



ESCUELA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA INFORMÁTICA

Sistema de realidad aumentada para la evaluación del alumnado
(vAnswer) y panel de gestión docente.

José Rubén Rivas Troncoso

10 de agosto de 2019

GRADO EN INGENIERÍA INFORMÁTICA

Sistema de realidad aumentada para la evaluación del alumnado (vAnswer) y panel de gestión docente.



ESCUELA SUPERIOR DE INGENIERÍA

- ◆ Departamento: Ingeniería Informática.
- ◆ Director del proyecto: Guadalupe Ortiz Bellot.
- ◆ Departamento: Ingeniería en Automática, Electrónica, Arquitectura y Redes de Computadores.
- ◆ Codirector: Alfonso García de Prado Fontela.

Autor del proyecto: José Rubén Rivas Troncoso

Cádiz, 10 de agosto de 2019

Fdo: José Rubén Rivas Troncoso

Agradecimientos

A mis padres, mi hermano Rafael, el cual me ha ayudado todo el tiempo que ha podido para aconsejarme en muchas facetas de este trabajo, y amigos, los cuales se han prestado para muchas de las pruebas que he podido realizar para el proyecto.

A todos y cada uno de los profesores que he tenido en los últimos 5 años, a alguno de los que he tenido antes y a mis compañeros de clase.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyleft 2019 José Rubén Rivas Troncoso.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Notación y formato

Cuando nos refiramos a un programa en concreto, utilizaremos la notación:
unity.

Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación:
youtube-dl.

Tabla de contenidos

1. Motivación y contexto del proyecto	1
1.1. Motivación.....	1
1.2. Contexto del proyecto.....	1
1.3. Objetivos.....	2
1.3.1. Lectura de datos de base de datos.....	2
1.3.2. Procesamiento de los datos.....	2
1.3.3. Almacenamiento de los datos.....	2
1.3.4. Ordenamiento y adecuación de los datos.....	2
1.3.5. Presentación de los datos.....	3
1.3.6. Gestión administrativa.....	3
1.3.7. Seguridad en la gestión docente.....	3
1.4. Estructura del documento.....	3
2. Descripción general del proyecto	5
2.1. Descripción general.....	5
2.2. Visión del sistema.....	6
2.3. Características del usuario de la aplicación.....	7
2.3.1. Usuario alumno.....	7
2.3.2. Usuario docente/administrador.....	7
2.4. Contexto tecnológico.....	7
2.4.1. Unity y Vuforia.....	8
2.4.2. Eclipse.....	8
2.4.3. MySQL.....	8
2.4.4. C#.....	8
2.4.5. PHP.....	8
2.4.6. PHPMailer.....	9
2.4.7. Java.....	9
2.4.8. HTML5 y CSS3.....	9
2.4.9. JavaScript.....	9
2.5. Restricciones generales del proyecto.....	10
2.5.1. Herramienta de control de versiones.....	10
2.5.2. Herramientas auxiliares.....	10
2.5.3. Calidad del código.....	11
3. Planificación	13
3.1. Comienzo del proyecto.....	13
3.2. Análisis de requisitos y recursos.....	14
3.3. Aprendizaje y perfeccionamiento.....	14
3.4. Desarrollo.....	16
3.5. Testing.....	16

3.6. Correcciones.....	17
3.7. Memoria.....	17
3.8. Diagrama de Gantt.....	17
3.9. Modelo de ciclo de vida.....	17
4. Análisis	19
4.1. Especificación de requisitos del sistema.....	19
4.1.1. Requisitos de interfaces externas.....	19
4.1.2. Requisitos funcionales.....	22
4.1.3. Requisitos de rendimiento.....	23
4.1.4. Restricciones de diseño.....	23
4.1.5. Requisitos de sistema software.....	23
4.2. Modelo de casos de uso.....	24
4.2.1. Diagramas de casos de uso.....	24
4.2.2. Descripción de los casos de uso.....	25
4.3. Modelo de comportamiento del sistema.....	31
4.3.1. Diagramas de secuencia.....	31
5. Diseño	35
5.1. Esquema de base de datos.....	35
5.2. Interfaz gráfica del portal web.....	36
5.3. Interfaz de la aplicación móvil.....	38
6. Implementación	45
6.1. Arquitectura general del sistema.....	45
6.1.1. Controladores (ficheros de código C#).....	46
6.1.2. Componente de realidad aumentada.....	46
6.1.3. Escenas y componentes de la interfaz de usuario.....	48
6.1.4. Ficheros PHP, HTML5 y CSS3.....	49
6.1.5. Servicio REST.....	49
6.1.6. Comunicación con la aplicación móvil.....	50
6.1.7. Comunicación con panel de registro/login.....	50
6.2. Proceso de obtención de información.....	51
7. Pruebas	53
7.1. Tests unitarios.....	53
7.1.1. Pruebas unitarias del servicio REST.....	54
7.1.2. Pruebas unitarias de la aplicación móvil.....	54
7.1.3. Pruebas unitarias del panel de gestión.....	55
7.2. Test de integración.....	55
7.2.1. Resultados.....	55
8. Conclusiones y trabajo futuro	57
8.1. Valoración del proyecto.....	57
8.2. Objetivos cumplidos.....	58
8.3. Lecciones aprendidas.....	59
8.3.1. Conocimientos técnicos.....	59
8.3.2. Conocimientos adquiridos en el ámbito personal.....	61

8.4. Mejoras y ampliaciones: Trabajo futuro.....	62
8.4.1. Mejoras.....	62
8.4.2. Ampliaciones al proyecto actual.....	63
Bibliografía y referencias	65
A. Programas utilizados	67
A.I. Eclipse IDE.....	67
A.II. Advanced REST Client.....	69
A.III. PuTTY.....	70
A.IV. Unity.....	71
A.V. MySQL Workbench.....	72
A.VI. Gantt Project.....	72
B. Manual del usuario	73
B.I. Rol administrador.....	79
B.II. Rol docente.....	79
B.III. Rol alumno.....	80
C. Descarga del proyecto	81
D. GNU Free Documentation License	83
1. APPLICABILITY AND DEFINITIONS.....	83
2. VERBATIM COPYING.....	85
3. COPYING IN QUANTITY.....	85
4. MODIFICATIONS.....	86
5. COMBINING DOCUMENTS.....	87
6. COLLECTIONS OF DOCUMENTS.....	88
7. AGGREGATION WITH INDEPENDENT WORKS.....	88
8. TRANSLATION.....	88
9. TERMINATION.....	89
10. FUTURE REVISIONS OF THIS LICENSE.....	89
11. RELICENSING.....	90
12. How to use this License for your documents.....	90

Índice de figuras

2.1. Situación del alumno.....	5
2.2. Ecosistema de tecnologías.....	6
2.3. Fragmento de código C# de ejemplo.....	11
3.1. Diagrama de Gantt.....	18
4.1. Modelo de casos de uso del sistema.....	24
4.2. Vincular/desvincular profesor de asignatura.....	31
4.3. Dar de alta un docente.....	32
4.4. Registro de docente.....	32
4.5. Restablecimiento de contraseña.....	33
4.6. Gestionar asignaturas y calificaciones.....	33
4.7. Secuencia de juego de la aplicación móvil.....	34
5.1. Esquema de base de datos relacional.....	35
5.2. Pantalla de gestión administrativa.....	37
5.3. Panel de gestión docente.....	37
5.4. Panel de registro e inicio de sesión.....	38
5.5. Escena portada	39
5.6. Escena inicio.....	40
5.7. Escena elección modo.....	40
5.8. Escena elección asignatura.....	41
5.9. Escena realidad aumentada con controles.....	41
5.10. Escena cuestionario.....	42
5.11. Escena fin del juego.....	42
5.12. Escena examen no disponible.....	43
5.13. Escena repaso completado.....	43
6.1. Aspecto de la arquitectura del sistema.....	45
A.1. Pantalla de inicio de Eclipse.....	68
A.2. Interfaz Eclipse IDE.....	68
A.3. Interfaz Advanced REST client.....	69
A.4. Interfaz PuTTY.....	70
A.5. Interfaz entorno Unity.....	71
A.6. Interfaz MySQL Workbench.....	72
A.7. Interfaz Gantt Project.....	72
B.1. Instalación FortiClient VPN.....	74
B.2. Interfaz configuración VPN.....	75
B.3. Configuración avanzada VPN	76
B.4. Conexión a VPN.....	77
B.5. Conectado al VPN.....	78

Capítulo 1

Motivación y contexto del proyecto

1.1. Motivación

Este proyecto fin de carrera surge debido a que un grupo de docentes perteneciente al departamento de innovación toma la idea de desarrollar un producto software con uso de realidad aumentada, aportando a la docencia un ingrediente que pueda ser beneficioso. A su vez, para mí es una idea conveniente de llevar a cabo, ya que además de disfrutar con el aspecto de la realidad aumentada, me permitirá desarrollar un producto que favorezca al profesorado y a los alumnos.

Si conseguimos sacar el mayor provecho a la aplicación, hará la vida más fácil tanto a profesores como a alumnos, y eso es algo que por mi parte pienso que ayudará mucho al desarrollo de las asignaturas, sobre todo en aquellas cuyas partes se hacen más “pesadas” por llamarlo de algún modo.

1.2. Contexto del proyecto

Como ya sabemos, nos encontramos en la situación en la que como comentábamos antes, muchas asignaturas se hacen duras o cansinas de cursar. A veces y en muchos alumnos ocasiona la decisión de no cursar dichas asignaturas incluso, lo cual tiene un impacto negativo en la variedad de materias (siempre se tienden a cursar las mismas).

Planteemos como ejemplo la siguiente ocasión:

Manuel, un alumno estudiante de Ingeniería Informática en tercer curso, pronto deberá decidir en que optativas de cuarto año matricularse y para ello recurre a sus conocidos compañeros para tomar consejo de ellos. Como es de costumbre, Manuel recibirá el siguiente mensaje:

“Nosotros elegiremos las asignaturas X e Y ya que son las más fáciles para aprobar y las imparte el profesor Z”.

Manuel se dejará guiar por dichas personas y tomará como optativas las que sus amigos/compañeros les aconsejen, obviando totalmente la parte del aprendizaje (que puede ser una gran pérdida).

A pesar de la influencia negativa que podamos tener a nuestro alrededor, existe un porcentaje que es el que nosotros intentaremos reducir con este trabajo. Este porcentaje corresponde con introducir una nueva forma de evaluar los alumnos para hacer que de las “peores” asignaturas se hable de buena forma.

1.3. Objetivos

El objetivo principal de este proyecto consiste en desarrollar una aplicación móvil para el alumnado que permita evaluarlos, así como permitirles a ellos hacer uso de la misma desde casa para practicar, repasar e incluso realizar exámenes sobre las asignaturas en el aula.

Por otro lado, como objetivo de cara al personal docente, se pretende desarrollar un panel de gestión que permita al profesorado gestionar dichas asignaturas y sus preguntas. Todo ello consta de una serie de subobjetivos.

Estos subobjetivos son los siguientes:

- Lectura de datos de una base de datos.
- Procesamiento de los datos.
- Almacenamiento de datos.
- Ordenamiento y adecuación de los datos.
- Gestión de administración.
- Seguridad de la gestión docente.

1.3.1. Lectura de datos de la base de datos

Lo primero a tener en cuenta es que nuestro sistema tiene como pilar fundamental cumplir con la correcta lectura de los datos, y para ello trabajaremos en un diseño de una base de datos adecuada a la necesidad que vamos a tener.

Dichas lecturas de datos serán realizadas mediante diversas formas, entre ellas, mediante el lenguaje C# [1] respecto a los ficheros desarrollados en el entorno *Unity* [2], como también los ficheros de código en el lenguaje PHP [3] alojados en el servidor de la universidad.

1.3.2. Procesamiento de los datos

Una vez obtenidos los datos en el subobjetivo previo, estos son enviados a la aplicación donde posteriormente pasando por una cierta ordenación serán presentados al alumno usuario de la aplicación.

1.3.3. Almacenamiento de los datos

El almacenamiento de los datos es otro factor clave en nuestro sistema. Será el que nos permite almacenar en la base de datos los resultados de los exámenes realizados por los alumnos y permitirá almacenar las nuevas asignaturas y sus preguntas por parte de la gestión del docente.

1.3.4. Ordenamiento y adecuación de los datos

Los datos una vez leídos y tratados por el sistema, son codificados/decodificados tanto a la hora de operar sobre la base de datos como cuando son obtenidos de ella de cara a ser mostrados al alumno o al docente en el panel de gestión.

1.3.5. Presentación de los datos

En el sistema existen diferentes elementos que mostrarán los datos al usuario de manera adecuada y con un mínimo de estética y posicionamiento.

1.3.6 Gestión de administración

Según como se encuentra el sistema, por la parte de gestión, cabe mencionar que existirá una persona física que será el administrador general del sistema, que tendrá un panel adicional para vincular/desvincular a profesores con temáticas ya existentes.

1.3.7 Seguridad en la gestión docente

Respecto al ámbito de la seguridad, se establecerá un mínimo de la misma, para evitar accesos indeseados al panel de gestión del profesorado tanto por parte del alumnado como de personas ajenas a la actividad mediante un sistema de registro e inicio de sesión, en el cual se verificarán credenciales.

1.4. Estructura del documento

El presente documento está compuesto por las siguientes secciones principales, pudiendo estar divididas éstas, en algunos casos, en subsecciones:

- **Introducción:** Motivación, contexto, objetivos y estructura principal del proyecto.
- **Descripción general:** Descripción detallada del proyecto en sí y de los aspectos más relevantes de este para el lector.
- **Planificación:** Explicación de la planificación realizada para el abordaje del proyecto y de las distintas fases de este.
- **Análisis:** Fase en la que se determinarán cuáles son los requisitos funcionales y no funcionales del sistema y se definirán los diagramas de secuencia y casos de uso.
- **Diseño:** Explicación del diseño del sistema software desarrollado, del diseño lógico, de clases y de las posibles secuencias de ejecución de este.
- **Implementación:** Repaso por los aspectos más importantes de la implementación del sistema. En esta sección se expondrán las relaciones existentes entre las distintas tecnologías usadas para conseguir el comportamiento deseado del sistema, además de los problemas derivados de la implementación del producto.
- **Pruebas:** Diseño del plan de pruebas y ejecución de dicho plan para asegurar el correcto funcionamiento de la aplicación.

