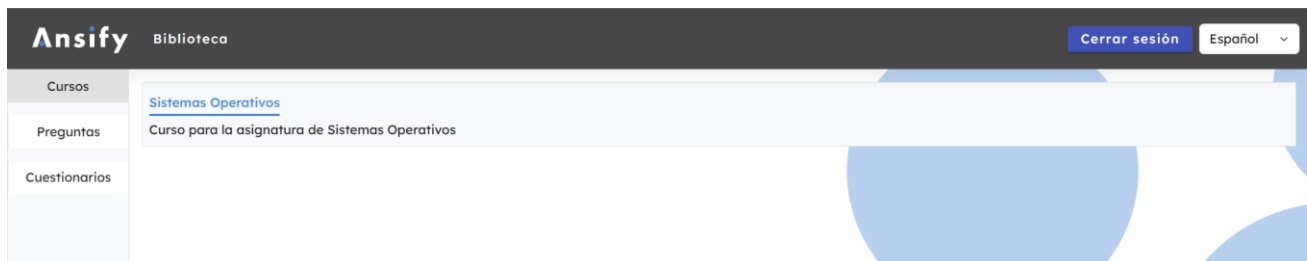
Para crear un cuestionario simplemente es necesario añadir un título y al menos una pregunta. Para añadir preguntas hay que pulsar el botón Añadir preguntas, lo que abrirá un desplegable con nuestras preguntas. Una vez que se añada alguna pregunta, podemos ordenarlas cogiéndolas y arrastrándolas a la posición deseada. Pinchando sobre ellas también podremos ver sus respuestas y eliminarlas del cuestionario.



*Ilustración 111 Formulario de creación de cuestionario del profesor*

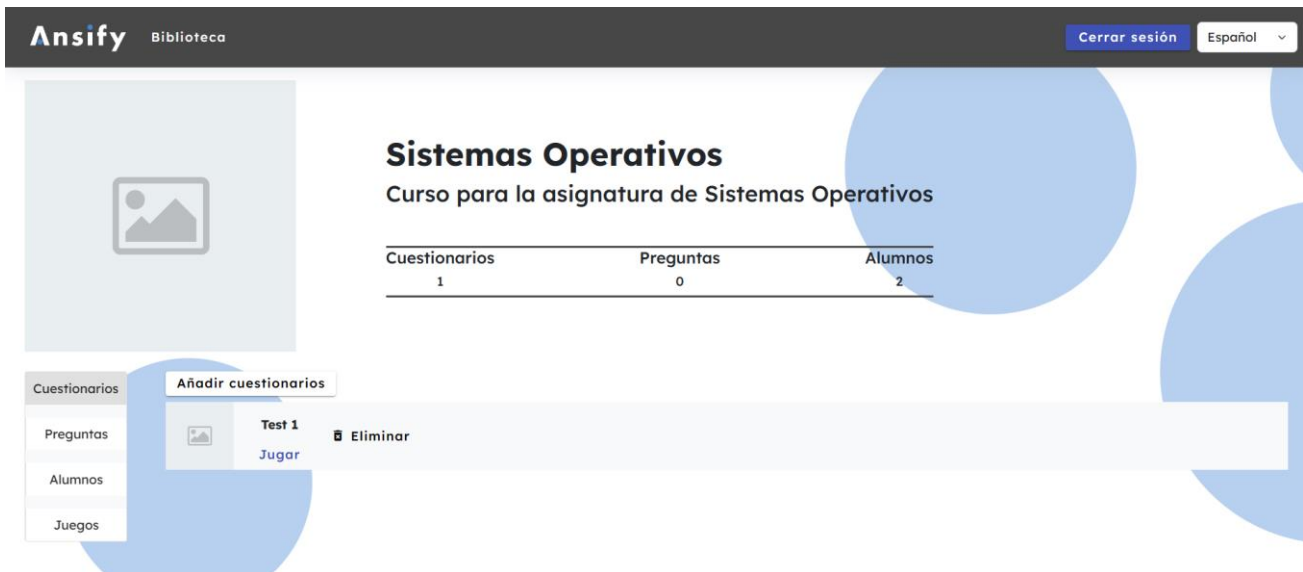
### 8.6.3.7 Gestión de cursos como profesor

Desde esta pantalla de aquí, el profesor podrá clicar encima de un curso y entrar a la vista en detalle del curso, desde la cual podemos asociar cuestionarios, preguntas y alumnos al curso



*Ilustración 112 Lista de cursos de la librería del profesor*

Como se puede ver en la siguiente pantalla, hay otro menú lateral izquierdo para cambiar la vista entre los cuestionarios, preguntas y alumnos. En cada una de ellas se muestra la lista que pertenecen al curso y un botón para añadir más. También se podrán eliminar del curso dándole al botón de eliminar.