- **Conclusiones y trabajo futuro:** Presentación de las conclusiones obtenidas tras haber realizado la investigación y posibles líneas de trabajo futuro.
- **Bibliografía:** Publicaciones y referencias consultadas sobre las fases de análisis, diseño y desarrollo del proyecto.
- **Apéndices:**
 - Herramientas y tecnologías usadas.
 - Manual de usuario.
 - Glosario.

Capítulo 2

Descripción general del proyecto

2.1. Descripción general

Si nos basamos en la naturaleza de casi todo alumno, estos buscan el aspecto más obvio de cara a una asignatura aunque digan lo contrario, siendo este el de conseguir el aprobado. Siendo así nos aprovecharemos de dicho factor de cara a convencer tanto al alumno como a profesores al uso del sistema propuesto. Esto hará ver al alumno que ciertas asignaturas que adopten este método de evaluación como alternativa a los métodos clásicos de evaluación (exámenes finales, parciales, trabajos grupales, prácticas con entregas semanales, exámenes teórico-prácticos), pueden resultar en un beneficio mutuo, tanto para profesor como para alumno. Dicha intención es la que intentamos enfocar con nuestro sistema y no la situación en la que un alumno puede llegar a encontrarse inmerso con otra asignatura una vez mas (observe la Figura 2.1).



Figura 2.1: Situación del alumno

La aplicación evitirá la carga de trabajo por parte del profesorado como suele ser de costumbre en materias donde se necesitan elaborar exámenes con alta carga de contenido y duración, permitiéndoles a dichos docentes introducir las temáticas sobre las que evaluar a los alumnos en su panel de gestión (donde cada docente observará las asignaturas a las que esté vinculado únicamente), añadiendo las preguntas de dichas asignaturas, activando los exámenes cuando corresponda y consultando las calificaciones obtenidas por sus alumnos.

2.2. Visión del sistema

El objetivo de este proyecto es utilizar varias de las tecnologías más populares cuando hablamos de realidad aumentada y comunicación cliente-servidor. En concreto, se utilizan entornos de desarrollo como es *Unity* [2] (en el cualaremos uso de lenguajes como *C#* [1]) con la extensión Vuforia (realidad aumentada), ficheros de código con lenguaje PHP [3] y librería PHPMailer [4] (envío de correos) y JavaScript [5] (que permitirá la comunicación con la aplicación desarrollada en *Unity*, así como con los docentes en el panel de gestión).

A su vez se va a desarrollar un servicio REST [6], exclusivamente para la consulta de calificaciones del alumnado, elaborando un proyecto en el entorno de desarrollo *Eclipse* [7] (donde el servicio se desarrolla mediante código en lenguaje Java [8]) así como el uso de MySQL [9] para todo el manejo de datos en lo que respecta a la base de datos. Finalmente cabe destacar de cara a la presentación y para el manejo por parte del docente tecnologías como HTML5 [10] y CSS3 [11]. Cabe hacer mención a una aplicación a utilizar llamada FortiClient VPN [12], que es vital para el funcionamiento de la app móvil si no estamos bajo la red de la universidad.

Podemos obtener una visión más clara del ecosistema de tecnologías en el siguiente diagrama (2.2):

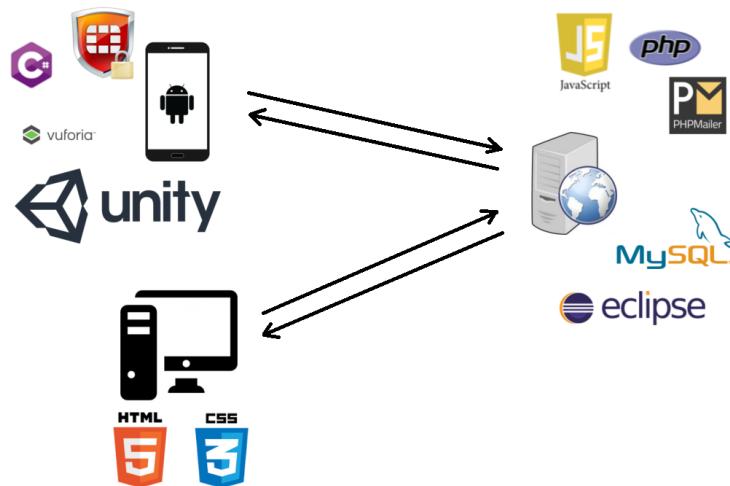


Figura 2.2: Ecosistema de tecnologías

En este diagrama, podemos observar cómo el elemento que nutre de datos al sistema es el servidor (el cual pertenece a la universidad). Este mantiene comunicación tanto con la aplicación móvil como con la parte del panel de gestión docente. Es el encargado de responder con los datos a las peticiones realizadas por la app móvil y a su vez de realizar las operaciones de insertado, eliminado, lectura y actualización en base a las peticiones realizadas por los docentes.

El docente percibirá en el ordenador inferior sus asignaturas permitidas para gestionar (aquellas con las que está vinculado) y añadir nuevas. El alumno por otro lado, en la app móvil recibirá las posibles asignaturas que existan en dicho momento en la base de datos sobre las que poder practicar o examinarse.

2.3. Características del usuario de la aplicación

Aunque la funcionalidad de los usuarios en el sistema es muy clara y puede ser descrita de forma fácil, los tipos de perfiles de los posibles usuarios del mismo se diferencian como ya hemos mencionado. Nos encontramos, dependiendo del uso que pueden hacer del sistema desarrollado, a tres tipos distintos de usuarios:

2.3.1. Usuario alumno

El usuario alumno será el perfil de usuario que será capaz de usar la aplicación móvil, autenticado con LDAP [20].

2.3.2. Usuario docente/administrador

El usuario docente es el perfil de usuario que será capaz gestionar sus asignaturas así como añadir nuevas asignaturas (junto con su vídeo correspondiente), consultar calificaciones, activar/desactivar exámenes, editar/borrar preguntas, insertar preguntas de tipo verdadero/falso e insertar preguntas con cuatro posibles respuestas, todo ello desde la aplicación web.

2.4. Contexto tecnológico

En el diseño de las aplicaciones móviles nativas como es el caso (específica para dispositivos Android [13]), se ha convertido en un asunto ampliamente aceptado y usado en la sociedad, siendo de las más usadas hoy en día. Claros ejemplos son Whatsapp, Facebook...

Las aplicaciones nativas están desarrolladas y optimizadas específicamente para el sistema operativo determinado y la plataforma de desarrollo del fabricante (Android, iOS, etc). Este tipo de aplicaciones se adapta al 100% con las funcionalidades y características del dispositivo obteniendo así una mejor experiencia de uso.

Sin embargo, el desarrollo de una aplicación nativa comporta un mayor coste, puesto que si se desea realizar una aplicación multiplataforma se ha de realizar una nueva versión para cada sistema operativo, multiplicando así los costes de desarrollo.

Dentro del ecosistema de tecnologías presentado, vamos a explicar en qué se basan dichas tecnologías, entornos de desarrollo y lenguajes.

Las tecnologías, entornos usados y lenguajes utilizados son las siguientes:

2.4.1. Unity y Vuforia

En el sistema, la parte de la aplicación móvil se desarrollará en el entorno *Unity* [2], el cual será el pilar fundamental del juego. Vuforia [14] es la extensión de *Unity* para proporcionar el concepto de realidad aumentada permitiéndonos proyectar elementos sobre patrones reconocidos por la aplicación.

2.4.2. Eclipse

Se desplegará en un servidor de la universidad un servicio REST [6], que será lo que desarrollaremos en dicho entorno Eclipse [7], como hemos mencionado previamente, siendo el que nos permitirá consultar posteriormente calificaciones de alumnos basadas en una fecha y asignatura.

2.4.3. MySQL

Es un sistema de gestión de bases de datos relacional, que nos permitirá manejar el grueso del almacenamiento, guardándose información en dicha base de datos (preguntas, respuestas, calificaciones, acceso profesorado, usuarios alumnos, asignaturas, etc) de manera estructurada, haciendo uso de las tablas necesarias acorde a las necesidades.

2.4.4. C#

Es un lenguaje de programación orientado a objetos, utilizado en nuestro caso en ficheros de código en *Unity* [2] (scripts) siendo dicho lenguaje mediante el cual se realiza toda la gestión de la funcionalidad de la aplicación móvil y el encargado de conectar con los ficheros en lenguaje PHP [3] alojados en la máquina servidora para relacionarse con la base de datos, ya que *Unity* no proporciona soporte para conexiones MySQL [9].

2.4.5. PHP

El lenguaje interpretado PHP [3] es el utilizado en una gran cantidad de ficheros alojados en el servidor, desde los encargados de realizar las peticiones a la base de datos para proporcionar información a la aplicación, hasta los ficheros encargados de realizar la gestión por parte del profesorado, así como el encargado de gestionar el apartado de registro/inicio de sesión del personal docente.

2.4.6. PHPMailer

Cabe destacar la librería externa de PHP [3] utilizada en el ámbito de gestión por parte del profesorado, en el cual PHPMailer [4] toma carta en el asunto. Dicho momento donde es usado es cuando el usuario se registra y se envía un correo al administrador para dar de alta a dicho docente, y en segundo uso, para enviar un correo a la dirección del profesor en cuestión, tras este solicitar un restablecimiento de contraseña.

2.4.7. Java

Es un lenguaje de programación orientado a objetos, usado en el código desarrollado en el entorno de *Eclipse* [7] como mencionamos previamente, siendo el lenguaje en el que el servicio REST [6] va a ser desarrollado para servir como método de consulta de los resultados obtenidos por los alumnos.

2.4.8. HTML5 & CSS3

Es un lenguaje web de marcado, junto con CSS3 [11] que corresponde con un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito mediante dicho lenguaje de marcado (HTML5 [10]). Ambos lenguajes son utilizados para el portal de gestión docente para estructurar y dar un toque de estética.

2.4.9. JavaScript

Es un lenguaje de programación interpretado, el cual nos permitirá incrustado en código PHP [3] en algunos ficheros, lanzar alertas respecto a las acciones llevadas a cabo en el formulario de gestión. Ejemplos de mensajes como: “Acción realizada con éxito”.

2.5. Restricciones generales del proyecto

En este apartado se hará un repaso por las distintas restricciones de las que consta el sistema desarrollado. Estas restricciones pueden ser de dos tipos:

- Restricciones relacionadas con la interacción del sistema con el usuario.
- Restricciones relacionadas con la base tecnológica en la que se ejecutará el sistema software.

2.5.1. Herramientas de control de versiones

A la hora de desarrollar cualquier tipo de sistema software, es necesario tener un control exhaustivo sobre las distintas versiones disponibles del sistema a desarrollar.

Es igualmente importante, por otra parte, poder disponer de un sistema que nos permita revertir cualquier cambio introducido en la aplicación que está siendo desarrollada sin suponer esto un impacto considerable en el tiempo del desarrollo.

Teniendo en cuenta estos dos aspectos y el conocimiento previo sobre herramientas enfocadas a realizar este control de versiones y tras su valoración, se consideró necesario el uso de por ejemplo **GitHub** [15].

El uso de este tipo de herramientas nos permite que los cambios realizados por un integrante del proyecto sean propagados de forma instantánea al resto de los actores que intervienen en este, permitiendo ser revisado y editado por éstos.

2.5.2 Herramientas auxiliares

A la hora de construir este sistema software, hemos utilizado varias herramientas que nos han permitido conseguir los siguientes objetivos:

- Reducir tiempos de desarrollo.
- Hacer más cómoda la generación de código y la gestión de dependencias.
- Permitir un conocimiento más extenso del ecosistema al desarrollador o desarrolladores.

Estas herramientas son las siguientes:

- **Visual Studio**: Ha sido clave en los factores anteriores en el aspecto del desarrollo de la aplicación móvil ya que la indentación del código y las autosugerencias, además de

proporcionar las librerías necesarias de manera automática, agilizando mucho el proceso de trabajo [16].

- **Gliffy**: Portal web (go.gliffy.com) que me ha permitido realizar el diseño gráfico de los diagramas de casos de uso y diagramas de secuencia de cara a la memoria.

2.5.3. Calidad del código

En cuanto a lo que se refiere al código, no considero oportuno adjuntar todos los fragmentos que conforman el proyecto, pero sí destacar aspectos como la correcta indentación, uso de código sencillo, evitando que este se encuentre duplicado, comentado en los aspectos relevantes y adherido a estándares de codificación (aspectos de nomenclatura) como es por ejemplo la notación húngara (véase Figura 2.3).

```
37     public Button btnD;
38     string sTema;
39     int iIteraciones; //Almacenar cantidad de preguntas enlazadas a tematica elegida por el usuario.
40     List<string> info;
41     int iIndex = 2;
42     char[] cSplitters = { ';' };
43     public string[] a_datos;
44
45     void Awake(){
46
47         mySound = GetComponent< AudioSource > ();
48         DontDestroyOnLoad (this.gameObject);
49     }
50
51     void Start()
52     {
53         usuario = PlayerPrefs.GetString ("usuario");
54         modo = PlayerPrefs.GetInt ("modo");
55
56         if (modo == 0 || modo == 1) {
57             tipo = 0;
58         }
59
56         if (modo == 2) {
57             tipo = 1;
58         }
59
60         txtTematica.text = PlayerPrefs.GetString ("tematica");
61         txtAciertos.text = "Aciertos:" + aciertos.ToString();
62         SecuenciaJuego ();
63         btnV.gameObject.SetActive (false);
64         btnF.gameObject.SetActive (false);
65         btnA.gameObject.SetActive (false);
66         btnB.gameObject.SetActive (false);
67         btnC.gameObject.SetActive (false);
68         btnD.gameObject.SetActive (false);
69     }
70 }
```

Figura 2.3: Fragmento de código C#

Capítulo 3

Planificación

Podemos descomponer la planificación del proyecto en varias fases. Cabe destacar, sin embargo, que el comienzo de la ejecución de algunas fases se sobreponen con el final de la ejecución de otras.