**Ansify** Biblioteca Cerrar sesión Español

## Sistemas Operativos

Curso para la asignatura de Sistemas Operativos

Cuestionarios	Preguntas	Alumnos
1	0	2

Cuestionarios **Añadir cuestionarios**

Preguntas  **Test 1**  Eliminar

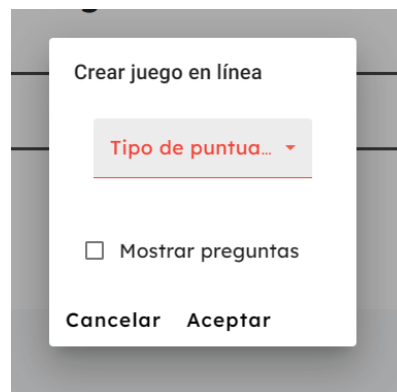
Alumnos **Jugar**

Juegos

*Ilustración 113 Pantalla de detalles del curso*

### 8.6.3.8 Lanzar un cuestionario y jugarlo como profesor

En la pantalla anterior se puede ver que sobre el cuestionario “Test 1” hay un botón que pone “Jugar”. Si pulsamos dicho botón, saldrá un diálogo que nos pedirá el tipo de puntuación del juego, ya sea normal, doble o sin puntuación, además de preguntarnos si queremos que se vean las preguntas en el dispositivo del alumno.



*Ilustración 114 Diálogo de creación de juego del profesor*

Una vez que se crea se muestra la lista de alumnos que van entrando al juego. Una vez que entra al menos un alumno, se puede empezar a jugar el cuestionario. Al darle a empezar, el juego irá ejecutándose solo, por lo que el usuario simplemente tendrá que darle clic a Siguiente cuando sea necesario.

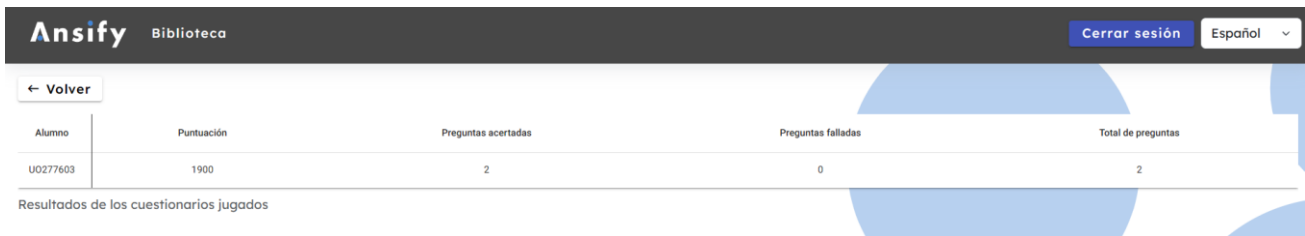




*Ilustración 115 Pantalla de lista de espera de alumnos*

### 8.6.3.9 Consultar resultados de un juego como profesor

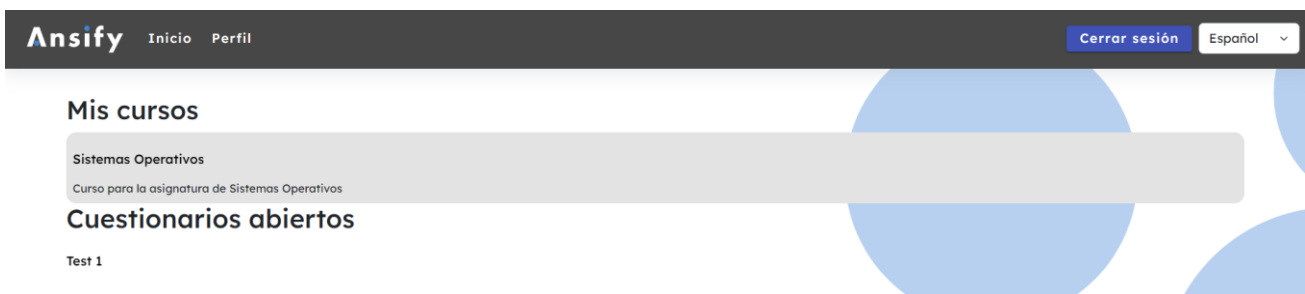
En la vista de detalle del curso, seleccionamos la pestaña de Juegos, donde podremos ver los últimos juegos jugados. Si pinchamos en alguno de ellos, veremos el resultado de cada uno de los alumnos que haya participado.