3.1. Comienzo del proyecto

La primera fase del proyecto consistió en definir claramente cuál era el objetivo del sistema. Para ello, conté con la inestimable ayuda de Guadalupe Ortiz Bellot y Alfonso García de Prado Fontela.

El primer enfoque fue algo distinto a lo que se pensó posteriormente. Descartamos varias tecnologías propuestas en el inicio del proyecto por la escasez de documentación y comunidad.

Uno de los aspectos más importantes destacando el desarrollo y siendo consciente del factor de escalabilidad del sistema, ha sido en cuanto a los vídeos almacenados, que en primera instancia eran integrados de manera directa en la app. Posteriormente con gran esfuerzo se consiguió que dichos vídeos fueran descargados en el servidor evitando así costes de almacenamiento en el teléfono del alumno.

Aprovechando la gran popularidad que está adquiriendo el concepto de 'Realidad aumentada' y, a su vez la versión reciente para reproducción de vídeos en realidad aumentada. La documentación y los recursos de esta tecnología (Vuforia [14]) son muy abundantes y claros, por tanto nos facilitaría enormemente la tarea de desarrollo.

Además, la utilización de esta herramienta nos permitía cargar para cualquier asignatura su vídeo informativo correspondiente basándonos en un único patrón (en este caso imagen de gravilla de Vuforia [14]) lo que era vital para nuestro proyecto al ser de tipo dinámico.

3.2. Análisis de requisitos y recursos

Una vez quedó definida de forma completa la idea del proyecto y la funcionalidad principal de este, la siguiente fase es la fase de análisis de requisitos y recursos.

En esta fase, el objetivo principal fue el de definir claramente *qué debe hacer* nuestra aplicación y *qué recursos debemos utilizar* para construirla.

Por tanto, podemos dividir la fase de análisis en dos sub-fases:

- ◆ Fase de toma y especificación de requisitos (funcionales y no funcionales) de la aplicación.
- ◆ Fase de definición de recursos necesarios para la construcción de la aplicación.

3.3. Aprendizaje y perfeccionamiento

Como hemos explicado previamente, a grandes rasgos el proyecto se divide en tres componentes principales: Front-end (panel de gestión docente), desarrollo de app móvil y Back-end (ficheros en el servidor). Cada uno de estos componentes está formado por un ecosistema de tecnologías y cada una de ellas requiere un tiempo concreto de aprendizaje y perfeccionamiento.

Con respecto al Front-end de la aplicación, las tareas de aprendizaje y perfeccionamiento que se han efectuado han sido las siguientes:

- Actualización y aprendizaje en diseño web: Hasta ahora, mi trabajo había consistido en realizar tareas de Back-end, por tanto mi experiencia con respecto al diseño web era muy baja, por lo que ha sido necesario dedicar algún tiempo a refrescar conocimientos y adquirir nuevos.
- Actualización y aprendizaje en maquetación (HTML5 [10], CSS3 [11]): Al igual que en lo que a diseño web se refiere, mi experiencia con la maquetación web era reducida, por tanto fue necesario dedicar algo de tiempo a adquirir conocimientos.
- Actualización en Javascript: Para presentar los datos de forma clara y vistosa para el usuario final, hemos utilizado código Javascript [5] de cara a las alertas según las acciones realizadas en el panel de gestión.

Con respecto a la aplicación móvil, las tareas de aprendizaje y perfeccionamiento que se han llevado a cabo han sido:

- Actualización y refresco de *Unity* [2], ya que ha sufrido muchísimos cambios a nivel de interfaz respecto a cuando estudié el entorno en la asignatura Desarrollo de Sistemas Hipermédia.
- Aprendizaje y perfeccionamiento de la extensión Vuforia [14] para el uso del concepto de realidad aumentada.
- Aprendizaje y perfeccionamiento del lenguaje C# [1] ya que es el utilizado en todos los ficheros de código en *Unity* [2].
- Aprendizaje y perfeccionamiento de métodos para comunicarme con la base de datos debido al carente soporte de conexión MySQL [9] proporcionado por *Unity* [2].
- Aprendizaje y profundización en el concepto de uso de co-rutinas en C# [1].

Con respecto al Back-end de la aplicación, las tareas de aprendizaje y perfeccionamiento que se han efectuado han sido las siguientes:

- Aprendizaje de y refresco de servicios REST [6], utilizado en la parte servidora.
- Aprendizaje de comandos y comunicación con la máquina servidora para transferir los ficheros de código/scripts PHP [3] desarrollados en mi máquina a la del servidor.
- Aprendizaje de método para descarga automática de vídeos en el servidor en la adición de nuevas asignaturas. Esto ha conllevado una gran cantidad de tiempo y ha sido realizado mediante el programa de línea de comandos: “youtube-dl” [21].
- Refresco y puesta en práctica de uso de base de datos estructurada, haciendo uso de tablas y sus correspondientes relaciones (alojada en el servidor).
- Aprendizaje del funcionamiento del sistema de directorios LDAP [20] (Protocolo ligero de acceso a directorios) de la universidad para aprender a autenticar a los alumnos haciendo uso de él en la app.

3.4. Desarrollo

No existe un límite claro entre la etapa de aprendizaje y la etapa de desarrollo, ya que, a medida que se ha ido analizando documentación, se han ido realizando pruebas de concepto funcionales que, en ocasiones, han acabado formando parte del sistema software desarrollado.

Esta etapa se trata de la etapa más intensa en cuanto a trabajo y también más larga con respecto al tiempo.

Con el objetivo en mente de hacer más liviana la implementación, se decidió implementar, por separado aunque desarrollados en paralelo por la dependencia entre ellos, cada uno de los tres aspectos comentados previamente que aparecen en nuestro sistema:

- ◆ Aplicación móvil.
- ◆ Elementos del front-end (Panel de gestión).
- ◆ Elementos en el back-end (Parte servidora).

Una vez resuelta la implementación y probados cada uno de los módulos procedimos a integrar todos los módulos y realizar pruebas de integración.

3.5. Testing

La fase de pruebas, como hemos podido estudiar a lo largo del grado, es una de las fases más importantes en el desarrollo de un proyecto software.

Esta fase se ha realizado de forma paralela a la fase de desarrollo, ya que la idea principal era la de ir ejecutando el plan de pruebas con respecto se iban implementando los distintos componentes.

Con esto conseguimos que a medida que incorporamos mejoras y avances en el sistema lo que funcionaba previamente y mantenía relación entre sí, tengamos la seguridad de que siga correcto. De lo contrario si finalizamos por completo un elemento del sistema, si nos desemboca en errores, llegará a ser de mayor envergadura localizar el error y más aun si existe relación con otro elemento del ecosistema general.

3.6. Correcciones

Además de realizar las tareas descritas en el plan de pruebas del proyecto, hemos reservado tiempo para las posibles correcciones que pudieran surgir tras la valoración de la tutora del proyecto.

Además de las correcciones del código en sí de la aplicación, se tendrán en cuenta las posibles modificaciones que haya que realizar en la presente memoria.

3.7. Memoria

El procedimiento a la hora de redactar la memoria ha sido el clásico: Se ha ido creando con respecto iba avanzando la implementación, de forma paralela.

Sin embargo, una vez completado el desarrollo, se ha dedicado tiempo a revisar la documentación en busca de posibles incoherencias con la versión final de la aplicación.

3.8. Diagrama de Gantt

En la siguiente figura (3.1) se representa la planificación desglosada por tareas recogidas en un diagrama de Gantt. Se incluye en este, además del aprendizaje y el desarrollo, el tiempo dedicado a testing.

3.9. Modelo de ciclo de vida

Durante el desarrollo de este proyecto se ha seguido un modelo de desarrollo iterativo e incremental.

En un desarrollo de este tipo, el modelo de ciclo de vida que más se ajusta es el iterativo e incremental. En este tipo de desarrollo, el proyecto se planifica en distintos bloques temporales llamados iteraciones.

Estas iteraciones se pueden entender como proyectos a escala reducida: En todas ellas se repite un proceso similar para proporcionar un producto. Con las iteraciones podemos resolver algunos problemas que surgen en el modelo de ciclo de vida en cascada, en el que no se continuarán con las fases del proceso de desarrollo si los requisitos en una fase no han sido satisfechos.

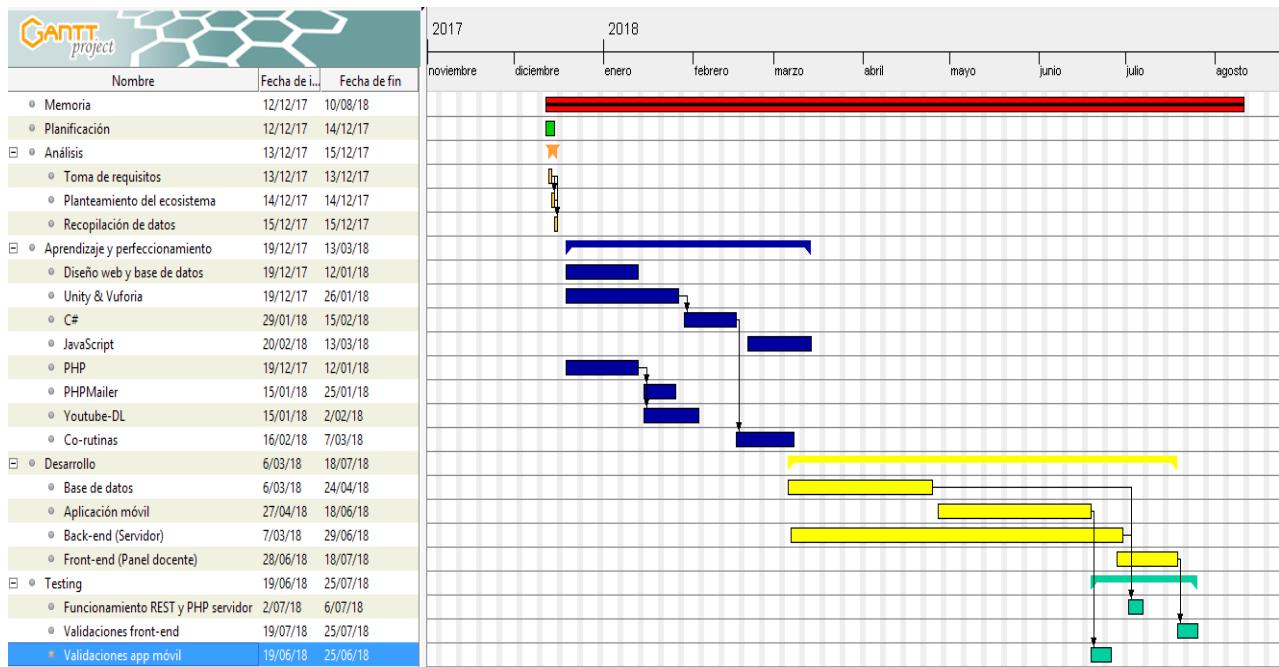


Figura 3.1: Diagrama de Gantt

Las necesidades de este proyecto hacen que debamos adoptar un modelo de ciclo de vida iterativo e incremental, ya que es el que mejor encaja en el modelo de trabajo que seguiremos, en el que se producirán reuniones periódicas (al menos, una vez por semana) vía correo electrónico y reuniones periódicas cara a cara, en la que se presentarán las funcionalidades desarrolladas para la herramienta en forma de un producto totalmente funcional.

Capítulo 4

Análisis

4.1. Especificación de requisitos del sistema

Antes de proceder al desarrollo del sistema, es necesario que hagamos una descripción completa del comportamiento deseado del sistema que se va a desarrollar, así como la necesidad que ha de satisfacerse. Esta descripción puede ser definida como la **especificación de requisitos de software**. Procederemos a definir esta especificación de requisitos de una forma estructurada y lógica.

4.1.1. Requisitos de interfaces externas

En esta sección se describirán los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.

Requisitos que afectan a la interfaz del usuario docente

La interacción del usuario docente con el sistema se hará a través de la interfaz web de su ordenador, por tanto definir los requisitos de interfaz significa definir los requisitos de cada uno de los componentes que forman el front-end de este.

Los diferentes componentes que permitirán al usuario docente interactuar de forma explícita con el sistema serán los siguientes:

Panel de registro/inicio de sesión

Con respecto a dicha interfaz, el docente procede a realizar su registro si es la primera vez, mientras que el administrador lo dará de alta. En dicha interfaz también realizará el inicio de sesión una vez esté dado de alta e incluso tiene la capacidad de solicitar restablecimiento de contraseña si la olvida.

Panel de gestión docente

Respecto a ello, permitirá al profesor realizar determinadas acciones tales como indican las siglas CRUD de docente. Crear asignaturas (enlazando su vídeo), activar y desactivar un examen, cargar preguntas de asignaturas para borrarlas o editarlas, consultar calificaciones e incluso insertar preguntas Verdadero/Falso y preguntas de múltiple respuesta.

Panel de gestión administrativo

Permitirá al administrador del sistema únicamente, la funcionalidad de enlazar/desvincular a un docente de una asignatura.

Requisitos que afectan a la interfaz del usuario alumno

La interacción del usuario alumno con la aplicación móvil se hará a través de la interfaz de su teléfono (smartphone Android [13]), por tanto definir los requisitos de interfaz significa definir los requisitos de cada uno de los componentes que forman el front-end de este.

Los diferentes componentes que permitirán al usuario alumno interactuar de forma explícita con el sistema serán los siguientes:

- Escena de registro/inicio, donde el alumno ingresa su identificador (ej: “u45323398”).
- Escena de elección de modo, donde a través de un desplegable el alumno seleccionará.
- Escena de elección de asignatura, donde a través de un desplegable el alumno seleccionará.
- Escena de realidad aumentada, donde el alumno posee controles en pantalla de play/pause.
- Cuestionario donde contestará las preguntas consecuentes a la asignatura elegida.

Requisitos de interfaz con otros sistemas

Parte del objetivo principal del sistema es el de poder servir como herramienta software de sistemas más complejos. Es por esto por lo que, además de interactuar con el usuario de forma directa, el sistema debe ser capaz de interactuar con otros sistemas externos. Estos sistemas externos pueden clasificarse en algunos tipos:

Aplicación móvil con elemento servidor:

El primer requisito dentro de los requisitos de interfaz con otros sistemas es que el sistema debe ser capaz de obtener información sobre las asignaturas existentes, preguntas sobre dicha asignatura y capacidad para almacenar datos sobre el alumno en el transcurso de su paso por la app, con el objetivo de permitir a otros elementos del sistema software nutrirse de esta información.

Front-end con back-end (servidor)

Dado la necesidad por parte del docente de conocer la situación de los datos de las asignaturas, debe ser capaz de recopilar información almacenada en la base de datos como por ejemplo datos sobre alumnos (calificaciones) y preguntas correspondientes a las diferentes asignaturas a las que dicho profesor se encuentre asignado (p.e).

Requisitos de comunicación

La base del funcionamiento de este sistema software es la de manejar datos de alumnos y asignaturas, procesarlos y proporcionarlos de forma clara tanto al usuario docente como al usuario alumno en su uso en la aplicación móvil Android [13], por tanto, el sistema necesita de una comunicación clara y fiable con las bases de datos.

La comunicación con este sistema se hará a través de:

- Respecto a consultar calificaciones por parte del docente y realizar operaciones CRUD en su panel (crear, leer, actualizar y eliminar), se realiza mediante la librería “mysqli” de PHP [3].
- Respecto a obtener los datos para el funcionamiento de la aplicación móvil y su almacenamiento, se realiza mediante la librería WWW y WWWForm de C# [1] junto con el uso de co-rutinas, en su comunicación con ficheros PHP [3] en el servidor mediante \$_POST.

4.1.2. Requisitos funcionales

A continuación procederemos a definir los requisitos funcionales del sistema, es decir, las funciones propiamente dichas que el sistema debe cumplir para asegurar su correcto funcionamiento.

Los requisitos funcionales que el sistema debe cumplir son los siguientes:

- El sistema debe estar comunicado con el servicio REST [6] de forma estable.
- Previo al manejo como usuario docente o usuario alumno, debemos encontrarnos conectados a la red UCA (véase manual de usuario con FortiClient VPN [12]).
- La aplicación móvil proyectará el vídeo de la asignatura en cuestión si reconoce adecuadamente la imagen (patrón) indicado a usar para el factor de realidad aumentada (imagen “gravilla”).
- El sistema debe asegurar que la información mostrada al usuario docente en su consulta sobre los alumnos es coherente con la información almacenada en la base de datos.
- La aplicación móvil deberá asegurar que si un alumno no posee preguntas a repasar (ha completado dicha asignatura sin fallos o repasado correctamente), mostrará una escena que hará regresar al alumno a la elección de modo y asignatura.
- La aplicación móvil mostrará al alumno que la partida ha finalizado con una escena de “game over” indicándole su puntuación final y permitirá volver a jugar de nuevo.
- El sistema permitirá la interacción con una interfaz cómoda con uso de desplegables, permitiendo una experiencia de usuario más sencilla y rápida tanto para administrador como para los docentes.

4.1.3. Requisitos de rendimiento

En este tipo de aplicaciones móviles (dentro del sistema) quizás sea el punto a enfocar en el aspecto de rendimiento. Aun así la necesidad de una cantidad de memoria RAM en nuestro smartphone no será crucial, pero sí clama un mínimo para la fluidez en la carga de escenas y elementos correspondientes a la escena de realidad aumentada, así como la sincronía entre video y audio para dicha escena.

En smartphones cuya cantidad de memoria RAM sea muy inferior a 2GB su rendimiento afectará levemente de todos modos provocando quizás ciertos picos de colapso de imágenes por segundo.

A su vez la conexión Wi-Fi del dispositivo si puede ser un factor influyente a la hora de mostrar videos sobre las asignaturas, ya que estos son leídos directamente del servidor y no tendremos problema o no deberíamos tener si nos encontramos bajo la red Wi-Fi en la universidad.

Si nos encontramos en casa (siempre bajo conexión a la VPN de la UCA) y dicha calidad de conexión a Internet no es suficiente puede provocar demora en la carga de dichos videos en la escena de realidad aumentada. Aun así se encuentra preparado para no lanzar el video hasta asegurar un buffer decente del video.

4.1.4. Restricciones de diseño

La capacidad del sistema de mostrar los datos con rapidez y, por tanto, minimizar el tiempo de procesamiento de estos datos debe primar sobre cualquier otro aspecto en este sistema software. Esta es, por tanto, la principal restricción de diseño que tendrá el sistema.

4.1.5. Requisitos de sistema software

El sistema deberá cumplir los siguientes requisitos:

- La aplicación móvil deberá poderse visualizarse de forma correcta en Android 4.4.+.
- La arquitectura del sistema debe ser similar a la descrita en los documentos de buenas prácticas de las tecnologías utilizadas.
- La aplicación móvil escalará según el dispositivo que utilicemos, adaptándose sus elementos a las dimensiones del smartphone donde se inicie dicha app al igual que la interfaz web para el docente la cual será de tipo **responsive**.

- El código debe ser limpio, bien estructurado y documentado, con el objetivo principal de ser mantenable y escalable.

4.2. Modelo de casos de uso

Para describir de forma correcta el comportamiento del sistema, usaremos el lenguaje de modelado de sistemas UML [17]. Este lenguaje nos permitirá definir de forma clara los requisitos del sistema, centrándonos en la funcionalidad del sistema y no en cómo conseguir esa funcionalidad.

4.2.1. Diagramas de casos de uso

En esta primera sección mostraremos el modelo de casos de uso que representa la funcionalidad completa del sistema software. Para ello, se ha usado el siguiente esquema:

- Identificar los actores, es decir, las entidades que puedan interactuar con el sistema.
- Para cada una de estos actores, definir de forma clara sus formas de interactuar con el sistema.
- Construcción de los casos de uso
- Estructurar los casos de uso

El modelo de casos de uso es el siguiente (4.1):

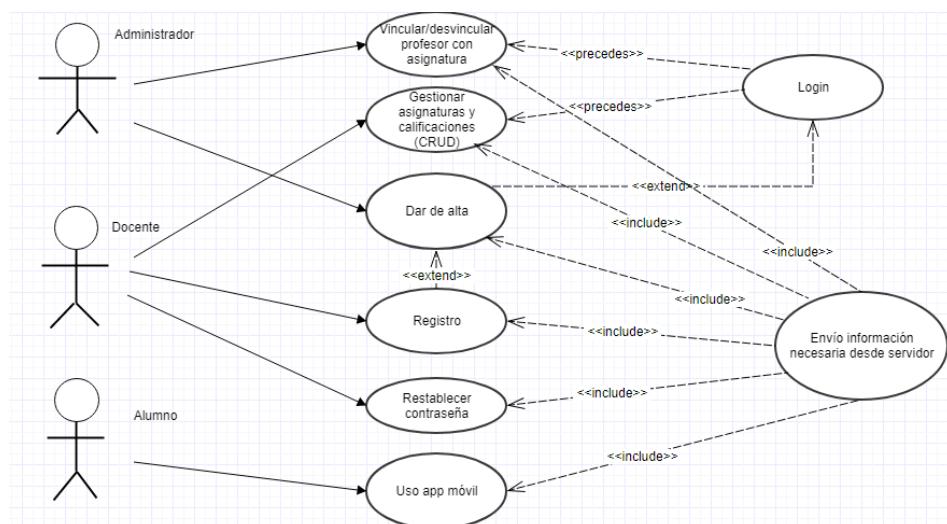


Figura 4.1: Modelo de casos de uso del sistema

4.2.2. Descripción de los casos de uso

En la presente sección nos dedicaremos a describir todos y cada uno de los casos de uso que se muestran en el diagrama anterior. Para la descripción usaremos una notación formal usando plan-tillas, ya que la descripción debe ser legible y comprensible para un usuario no experto.

Vincular/desvincular profesor y asignatura

Actores: En este caso de uso participa el actor Administrador.

Precondiciones: El administrador quiere desvincular o vincular una asignatura de un profesor en concreto.

Postcondiciones: Tras iniciar sesión y comprobar que es el administrador del sistema dicha persona, pasará a seleccionar el profesor y la asignatura, haciendo clic posteriormente en el botón vincular o desvincular y aparecerá una alerta en la interfaz web indicando lo ocurrido.

Escenario principal:

1. El administrador accede al portal web e inicia sesión con sus credenciales.
2. El sistema lo redirige a una página para gestionar la relación asignatura/profesor.
3. Cuando el administrador hace clic en el botón vincular o desvincular, se realiza la acción sobre la base de datos y el servidor responde notificando mediante una alerta en pantalla si se ha realizado con éxito o no ha sido posible.

Escenario alternativo: No existe camino alternativo en este caso de uso.

Extensiones: No existen extensiones en este caso de uso.

Dar de alta un docente

Actores: El encargado de llevar a cabo la tarea es administrador.

Precondiciones: El administrador necesita dar de alta un docente que se ha registrado recientemente.

Postcondiciones: El sistema permitirá el acceso al docente tras este haber sido dado de alta por el administrador.

Escenario principal:

1. El administrador accede a su bandeja de correo electrónico (gmail).
2. Accede al enlace que el correo le indica para dar de alta al profesor correspondiente.
3. El servidor notificará en pantalla le notifica que la cuenta de dicho profesor ha sido activada con éxito.

Escenario alternativo: No existe camino alternativo en este caso de uso.

Extensiones: No existen extensiones en este caso de uso.

Registro en el sistema

Actores: El usuario docente.

Precondiciones: El docente necesita registrarse en el sistema.

Postcondiciones: El sistema muestra en pantalla que la cuenta ha sido registrada con éxito y está a la espera de ser dada de alta por el administrador para poder iniciar sesión.

Escenario principal:

1. El usuario docente accede al portal web.
2. El docente rellena los campos nombre, apellidos, correo y contraseña.
3. El docente se registra y en base a la respuesta del servidor; si se ha realizado con éxito aparecerá en pantalla que está a la espera de ser dado de alta.

Escenario alternativo: No existe camino alternativo en este caso de uso.

Extensiones: No existen extensiones en este caso de uso.

Restablecimiento de contraseña

Actores: El usuario docente.

Precondiciones: El usuario docente ha olvidado su contraseña y necesita restablecerla.

Postcondiciones: El sistema envía un correo al docente para que este acceda e introduzca su nueva contraseña y quede satisfecho.

Escenario principal:

1. El usuario docente accede al portal web.
2. El docente hace clic en el apartado “Olvidaste la contraseña”.
3. El servidor envía un correo con un enlace al docente, el cual accede a dicha dirección, introduce su nueva contraseña y responde el servidor apareciendo en pantalla un mensaje de haber restablecido con éxito.

Escenario alternativo: No existe camino alternativo en este caso de uso.

Extensiones: No existen extensiones en este caso de uso.

Gestionar asignaturas y calificaciones

Actores: El usuario docente.

Precondiciones: El usuario docente necesita realizar algún tipo de gestión.

Postcondiciones: El sistema lleva a cabo la acción solicitada por el docente y notifica en pantalla con una alerta el resultado de haber ejecutado dicha operación (éxito o fallo).

Escenario principal:

1. El usuario docente accede al portal web.
2. El docente elige la operación a realizar en base a sus asignaturas a las que está vinculado:
 - Añadir temática → escribe la nueva asignatura e introduce la URL de su vídeo y hace clic en insertar.
 - Activar/desactivar examen → elige del desplegable la asignatura y hace clic en activar o desactivar.
 - Cargar preguntas de asignatura → elige la asignatura y carga las preguntas. Se le permite editar los títulos de las preguntas o borrarlas directamente.
 - Consultar calificaciones de alumnos → elige una fecha y una asignatura.
 - Añadir preguntas tipo V/F → elige asignatura, introduce título de la pregunta, indica si es V ó F y escribe el tipo de pregunta.
 - Añadir preguntas de múltiples respuestas → elige asignatura, introduce título de la pregunta, introduce la respuesta correcta y las tres respuestas incorrectas y finalmente escribe el tipo de pregunta.
3. El docente hace clic en el botón correspondiente tras haber llenado los campos solicitados para la operación y responde el servidor mostrando en pantalla una tabla con resultados si corresponde a ciertas operaciones o simplemente muestra un mensaje en pantalla indicando éxito o fallo.

Escenario alternativo: No existe camino alternativo en este caso de uso.

Extensiones: No existen extensiones en este caso de uso.

Uso de aplicación móvil

Actores: El usuario alumno.

Precondiciones: El usuario alumno necesita ser evaluado o practicar de cara a una evaluación.

Postcondiciones: Se reproduce una secuencia en el recorrido del juego que indicará al alumno su resultado final.

Escenario principal:

1. El usuario alumno accede a la aplicación móvil (garantizando que se encuentra bajo la red de la universidad) y autentica con LDAP [20].
2. El alumno procede a comenzar el juego y sucede la siguiente secuencia:
 - Escena portada, el alumno hace clic en “Comenzar”.
 - Escena inicio cargada, el alumno introduce su identificador (si no lo hace como se le indica, sino se le redirige a la misma escena hasta hacerlo con éxito).
 - Tras identificarse, carga la escena de elección de modo (Practicar todas las preguntas, repasar las falladas, hacer el examen).
 - Se inicia la escena de elección de asignatura de un desplegable.
 - Tras elegir temática, carga la escena de realidad aumentada donde al reconocer la imagen de la gravilla previamente mencionada, se proyectará el vídeo vinculado a dicha asignatura con controles para pausar o reproducir en pantalla y cuando hayamos terminado de verlo, hacemos clic en “Comenzar cuestionario”.
 - Se suceden una serie de preguntas según corresponda y en pantalla veremos nuestra puntuación.
3. El docente hace clic en el botón correspondiente tras haber llenado los campos solicitados para la operación y responde el servidor mostrando en pantalla una tabla con resultados si corresponde a ciertas operaciones o simplemente muestra un mensaje en pantalla indicando éxito o fallo.

Escenario alternativo:

Si elegimos realizar el examen sin estar activo, nos redirigirá al inicio de la aplicación.

Si elegimos repasar teniendo correctas todas las preguntas de una temática, nos redirigirá al inicio de la aplicación.

Extensiones: No existen extensiones en este caso de uso.