*Ilustración 116 Pantalla de resultado de un juego como profesor*

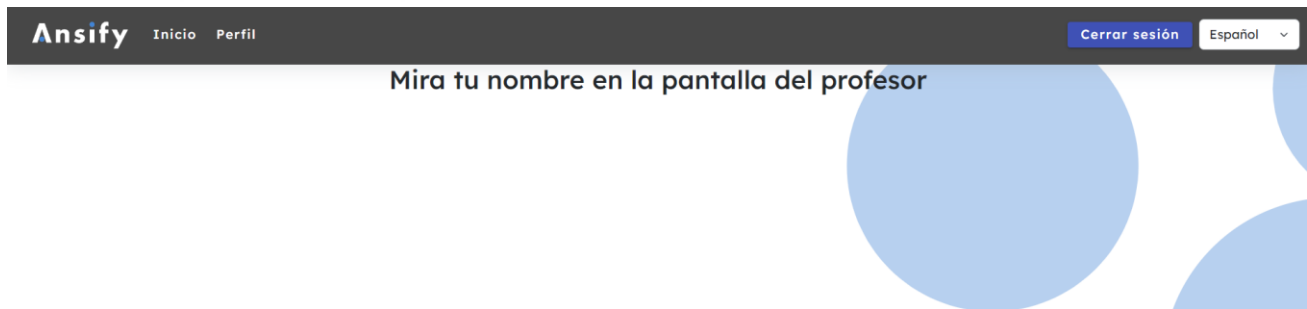
### 8.6.3.10 Jugar un cuestionario como alumno

Para jugar un juego con el alumno, simplemente habrá que esperar en la pantalla de inicio y esperar a que el profesor abra el cuestionario. Este aparecerá automáticamente en la pantalla. De no ser así, actualizar la página. Para pasar a jugar habrá que pulsar simplemente sobre el juego.



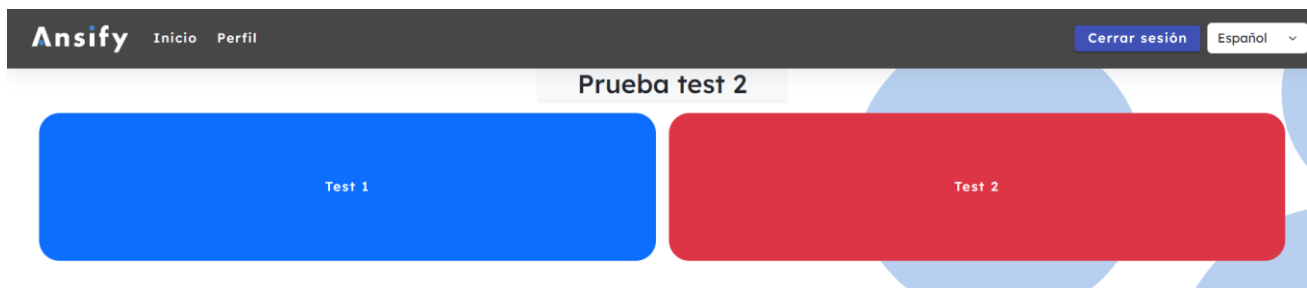
*Ilustración 117 Pantalla de inicio del alumno*

Una vez le demos, el alumno se encontrará en la sala de espera, donde podrá ver su nombre en la pantalla del profesor.



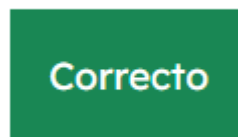
*Ilustración 118 Sala de espera del alumno*

El profesor le dará a empezar y se verá una previsualización de la pregunta en la pantalla del profesor. Una vez finalicé el tiempo de previsualización, el alumno podrá responder a la pregunta simplemente pulsando en la respuesta que crea que es la correcta. En el caso de las preguntas de respuesta corta, necesitará que la escriba y acepte.

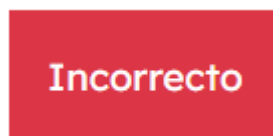


*Ilustración 119 Pantalla de pregunta del alumno*

Cuando termine el tiempo de la pregunta, el sistema mostrará al usuario si la respuesta es correcta o no.



*Ilustración 120 Respuesta correcta*



*Ilustración 121 Respuesta incorrecta*

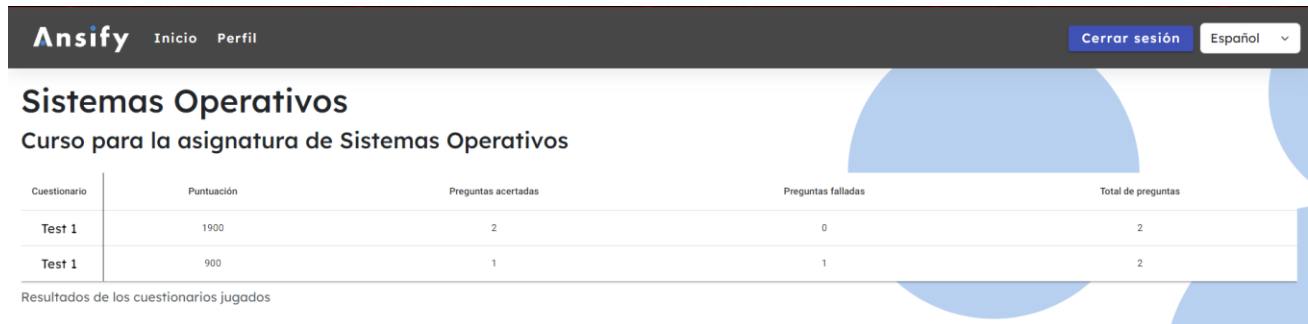
La siguiente pantalla mostrará la puntuación que lleva el alumno hasta el momento. La partida seguirá igual hasta que finalicen todas las preguntas del cuestionario.