4.3. Modelo de comportamiento del sistema

El modelo de comportamiento especifica como debe actuar el sistema. En este modelo podemos diferenciar dos partes:

- **Diagramas de secuencias del sistema:** muestran la secuencia de eventos entre el usuario y el sistema.
- **Contrato de las operaciones del sistema:** describen el efecto que producen las operaciones en el sistema.

4.3.1. Diagramas de secuencia

A continuación se muestran unos sencillos diagramas de secuencia correspondientes con cada caso de uso explicado anteriormente:

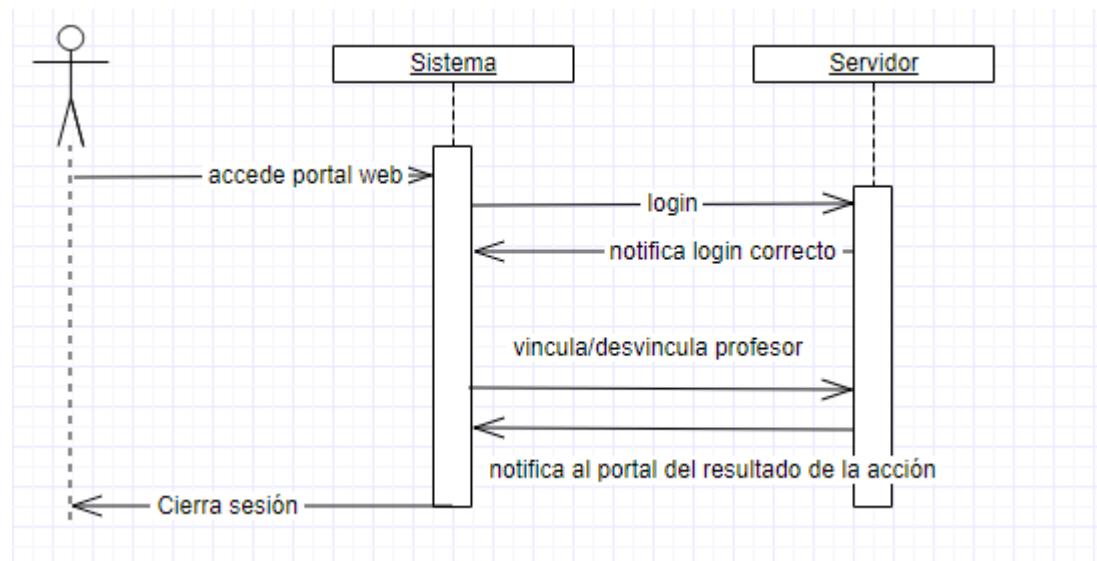


Figura 4.2: Vincular/desvincular profesor de asignatura

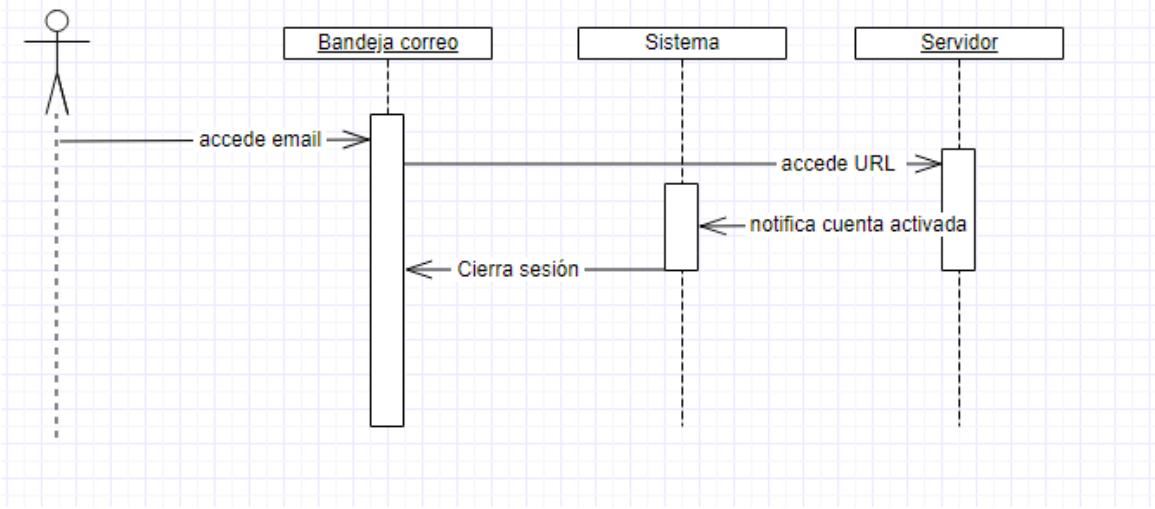


Figura 4.3: Dar de alta un docente

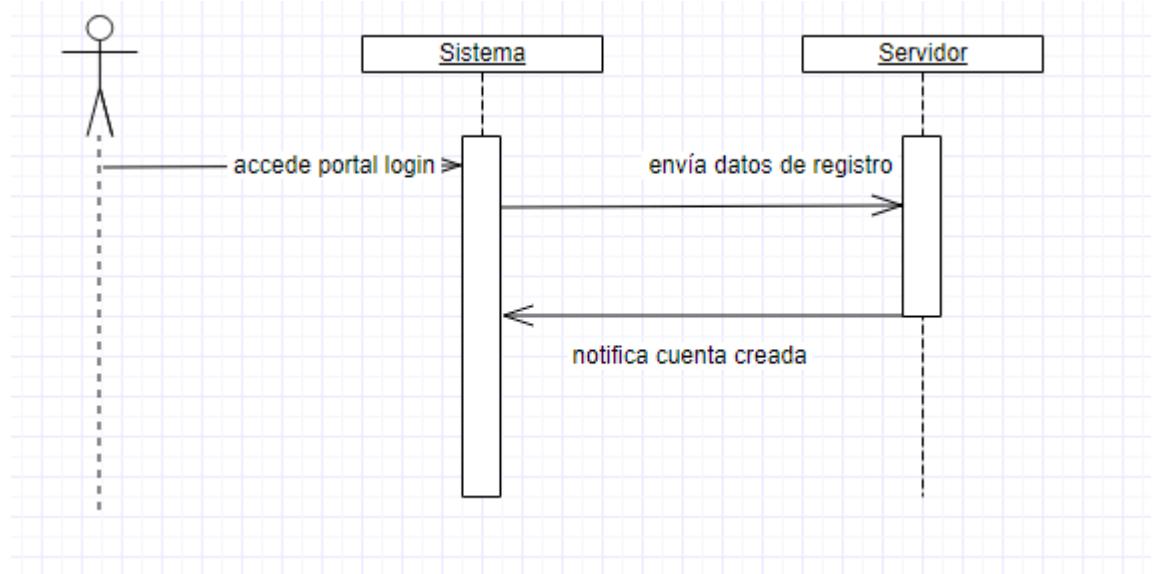


Figura 4.4: Registro de docente

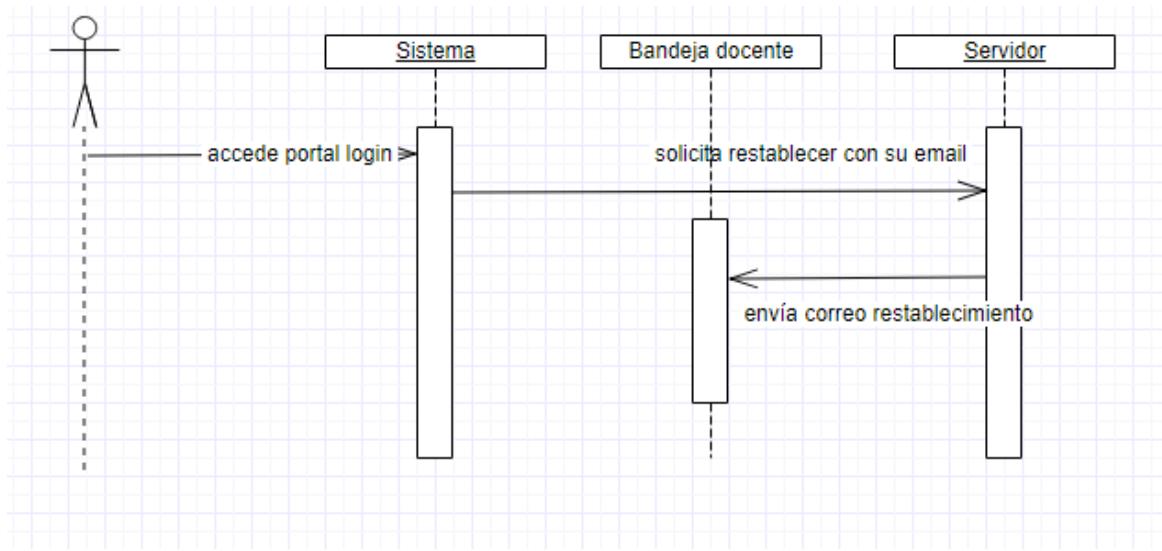


Figura 4.5: Restablecimiento de contraseña

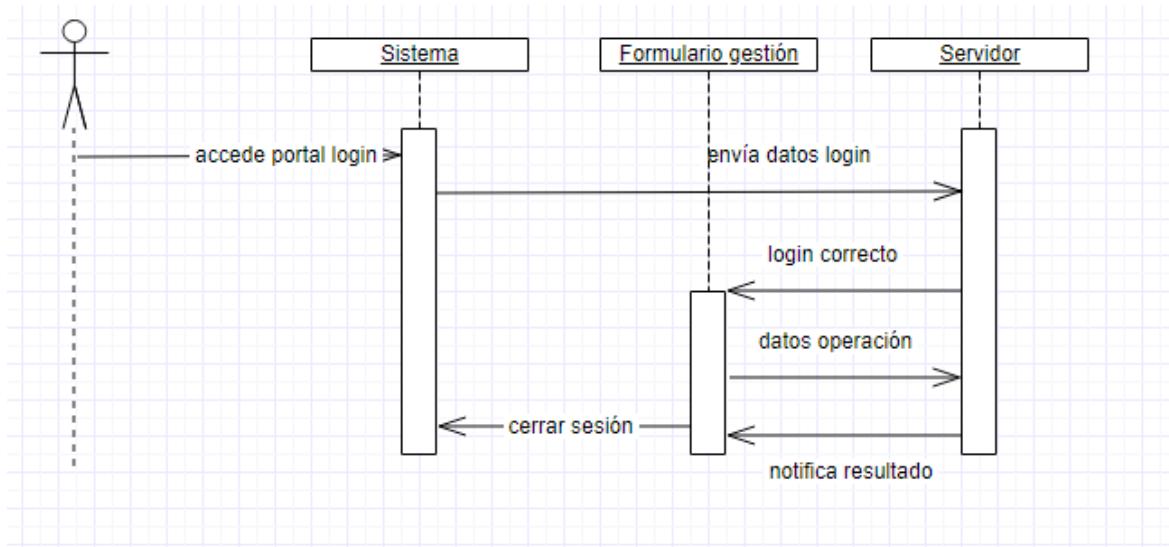


Figura 4.6: Gestionar asignaturas y calificaciones

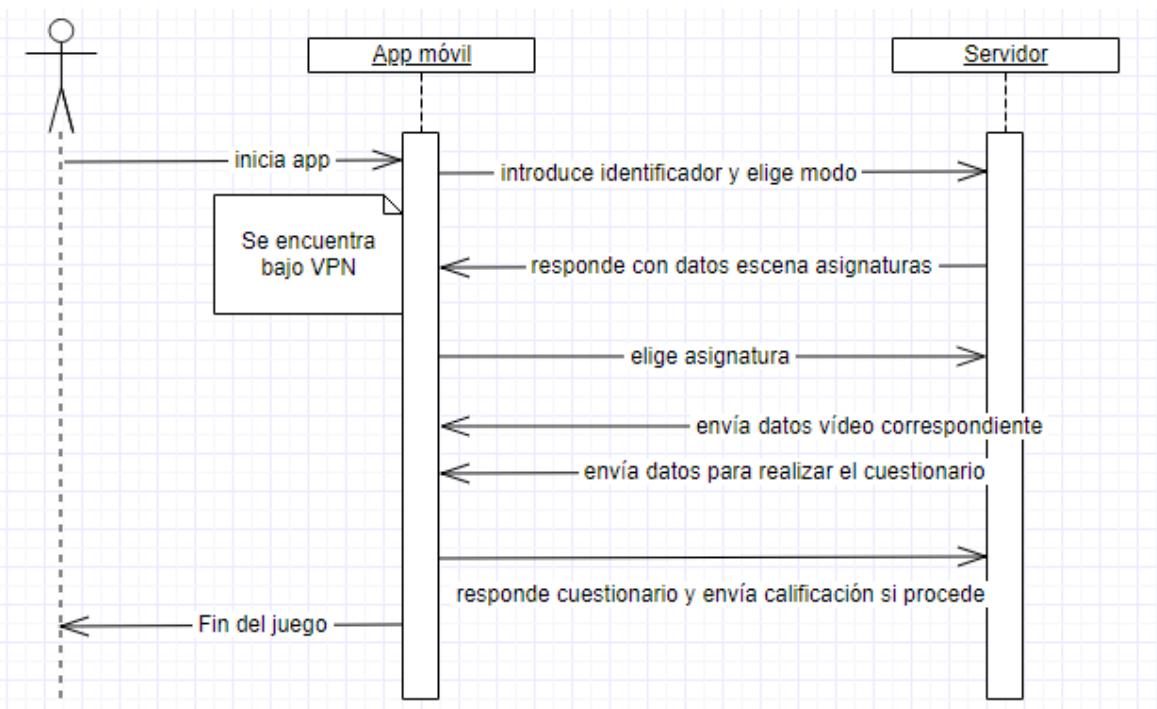


Figura 4.7: Secuencia de juego app móvil

Capítulo 5

Diseño

Una vez hemos descrito la funcionalidad completa del sistema, podemos describir cual será el diseño completo de este.

Cabe destacar el diseño que tiene la base de datos y será mostrado en primer lugar.

El diseño de la interfaz gráfica del sistema se ha realizado de forma que la información se presente al usuario de forma dinámica y sencilla, haciendo transparente para este todo el proceso de captura y procesamiento de los datos.