## Puntuación: 900

### Ilustración 122 Puntuación del alumno

#### 8.6.3.11 Consultar resultados y revisar cuestionario como alumno

En la pantalla principal del alumno mostrada antes, el jugador podrá pinchar en el nombre de uno de los cursos y pasará a ver los resultados de todos los cuestionarios que haya jugado para ese curso.

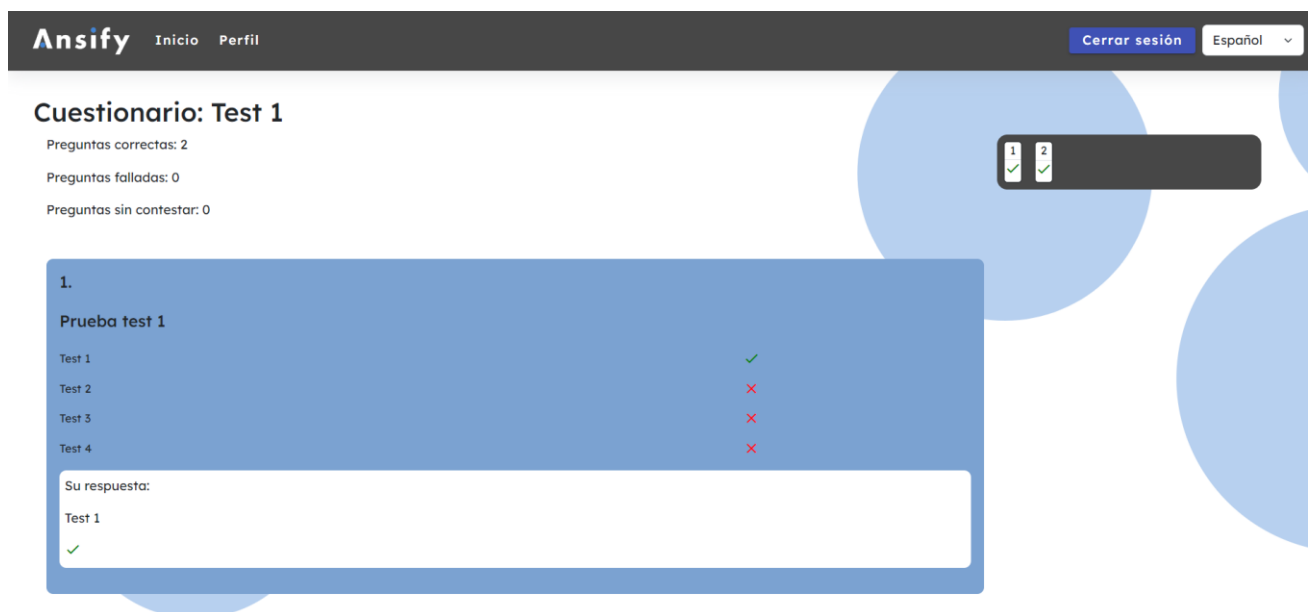


Results of the questionnaires played

Cuestionario	Puntuación	Preguntas acertadas	Preguntas falladas	Total de preguntas
Test 1	1900	2	0	2
Test 1	900	1	1	2

### Ilustración 123 Pantalla de detalles de curso del alumno

Además, podrá pinchar sobre el nombre de cualquiera de los cuestionarios jugados y podrá revisar las preguntas y respuestas del cuestionario.



Cuestionario: Test 1

Preguntas correctas: 2  
Preguntas falladas: 0  
Preguntas sin contestar: 0

1. Prueba test 1

Test 1	✓
Test 2	✗
Test 3	✗
Test 4	✗

Su respuesta:

Test 1  
✓

### Ilustración 124 Pantalla de revisión de cuestionario

#### 8.6.3.12 Modificar contraseña

El alumno podrá navegar al perfil utilizando la barra de navegación superior. Una vez dentro del perfil podrá cambiar la contraseña indicando la contraseña actual y la nueva. Además, en la pantalla del perfil también se mostrarán los resultados de los últimos cuestionarios que haya jugado.



**UO277603**

### Modificar contraseña

Contraseña actual\*

Contraseña nueva\*

Aceptar

### Últimos cuestionarios jugados

Sistemas Operativos	Test 1	Puntuación: 1900
Sistemas Operativos	Test 1	Puntuación: 900

*Ilustración 125 Pantalla del perfil*

# Capítulo 9 CONCLUSIONES Y AMPLIACIONES





## 9.1 CONCLUSIONES

---

Una vez concluido el proyecto, puedo decir que estoy orgulloso del resultado final. El objetivo principal de este proyecto era llevar a cabo cuestionarios en línea de manera simultánea entre el profesor y los alumnos, aspecto que ha sido probado exitosamente. Asimismo, me gustaría haber añadido más funcionalidades que comentaré en el siguiente apartado, sin embargo, creo que el sistema ha quedado lo suficientemente acotado como para concluir que su construcción se ha realizado de manera satisfactoria.

A lo largo del proyecto, han ido surgiendo una serie de complicaciones debido a ciertas decisiones que se han tomado. Una de ellas ha sido la elección de tecnologías nuevas, que me han supuesto un reto aprenderlas, aunque me siento realizado con ello. Por otro lado, me encontré con otra serie de limitaciones, como, por ejemplo, realizar pruebas automáticas para los cuestionarios en línea. Desde la fase de análisis se tenía previsto realizar pruebas de carga y e2e de esta funcionalidad, pero debido a su naturaleza, cómo se ha comentado en la especificación técnica de las pruebas, no he sido capaz de realizarlas debida a su alta complejidad. Un problema que se identificó como riesgo y, finalmente ocurrió, es que al desplegar la aplicación no funcionaba correctamente, pues el protocolo de comunicación de websockets necesita que la comunicación entre la aplicación web y la api sea mediante HTTPS, por lo que tuve que investigar cómo realizarlo. Otro de los grandes problemas es el tiempo, a pesar de que la planificación ha ido bastante bien, la presión de la fecha de entrega siempre estaba ahí, y cuanto más se acercaba, con más estrés tenía que lidiar.

A nivel personal, puedo decir que el conocimiento adquirido en el proceso de creación de este trabajo ha merecido totalmente la pena, pues he podido comprobar personalmente cómo es crear un proyecto desde cero, pasando por todas las fases del desarrollo, hasta finalmente verlo terminado. Además, el haber seleccionado tecnologías nuevas con las que nunca antes había tratado me ha hecho ver que soy capaz de afrontar situaciones difíciles con tecnologías que no controlo totalmente, y ahora al final, puedo decir que haberlas aprendido me han aportado mucho, no solo a nivel académico y de formación, si no a nivel personal. Además, estoy muy satisfecho con el uso del framework Angular, puesto que me ha parecido muy sencillo de aprender y creo que es perfecto para un proyecto como este.

Para finalizar, quiero decir que, pese a todas las decisiones que he tomado durante todo el proyecto, ya sean buenas o malas, todas me han servido para aprender y solidificar todos los conocimientos adquiridos a lo largo de toda la carrera y para experimentar cómo se lleva a cabo la implementación de un proyecto de esta envergadura.



## 9.2 AMPLIACIONES

---

Cómo se ha comentado en el apartado anterior, hubiera sido interesante añadir más funcionalidades y aspectos al sistema para hacerlo más completo, pero debido al tiempo de realización, haberlas hecho hubiera aumentado en gran medida, no solo el tiempo, si no la complejidad del proyecto.

### 9.2.1 Realización de exámenes tipo test

Una funcionalidad muy interesante hubiera sido darles la capacidad a los usuarios de tipo profesor generar exámenes tipo test con la batería de preguntas asociadas un curso. Así, podría generar diferentes modelos de un examen con las preguntas de un curso que selecciona e, incluso, generar un modelo de examen distinto para cada uno de los alumnos que pertenecen al curso. Una vez hecho el examen, se descargará el PDF de los exámenes, se deja tiempo para hacerlos y, al final, se usa el sistema para mecanizar la corrección de estos.

### 9.2.2 Aplicación para móvil

Otra opción que me hubiera gustado añadir, pero dentro del tiempo estipulado era imposible, era la realización de aplicaciones móvil, ya sea para Android, iOS o ambas. La idea era realizar una aplicación con la misma funcionalidad que la parte del estudiante de la aplicación web. Se optó por otra vía que disminuía el tiempo de finalización, que fue realizar la aplicación web accesible desde móvil, sin embargo, el usuario suele ver con mejores ojos una aplicación móvil nativa que una aplicación web.

### 9.2.3 Pruebas automáticas para los cuestionarios en línea

Una de las limitaciones del proyecto fue no poder realizar pruebas automáticas de rendimiento ni e2e en los cuestionarios en línea. Con más tiempo, me hubiera gustado investigar más sobre cómo realizar este tipo de pruebas, sobre todo las de rendimiento, puesto que hubiera sido muy interesante ver cómo se comporta el sistema con varios cuestionarios siendo jugados a la vez o ver cómo gestiona el servidor a muchos alumnos simultáneos en un cuestionario.

### 9.2.4 Realizar cuestionarios en diferido

Esta funcionalidad sería complementaria a jugar los cuestionarios en línea. La idea es poder elegir un cuestionario para dejar a los alumnos que los respondan de manera autónoma dentro de un plazo establecido, así como se realizan los cuestionarios en el campus virtual de la universidad.