5.1. Esquema de la base de datos

A continuación se muestra un esquema de la estructura de la base de datos utilizada (Figura 5.1):

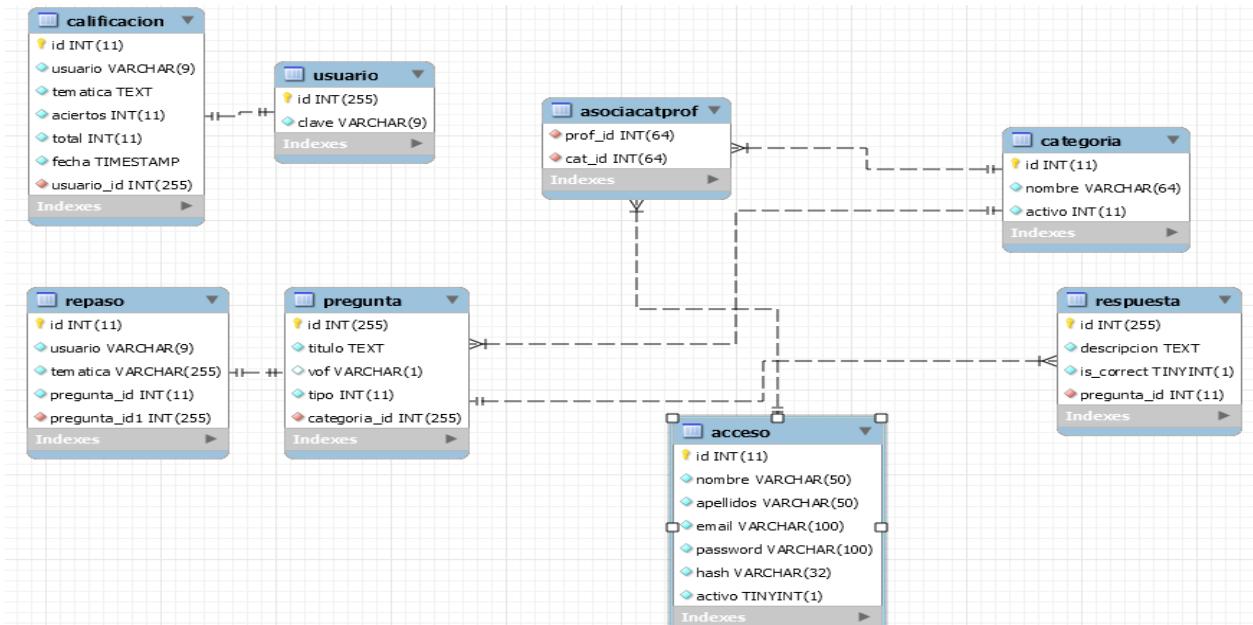


Figura 5.1: Esquema base de datos relacional

5.2. Interfaz gráfica del portal web

Tras los resultados que se han obtenido en las fases de análisis del sistema, es necesario desarrollar una interfaz sencilla y agradable para el usuario de la misma. En el diseño de estos, se intentará en todo momento que el flujo de ejecución sea lo más intuitivo posible, ya que pretendemos realizar simples procesos de gestión.

El objetivo principal de la interfaz gráfica del sistema es el de mostrar al usuario final de forma clara y sencilla el resultado de la lectura y procesamiento de los datos de forma correcta.

Para ello se establece un mínimo de seguridad a través del portal de login/registro previo, para evitar accesos no autorizados o indebidos, como ya hemos comentados en secciones previas.

Como hemos comentado en capítulos anteriores de la presente documentación, la interfaz gráfica presenta al usuario información actualizada y coherente.

A continuación, mostraremos capturas de pantalla (Figuras 5.2, 5.3, y 5.4) de las partes diseñadas tanto para administrador (5.2) como para docente (5.3) y presentar así la información, sin olvidarnos como parte importante el panel de registro/inicio de sesión (5.4):



Figura 5.2: Pantalla de gestión de administrador

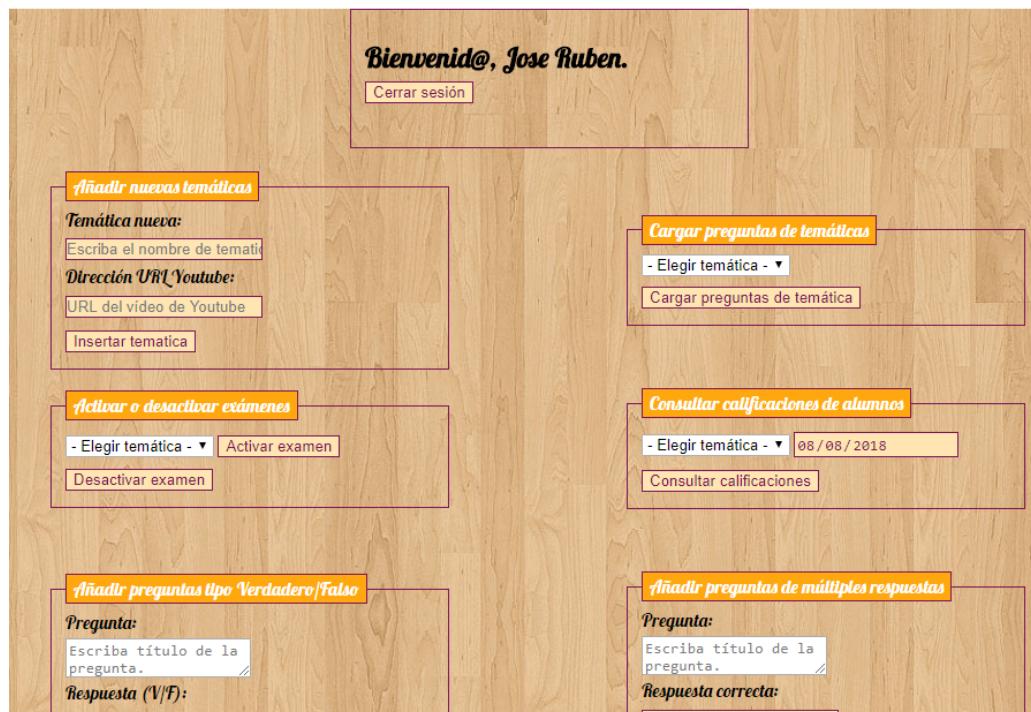


Figura 5.3: Panel de gestión del docente

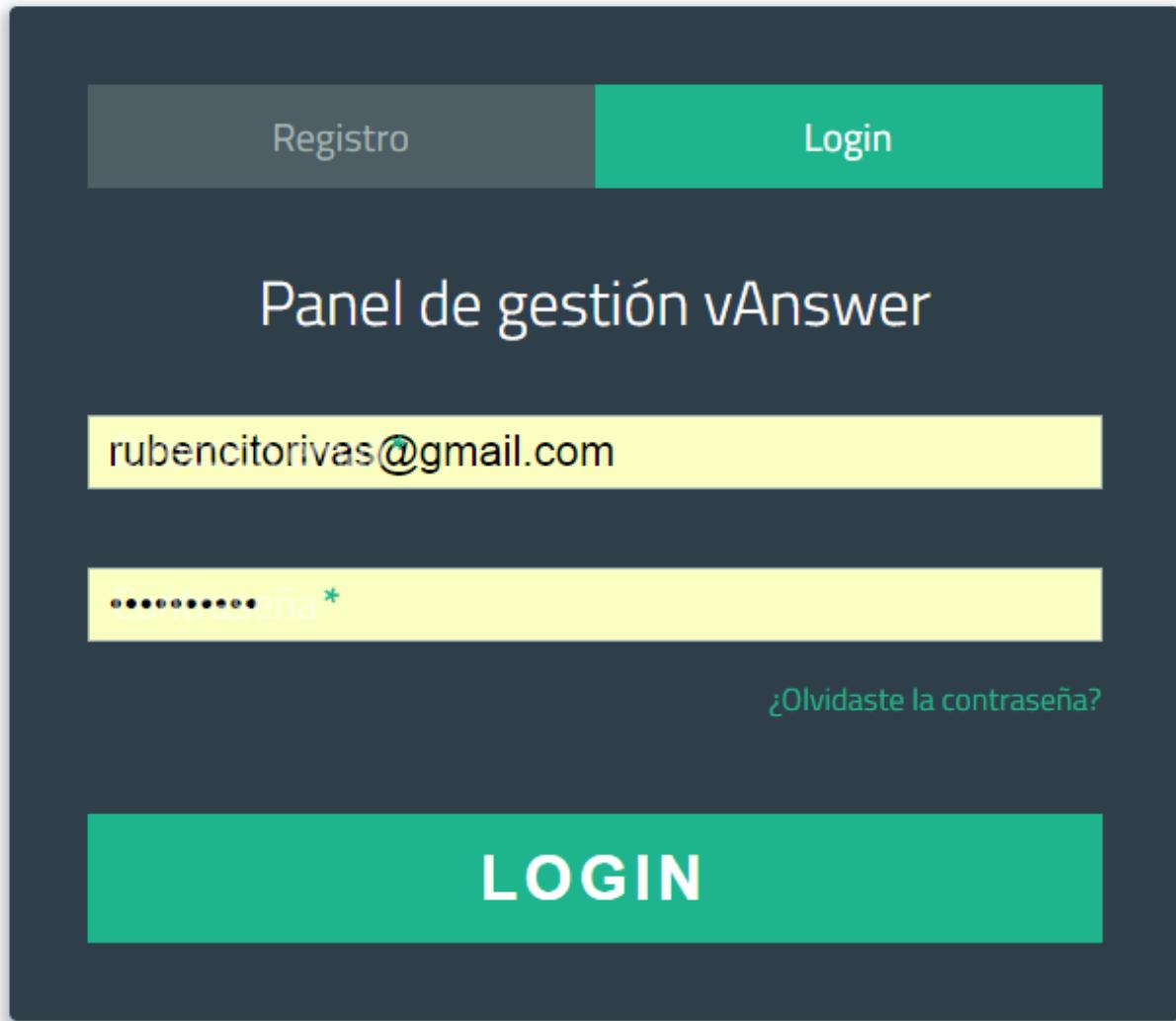


Figura 5.4: Panel de registro y login

5.3. Interfaz de la aplicación móvil

Una vez hemos observado la interfaz sencilla sobre la que hemos trabajado para desarrollar la parte del front-end del docente y administrador, procederemos a mostrar una serie de capturas que hagan comprender en profundidad y de manera gráfica las diversas escenas por las que el alumno pasa durante su uso de la aplicación móvil.

A continuación se exponen las diferentes escenas a modo de imágenes (Figuras 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12 y 5.13).

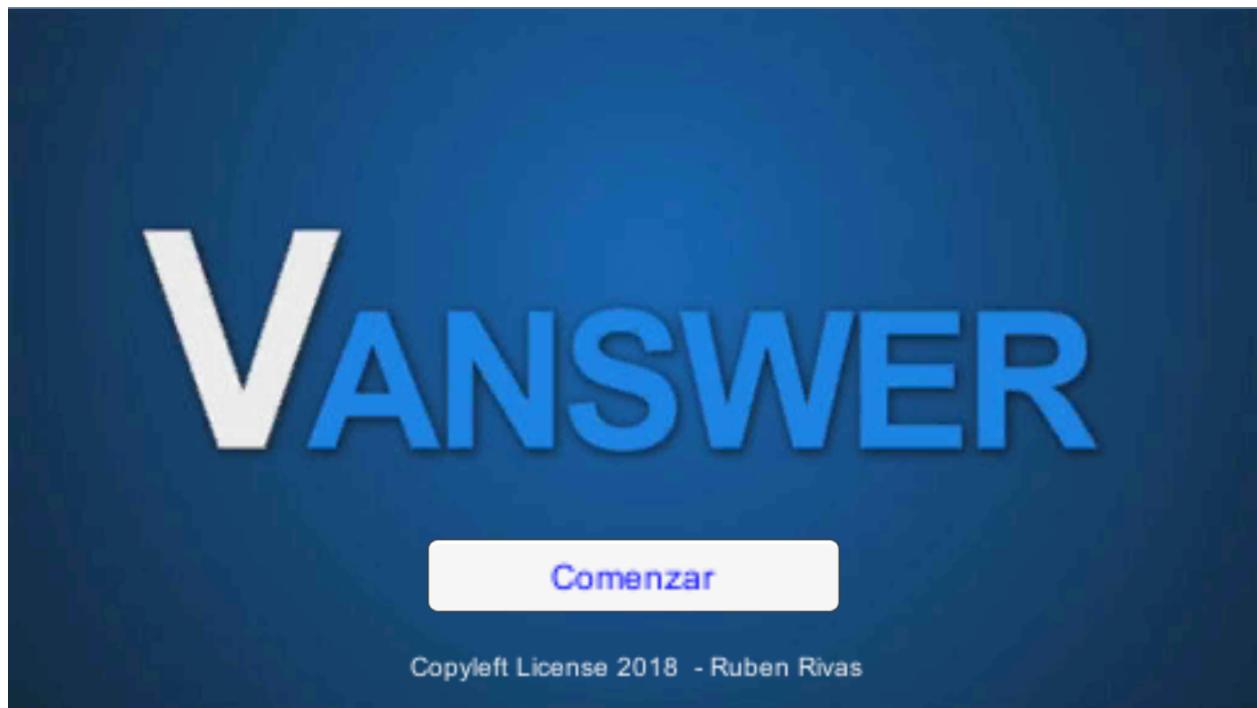


Figura 5.5: Escena de la portada



Figura 5.6: Escena de inicio



Figura 5.7: Escena de elección de modo



Figura 5.8: Escena de elección de asignatura

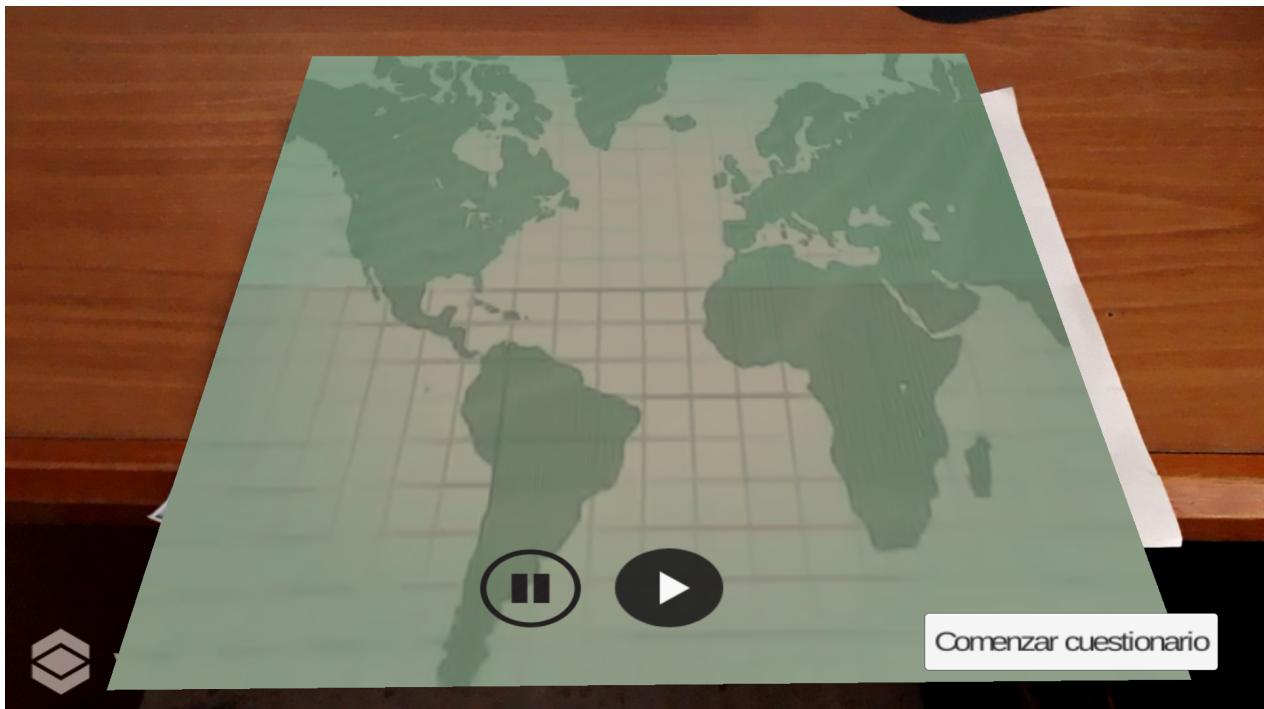


Figura 5.9: Escena de realidad aumentada con controles

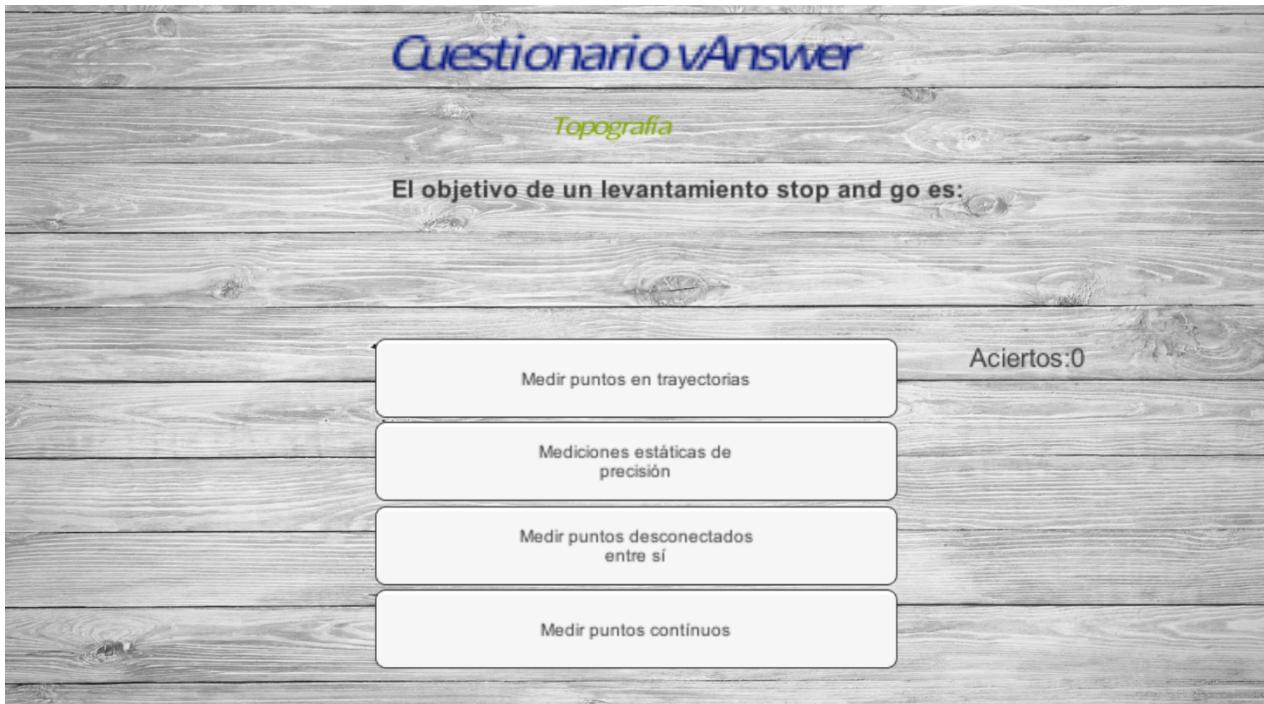


Figura 5.10: Escena del cuestionario

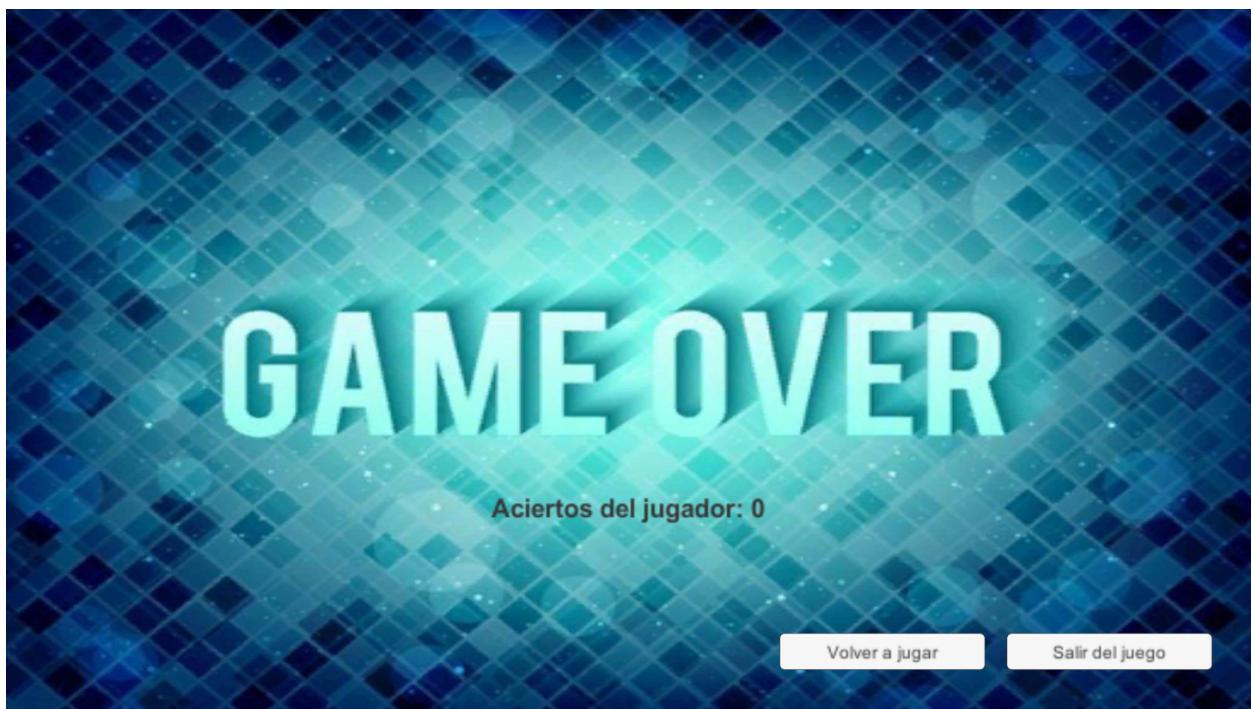


Figura 5.11: Escena fin del juego

☆ EXAMEN NO DISPONIBLE ☆

**ESPERE A QUE SU DOCENTE
ACTIVE EL EXAMEN
O HÁGASELO SABER.**

[Volver a elegir modo y temática](#)

Figura 5.12: Escena examen no disponible



Figura 5.13: Escena de repaso completado

Capítulo 6

Implementación

6.1. Arquitectura general del sistema

Teniendo en cuenta lo anterior, a continuación se muestra el esquema general de la arquitectura de la aplicación. Los siguientes apartados, explican la función de cada uno de los componentes que se usan para su implementación (ver Figura 6.1).

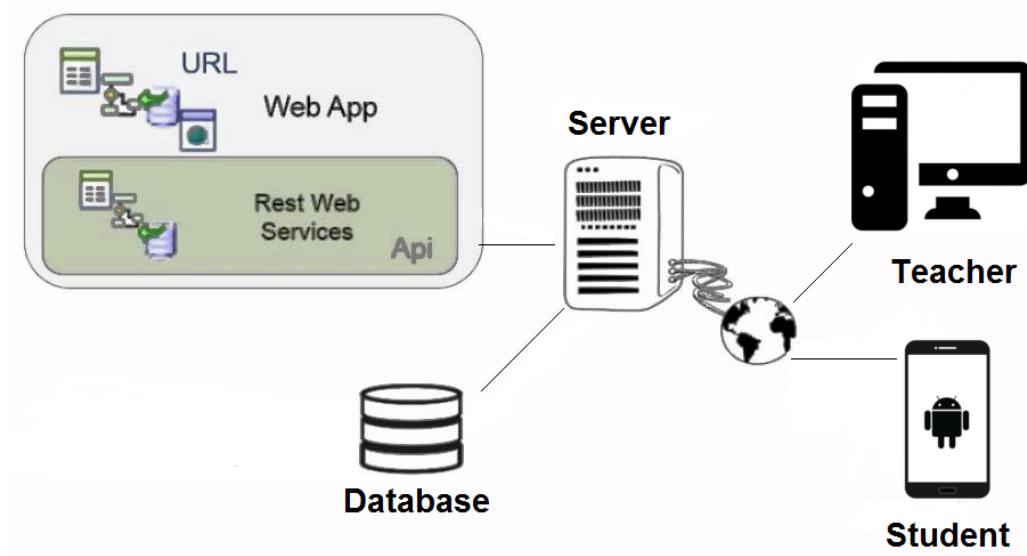


Figura 6.1: Aspecto de la arquitectura del sistema.

Aplicación móvil

- Ficheros de código C# (Controladores de la aplicación) [1].
- Componente realidad aumentada.
- Escenas y sonidos.

Panel de gestión administrador/docente

- Ficheros PHP [3], HTML5 [10] y hoja de estilos CSS3 [11].

Back-end (parte servidor)

- Servicio REST [6].
- Ficheros PHP [3] para la comunicación con aplicación móvil.
- Directorio para almacenamiento de vídeos de las asignaturas.
- Ficheros PHP para comunicación con panel de registro/inicio de sesión.

Cada uno de los componentes que componen estos tipos tiene una función clara y concisa, intentando modularizar lo máximo posible con vistas al trabajo futuro.

Aplicación móvil

6.1.1. Controladores (ficheros de código C#)

En este apartado haremos mención a los diferentes ficheros de código implementados para servir como controladores de las diferentes situaciones en el transcurso de la aplicación.

Se han desarrollado los siguientes controladores en *Unity* [2]:

- IdController → Manejo de ID introducido por el alumno.
- DropController → Enriquecimiento de desplegable con información necesaria.
- CargaEscenaController → Para cargar escenas correspondientes según el transcurso de la app.
- GameController → Para controlar los aciertos, la batería de preguntas a proyectar y almacenar ajustes del jugador.
- ManejaDropdownController → Auxiliar para “DropController” controlando cuando hay selección de un valor específico.
- VideoController → Encargado de funcionar sobre la escena de realidad aumentada, manejando los controles de vídeo y su carga para reproducción.
- RepassoController → Encargado de solicitar las preguntas a repasar por el alumno en cuestión.
- ResultController → Encargado de enviar a la base de datos los resultados de un alumno.

6.1.2. Componente de realidad aumentada

Vuforia [14] es un SDK que permite construir aplicaciones basadas en la Realidad Aumentada; una aplicación desarrollada con Vuforia utiliza la pantalla del dispositivo como un "lente mágico" en donde se entrelazan elementos del mundo real con elementos virtuales (como letras, imágenes, etc.). La cámara muestra a través de la pantalla del dispositivo, vistas del mundo real, combinados con objetos virtuales como: modelos, bloques de textos, imágenes, etc.

Una aplicación desarrollada con Vuforia ofrece la experiencia de reconocimiento de imágenes (como es en nuestro caso la “gravilla”).

Una aplicación desarrollada (la escena concretamente) con Vuforia está compuesta de los siguientes elementos:

Cámara: La cámara asegura que la imagen sea captada y procesada por el Tracker (rastreador).

Base de datos: La base de datos del dispositivo es creada utilizando el Target Manager (gestor objetivos); ya sea la base de datos local o la base de datos en la nube (en nuestro caso local), almacena una colección de targets (objetivos) para ser reconocidos por el tracker (rastreador).

Target: Son utilizadas por el rastreador (tracker) para reconocer un objeto del mundo real; los targets pueden ser de diferentes tipos; en nuestro caso **image targets** (imágenes; tales como: fotos, páginas de revistas, cubierta de libros, póster, tarjetas, etc).

Tracker: Analiza la imagen de la cámara y detecta objetos del mundo real a través de los frames de la cámara con el fin de encontrar coincidencias en la base de datos.