## 9.2.5 Poder filtrar los datos que se muestran

Esta ampliación tampoco se ha hecho por falta de tiempo. Se trata de poder filtrar y buscar en los datos mostrados por la aplicación. Por ejemplo, un profesor puede llegar a crear una gran cantidad de preguntas y cuestionarios, lo ideal sería darle la opción de filtrar por nombre, curso o cualquier otro campo que sea relevante, así la búsqueda de cuestionarios sería mucho más rápida. Esto se aplicaría tanto a las preguntas, cursos y resultados por parte del profesor, a la lista de usuarios y cursos del sistema por parte del administrador y a la lista de cursos, juegos y resultados de los alumnos.

## 9.2.6 Autenticación utilizando credenciales de Uniovi.

Por último, y una funcionalidad que encajaría perfectamente en el ámbito de este proyecto es la posibilidad de iniciar sesión en la aplicación utilizando las propias credenciales de uniovi. Esto haría que la experiencia del usuario mejorase considerablemente al no tener que controlar otra cuenta más, simplemente utilizan la suya propia de la universidad.





# Capítulo 10 BIBLIOGRAFÍA

- [1] J. Redondo, «Creación y evaluación de plantillas para trabajos de fin de grado como buena práctica docente.,» *Revista de Innovación y Buenas Prácticas Docentes*, p. pp, 2020.
- [2] «Single-page application,» Wikipedia, 1 Junio 2023. [En línea]. Available: [https://es.wikipedia.org/wiki/Single-page\\_application](https://es.wikipedia.org/wiki/Single-page_application). [Último acceso: 14 Marzo 2023].
- [3] «WebSockets,» mdn, 21 Febrero 2023. [En línea]. Available: [https://developer.mozilla.org/es/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/es/docs/Web/API/WebSockets_API). [Último acceso: 15 Marzo 2023].
- [4] «¿Qué es una API de REST?,» Red Hat, 8 Mayo 2020. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. [Último acceso: 15 Marzo 2023].
- [5] «Transferencia de Estado Representacional,» Wikipedia, 15 Marzo 2023. [En línea]. Available: [https://es.wikipedia.org/wiki/Transferencia\\_de\\_Estado\\_Representacional](https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional). [Último acceso: 15 Marzo 2023].
- [6] «Amazon API Gateway,» Amazon, [En línea]. Available: <https://aws.amazon.com/es/api-gateway/>. [Último acceso: 1 Marzo 2023].
- [7] «How it works,» Socket.IO, 23 Mayo 2023. [En línea]. Available: <https://socket.io/docs/v4/how-it-works/#socketio>. [Último acceso: 26 Junio 2023].
- [8] D. L. P. D. Aquilino Adolfo Juan Fuente, «Guía de Aprendizaje de la asignatura de Dirección y Planificación de Proyectos Informáticos,» de *Guía de Aprendizaje de la asignatura de Dirección y Planificación de Proyectos Informáticos*, Oviedo, 2022, p. 301.
- [9] H. Vocke, «The Practical Test Pyramid,» 26 Febrero 2018. [En línea]. Available: <https://martinfowler.com/articles/practical-test-pyramid.html>. [Último acceso: 12 Abril 2023].
- [10] «Prisma schema,» Prisma, [En línea]. Available: <https://www.prisma.io/docs/concepts/components/prisma-schema>. [Último acceso: 18 abril 2023].
- [11] «TypeScript,» Wikipedia, 20 Abril 2023. [En línea]. Available: <https://es.wikipedia.org/wiki/TypeScript>. [Último acceso: 2 Mayo 2023].



- 
- [12] «Node.js,» Wikipedia, 21 Junio 2023. [En línea]. Available: <https://en.wikipedia.org/wiki/Node.js>. [Último acceso: 3 Mayo 2023].
- [13] «HTML: Lenguaje de etiquetas de hipertexto,» mdn, 11 Junio 2023. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTML>. [Último acceso: 3 Mayo 2023].
- [14] «CSS,» mdn, 13 Marzo 2023. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/CSS>. [Último acceso: 4 Mayo 2023].
- [15] «Rooms,» Socket.IO, 23 Mayo 2023. [En línea]. Available: <https://socket.io/docs/v3/rooms/>. [Último acceso: 24 Mayo 2023].
- [16] «Jest (Framework),» Wikipedia, 28 Abril 2023. [En línea]. Available: [https://en.wikipedia.org/wiki/Jest\\_\(framework\)](https://en.wikipedia.org/wiki/Jest_(framework)). [Último acceso: 28 Mayo 2023].
- [17] «Jasmine (software),» Wikipedia, 23 Diciembre 2022. [En línea]. Available: [https://en.wikipedia.org/wiki/Jasmine\\_\(software\)](https://en.wikipedia.org/wiki/Jasmine_(software)). [Último acceso: 28 Mayo 2023].
- [18] A. Hussain, «Jasmine & Karma,» codecraft, [En línea]. Available: <https://codecraft.tv/courses/angular/unit-testing/jasmine-and-karma/>. [Último acceso: 28 Mayo 2023].
- [19] «Getting Started,» TestCafé, [En línea]. Available: <https://testcafe.io/documentation/402635/guides/overview/getting-started>. [Último acceso: 9 Junio 2023].
- [20] Project Management Institute, Inc., «PMBOK,» de *A Guide to the Project Management Body of Knowledge*, Pennsylvania, 2013, p. 616.
- [21] I. Llana, J. M. Redondo y L. Vinuesa, «Towards Lightweight Mobile Pentesting Tools to Quickly Assess Machine Security Levels,» *IEEE Latin American Transactions*, p. pp, 2019.
- [22] «Salario medio para Programador en España, 2023,» 2023. [En línea]. Available: <https://es.talent.com/salary?job=programador#:~:text=El%20salario%20programador%20promedio%20en,hasta%20%E2%82%AC%2038.292%20al%20a%C3%B1o..> [Último acceso: 3 Julio 2023].

# APÉNDICES





## PLAN DE GESTIÓN DE RIESGOS

En este apartado se especifica cómo se va a llevar a cabo la gestión de riesgos del proyecto, tanto su análisis y especificación, como su tratamiento.

### Metodología

La metodología usada será la metodología de Boehm [8]. Dicha metodología está dividida en dos fases:

1. **Valoración de los riesgos**, donde se realizará la identificación, análisis y priorización de éstos.
2. **Control de riesgos**, donde se especificará las diferentes estrategias a aplicar sobre los riesgos.

### Identificación de los riesgos

Riesgo	Descripción
Bajo rendimiento y escalabilidad	A medida que la cantidad de usuarios y cuestionarios aumente, se podrán encontrar problemas relacionados con el rendimiento, que afecte directamente al sistema.
Mal diseño y experiencia de usuario	Es uno de los aspectos clave del sistema. El riesgo radica en la creación de una interfaz de usuario intuitiva y atractiva fácil de usar.
Gestión del tiempo y planificación	El tiempo es un recurso crítico en un proyecto como este. El riesgo de una mala gestión del tiempo o una planificación inadecuada puede llevar a retrasos y dificultades para cumplir con los plazos establecidos.
Algún módulo deje de tener soporte	Algún modulo necesario usado en el sistema, ya sea en la aplicación web o en el servidor, deja de tener soporte.
No comprender cómo realizar los juegos en línea	Dado que es la funcionalidad principal del sistema, es crucial entender correctamente cómo hacerla y con qué tecnología.
Problemas al desplegar la aplicación en un servidor	Debido a que la mayor parte del desarrollo se realiza en local, existe un riesgo de que algo no funcione correctamente a la hora de desplegar la aplicación en el servidor.
Alcance excesivo	Existe el riesgo de querer abordar demasiado, lo que podría dificultar entregar el proyecto dentro del plazo.
Pruebas insuficientes	La falta de pruebas puede conducir a problemas no detectados y a fallos en la aplicación.



Problema de seguridad	Las aplicaciones web que involucran usuarios y datos sensibles deben tener medidas de seguridad adecuadas.
Baja médica	Es un riesgo que siempre existe y puede hacer que se retrase el proyecto.

### Clasificación de los riesgos

Según la Guía de Aprendizaje de la asignatura de Dirección y Planificación de Proyectos Informáticos [8] existen tres tipos de riesgos principales:

- Riesgos de **proyecto**: afectan directamente a la planificación o a los recursos.
- Riesgos del **producto**: afectan a la calidad o al desempeño del software por desarrollarse.
- Riesgos del **negocio**: aquellos que afectan a la organización que desarrolla el software y a la viabilidad del propio software.

Vistas las categorías, a continuación, se muestran la lista de riesgos categorizados:

Riesgo	Categoría
Bajo rendimiento y escalabilidad	Producto
Mal diseño y experiencia de usuario	Producto
Gestión del tiempo y planificación	Proyecto
Algún módulo deje de tener soporte	Negocio
No comprender cómo realizar los juegos en línea	Producto
Problemas al desplegar la aplicación en un servidor	Producto
Alcance excesivo	Proyecto
Pruebas insuficientes	Producto
Problema de seguridad	Producto
Baja médica	Proyecto

### Análisis y priorización de los riesgos

Para llevar a cabo el análisis y la posterior priorización es preciso definir la matriz de probabilidad e impacto proporcionada por el PMBOK [20]:

Probabilidad	Amenazas					Oportunidades				
	0,90	0,05	0,09	0,18	0,36	0,72	0,72	0,36	0,18	0,09
0,70	0,04	0,07	0,14	0,28	0,56	0,56	0,28	0,14	0,07	0,04
0,50	0,03	0,05	0,10	0,20	0,40	0,40	0,20	0,10	0,05	0,03
0,30	0,02	0,03	0,06	0,12	0,24	0,24	0,12	0,06	0,03	0,02
0,10	0,01	0,01	0,02	0,04	0,08	0,08	0,04	0,02	0,01	0,01
	0,05 Muy bajo	0,10 Bajo	0,20 Moderado	0,40 Alto	0,80 Muy Alto	0,80 Muy Alto	0,40 Alto	0,20 Moderado	0,10 Bajo	0,05 Muy Bajo
	Impacto Negativo					Impacto Positivo				

**Tabla 12 MATRIZ DE PROBABILIDAD E IMPACTO**

Como en este proyecto no se han identificado ninguna oportunidad, solo se tendrá en cuenta la parte izquierda de la matriz, la correspondiente a las amenazas. A continuación, una vez analizados los riesgos, se muestra la lista de riesgos ordenados por prioridad:

Prioridad	Riesgo	Prob.	Presup.	Planfic.	Alcance	Calidad	Impacto
1	No comprender cómo realizar los juegos en línea	Media	Alto	Alto	Alto	Crítico	0.45
2	Mal diseño y experiencia de usuario	Media	Bajo	Bajo	Bajo	Crítico	0,45
3	Bajo rendimiento y escalabilidad	Alta	Alto	Medio	Bajo	Alto	0,0,39
4	Gestión del tiempo y planificación	Media	Alto	Alto	Alto	Bajo	0,28
5	Pruebas insuficientes	Media	Bajo	Medio	Inapreciable	Alto	0,28
6	Problemas al desplegar la aplicación en un servidor	Muy alta	Bajo	Bajo	Bajo	Medio	0,27
7	Algún módulo deje de tener soporte	Baja	Medio	Medio	Bajo	Bajo	0,09
8	Alcance excesivo	Muy baja	Medio	Alto	Crítico	Bajo	0,09
9	Baja médica	Muy baja	Crítico	Crítico	Crítico	Crítico	0,09
10	Problema de seguridad	Muy baja	Bajo	Bajo	Bajo	Bajo	0,02

**Tabla 13 TABLA DE RIESGOS PRIORIZADOS**

## Estrategia y respuesta ante los riesgos

Las diferentes estrategias con las que serán enfrentados los riesgos son las siguientes:

- Transferir.
- Mitigar.
- Aceptar.
- Eliminar.

A continuación, se muestra la lista de riesgos una vez aplicadas las estrategias a los riesgos:

Riesgo	Respuesta	Estrategia
Bajo rendimiento y escalabilidad	En el caso de que esto suceda, habrá que hacer una refactorización y optimización en el código.	Asumir el riesgo
Mal diseño y experiencia de usuario	Para evitar que el usuario tenga una mala experiencia con el uso de la aplicación, se desarrollarán pruebas de usabilidad que ayuden a corregir los errores del diseño de la interfaz.	Eliminar el riesgo
Gestión del tiempo y planificación	Se llevará a cabo una planificación usando la herramienta de gestión de proyectos Microsoft Project. Puede darse el caso de que finalmente no se use, aun así, es necesario llevar un seguimiento de las actividades que se van haciendo, así como el tiempo que llevan.	Mitigar el riesgo
Algún módulo deje de tener soporte	No queda otra que asumir el riesgo y sustituir el módulo por otro que realiza una funcionalidad similar, o programar nosotros mismos dicha funcionalidad.	Asumir el riesgo
No comprender cómo realizar los juegos en línea	Se realizará un estudio previo a la construcción del sistema para obtener la información necesaria para llevar a cabo esta funcionalidad.	Eliminar el riesgo
Problemas al desplegar la aplicación en un servidor	Cuando llegue el momento de desplegar, si ocurren problemas debemos solucionarlos.	Asumir el riesgo
Alcance excesivo	Se intentará realizar un alcance del proyecto realista y factible para que esto no ocurra.	Eliminar el riesgo
Pruebas insuficientes	Se realizará una especificación de pruebas exhaustiva para probar los diferentes aspectos de la aplicación.	Eliminar el riesgo



Problema de seguridad	Debido a que la aplicación no trata con datos sensibles, el riesgo se asume.	Asumir el riesgo
Baja médica	Es algo que no se puede predecir, por lo que no queda otra que asumirlo y actuar en consecuencia.	Asumir el riesgo

**Tabla 14 ESTRATEGIA Y RESPUESTA ANTE LOS RIESGOS**





## CONTENIDO ENTREGADO

---

A continuación, se muestra el contenido entregado junto con esta documentación en fichero .zip.

Directorio	Contenido
<b>./Directorio raíz</b>	Contiene el directorio correspondiente al proyecto, la planificación, la hoja de riesgos y la carpeta de los diagramas.
<b>./AudienceResponseSystem</b>	Contiene todo el sistema construido, tanto la API Rest, como la aplicación web.
<b>./AudienceResponseSystem/restapi</b>	Subsistema de la API Rest.
<b>./AudienceResponseSystem/webapp</b>	Subsistema de la aplicación web.
<b>./Planificación.mpp</b>	Fichero de Microsoft Project correspondiente a la planificación.
<b>./Riesgos.xlsx</b>	Fichero Excel donde está la lista de riesgos identificados.
<b>./Diagramas</b>	Directorio donde se sitúan todos los diagramas vistos en el documento.

**Ilustración 126** Tabla del contenido entregado