Secuencia de desarrollo de la escena:

1. El dispositivo capta una escena (un video en vivo) tomada a través de la cámara.
2. La SDK de Vuforia crea un frame (una imagen particular dentro de una sucesión de imágenes) de la escena capturada y convierte la imagen capturada por la cámara, a una diferente resolución para ser correctamente tratada por el tracker.
3. Vuforia SDK analiza la imagen a través del tracker y busca coincidencias en la base de datos, la cual esta compuesta por targets.
4. Luego la aplicación hace "algo"; este "algo" es renderizar algún contenido virtual (imágenes, videos, modelos, etc.) en la pantalla del dispositivo, y así crear una realidad mixta con elementos virtuales combinados con los elementos reales, o lo que se conoce como Realidad Aumentada, siendo en nuestro caso el renderizado del vídeo del servidor.

6.1.3. Escenas y componentes de la interfaz de usuario

En este apartado, hablaremos sobre los elementos que han sido implementados a las escenas así como de las mismas escenas. Tal y como se observa en la parte de diseño, las escenas son las que aparecen ahí y dentro de cada una de ellas, se componen de diferentes elementos.

Estos elementos son cámaras, luces direccionales (que permitirán enfocar la situación de la escena), canvas (como elemento de agrupación de los posibles elementos de interfaz de usuario posibles en cada escena), sonidos y objetos donde acoplaremos algunos de los controladores.

Panel de gestión administrador/docente

6.1.4. Ficheros PHP, HTML5 y hoja de estilo CSS3

De cara el panel de gestión docente, se han desarrollado una diversa variedad de ficheros, tanto PHP [3] como HTML5 [10] junto con su hoja de estilo CSS3 [11].

Estos elementos se encuentran ubicados en la máquina del servidor (“ucalearn.uca.es”):

- Fichero principal de formulario (formulario.php) → Este fichero contiene la pantalla principal de gestión para los docentes, con código HTML5 incrustado dentro de PHP para evitar mayor cantidad de ficheros externos. A su vez incluye una referencia a la hoja de estilos ubicada en el mismo nivel de directorio (/var/www/html). Permite realizar todas las operaciones de gestión de los docentes.
- Fichero auxiliar del formulario (editar.php) → Al igual que el fichero de formulario, hace uso de código PHP con código HTML5 incrustado y usa la misma hoja de estilos que “formulario.php”. Permitirá editar y borrar las preguntas pertenecientes a una asignatura.
- Fichero de gestión por parte del administrador (gestiona.php) → Permitirá al admin del sistema vincular o desvincular a un profesor de una asignatura.

Back-end (servidor)

6.1.5. Servicio REST

El servicio REST [6] para la consulta de calificaciones y autenticación desarrollado en el entorno *Eclipse* [7] en lenguaje Java [8], es desplegado en el servidor en el Tomcat8 [18] que posee instalado. Se encuentra desplegado en la ruta del Tomcat (/var/lib/tomcat8/webapps). Recibirá peticiones a la dirección:

“<http://ucalearn.uca.es:8080/restLearn/rest/service/consulta/{nombreAsignatura}/fecha{Y-m-d}>”.

6.1.6. Comunicación con aplicación móvil

La comunicación con la aplicación móvil se realiza en dos vertientes del siguiente modo:

- Comunicación con los ficheros o scripts de los controladores → Envío y consulta de asignaturas a cargar, preguntas y respuestas, etc.
- Envío de datos sobre el vídeo a cargar en la escena de realidad aumentada.

6.1.7. Comunicación con panel de registro/login

En el aspecto de comunicación con el panel de registro/login, es decir, nuestra mínima barrera de seguridad, se harán cargo los siguientes ficheros en el servidor:

- db.php → Encargado de conexión con la base de datos.
- forgot.php → Encargado de procesar el restablecimiento de contraseña.
- error.php → Notifica si no se da con éxito alguna operación en el portal.
- login.php → Almacena variables de sesión del docente/admin para cargar unas u otras operaciones.
- logout.php → Notifica el cierre de sesión.
- profile.php → Informa de la persona actualmente iniciada en el sistema.
- register.php → Procesa los datos de cara al registro de un nuevo docente.
- reset.php → Procesa el restablecimiento de contraseña actualizando la antigua contraseña.
- success.php → Valida si se cumplen los requisitos para una operación y notifica exitosamente.
- verify.php → Confirma la situación de dar de alta a un docente.

6.2. Proceso de obtención de información

La funcionalidad para obtener información es realizada de diferentes modos, entrando en mayor profundidad, si comparamos por un lado como lo hace la aplicación móvil (*Unity* [2]) y como lo hace el docente en su panel de gestión:

- Aplicación móvil:

El método para realizar dicha consulta se hace a través de los ficheros de código de los controladores mencionados anteriormente. Estos lanzan unas co-rutinas cuyo objetivo es el de enviar datos al servidor (a la base de datos), pedir datos al mismo o ambas.

En el lanzamiento de las co-rutinas, las peticiones de datos o envíos se realizan mediante la librería WWW (petición) y mediante WWWForm (envío).

- Panel de gestión:

Esta manera de consultar es diferente a la que se da en el caso de la aplicación móvil, y más sencilla, ya que como se ha comentado en apartados anteriores, *Unity* no da soporte para consultas MySQL [9].

Sin embargo, en los ficheros PHP [3] se utiliza la librería “*mysqli*” que permitirá en pocos pasos lanzar consultas para obtener datos o enviarlos a la base de datos.

Capítulo 7

Pruebas

Como es de esperar la mayoría de los entornos de desarrollo hoy en día nos proporcionan un mecanismo para analizar el correcto funcionamiento y resultado de nuestro sistema.

Por tanto, de cara a la comprobación del correcto funcionamiento del servicio REST [6], la aplicación móvil en *Unity* y los diferentes aspectos del panel de gestión docente/administrador, nos hemos ayudado en algunos casos de herramientas auxiliares que comentaremos a continuación en el apartado de tests unitarios.

7.1. Tests unitarios

Podemos definir el concepto de test unitario como un test que comprueba que una unidad concreta de código cumple exactamente con la funcionalidad requerida de ésta.

La idea del plan de pruebas propuesto en esta memoria es la de ejecutar primero los tests unitarios, comprobando que se cumplen los requisitos para dichos elementos y posteriormente realizar tests de integración.

Los elementos que van a ser testeados usando tests unitarios son:

- ◆ Servicio REST
- ◆ App Móvil
- ◆ Panel de gestión docente

7.1.1. Pruebas unitarias del servicio REST

Para realizar pruebas de este tipo sobre el servicio desarrollado y alojado en el servidor, se va a hacer uso de una herramienta externa o programa llamado *Advanced REST Client* que nos permitirá indicando una dirección y elementos de una cabecera realizar una consulta o enviar datos a dicho servicio, obteniendo una respuesta en un formato concreto.

Para nuestro caso, tenemos que los **datos de entrada** van a ser por ejemplo la dirección donde se aloja el servicio:

<http://ucalearn.uca.es:8080/restLearn/rest/service/consulta/{nombreAsignatura}/fecha{Y-m-d}>.

En este caso como nombreAsignatura introduciremos “Topografía” y en el caso de la fecha será “2018-06-03”.

Los **datos de salida esperados** deberán ser en formato JSON y recibir los datos de cada alumno así como el número de aciertos y total de preguntas realizadas.

En el momento de realizar la **ejecución y obtener los resultados**, tenemos:

```
[{"id":13,"usuario":"u99765432","tematica":"Topografía","aciertos":4,"total":4,"fecha":"2018-06-03;11:22:15.0"}, {"id":14,"usuario":"u31313467","tematica":"Topografía","aciertos":4,"total":4,"fecha":"2018-06-03;11:58:41.0"}].
```

7.1.2. Pruebas unitarias de la aplicación móvil

Se han realizado pruebas de dicho tipo sobre la app, en el modo en el que han sido probadas cada una de las escenas y su funcionamiento dentro de la aplicación.

Destacar que lo esperado es una correcta consecución de escenas, lo cual se cumple y la funcionalidad de cada uno de los botones es la esperada, llevándonos al siguiente lugar donde debe cada uno.

7.1.3. Pruebas unitarias del panel de gestión

Si enfocamos las pruebas a este ámbito, comentar que se han realizado comprobaciones de **datos de entrada** en los diferentes elementos que permiten operar sobre el panel (tanto de administrador como de docente) y su funcionalidad es la **esperada**, esperando obtener por ejemplo alertas sobre el éxito o fallo de una operación llevada a cabo.

7.2. Test de integración

Una vez hemos definido correctamente nuestros tests unitarios para cada uno de los elementos que nos competen, es hora de comprobar que el funcionamiento de estos, en conjunto, es el correcto.

Para ello el enfoque es realizar un test de integración donde se compruebe la correcta ejecución de una situación real del sistema global.

Para este test, hemos comprobado que se lancen los siguientes eventos:

- El docente procede a registrarse, y es dado de alta por el administrador, este inicia sesión y añade una nueva temática junto con su correspondiente enlace al vídeo.

Para concluir, añade pregunta sobre dicha asignatura y activa el examen de la misma.

- El vídeo es descargado a la vez en el servidor en el directorio destinado para ello (/descarga/files). En dicho momento el alumno procede a usar la aplicación en el modo examen.

- La calificación almacenada en el sistema (comunicación app-servidor) es consultada por el profesor en su panel de gestión al finalizar el alumno.

7.2.1. Resultados

Los resultados arrojados por este test de integración son los siguientes:

1. Registro, alta, inicio de sesión, añadir temática con vídeo, preguntas y activación de examen – **100%**
2. Descarga del vídeo y uso de la aplicación en modo examen – **100%**
3. Consulta de calificación almacenada en el sistema – **100%**

Por tanto, podemos dar por concluidos los tests de la aplicación.

Capítulo 8

Conclusiones y trabajo futuro

El objetivo de este capítulo es hacer un repaso por las conclusiones extraídas sobre este a la hora de afrontar este proyecto, por tanto, es un conjunto de conclusiones recogidas a posteriori a la realización de este a fin de obtener una visión más amplia de las tecnologías, las metodologías y el proyecto en sí.

Además, se presentarán un conjunto de tareas que quedan pendientes (bien como deuda técnica o bien como trabajo futuro) con el objetivo de exponer los aspectos mejorables de este proyecto, dando una visión crítica de este.

8.1. Valoración del proyecto

Este trabajo fin de grado surge con la idea de mejorar un proyecto para el departamento de Innovación.

Tras darme cuenta realmente de los efectos que puede tener sobre el alumnado y profesorado, con una exigencia de mantenimiento mínima, la cual solo la realiza el rol administrador, merecía la pena llevarlo a cabo como trabajo fin de grado.

Reúne muchas de las tecnologías, entornos y lenguajes sobre los que he trabajado en el transcurso de la carrera y además recoge aquello que más me gusta como es el entorno de desarrollo que proporciona *Unity* [2].

En un alumno esto puede ocasionar la aparición de ganas de encarar una asignatura con doble aliciente, como son el aprobado y el entretenimiento, algo que no en muchas asignaturas se consigue, como ya comentamos en la sección de motivación del proyecto.

Sobre un docente, esto puede servir como una alternativa a evaluar el contenido de una asignatura que no tenga la suficiente envergadura para ser calificado bajo un examen parcial, final o trabajo de prácticas. Además de ello, en una misma interfaz permite al profesor realizar cualquier tipo de gestión, sin necesidad de tener que recordar donde acceder para llevar a cabo distintas operaciones. Todas ellas se encuentran concentradas en una misma interfaz (lo cual facilita el trabajo).

La idea sobre la que gira este sistema no es como puede llegar a pensarse la de hacer de sustituto de un examen, si no que además de ayudar a preparar al alumno para el mismo, sirve para el docente como un complemento a los métodos de evaluación. Muchos complementos como son entrega de

cuestionarios, ejercicios y similares tienen una ponderación de un punto sobre la nota final. ¿Por qué “vAnswer” no puede tener el mismo derecho a ello?

La cantidad de información con respecto a algunas tecnologías utilizadas es limitada, por tanto parte del tiempo empleado en la realización de este proyecto ha sido dedicada a la investigación sobre estas tecnologías respecto a la aplicación móvil como a código alojado en el servidor.

La valoración del proyecto a posteriori es, por tanto, positiva. Personalmente creo que se han conseguido los objetivos marcados al principio de este y se ha llegado a desarrollar un sistema que permite evaluar a un alumno de manera adecuada, entretenida para el mismo y con facilidades para el docente.

8.2. Objetivos cumplidos

Al menos un 95% de los objetivos marcados al inicio del proyecto han sido satisfechos. Ahora nuestro sistema es capaz de evaluar a un alumno y permitir a un docente o administrador, realizar operaciones de gestión bajo una interfaz.

A continuación, se enumeran cada uno de los objetivos marcados y su grado de completitud:

- Implementación de aplicación móvil con realidad aumentada - 100 %
- Implementación de servicio REST para consulta de calificaciones - 100 %
- Implementación del Back-End de la aplicación - 100 %
- Implementación del Front-End de la aplicación - 100 %
- Elaboración de la memoria - 100 %

Como podemos observar, el porcentaje de las tareas completadas al 100 % es bastante alto, por lo que podemos determinar que el proyecto ha sido terminado, quedando las tareas restantes pendientes para el trabajo futuro en la aplicación.

8.3. Lecciones aprendidas

Dentro de este apartado se recogerán los conocimientos adquiridos durante el desarrollo de este proyecto, a nivel personal y a nivel técnico.

8.3.1. Conocimientos técnicos

Este sistema, como hemos comentado en secciones previas, puede ser considerado un ecosistema de tecnologías trabajando juntas para conseguir un propósito concreto: El de evaluar a un alumno en base a una asignatura y un modo.

Muchas de las tecnologías que forman parte de este sistema eran nuevas para mí al inicio del proyecto, por lo que se ha tenido que emplear gran cantidad de tiempo en su estudio y perfeccionamiento.

Se muestran algunas de las tecnologías que se han perfeccionado en el transcurso de este proyecto en las siguientes secciones:

Java

El lenguaje de programación Java [8] es el lenguaje que más he utilizado y que más conozco desde el inicio del grado, además, me atrae junto con el desarrollo en el entorno *Unity* [2] lo que más.

Por tanto es un lenguaje que considero importante. No han aparecido problemas durante el transcurso de este proyecto con este famoso lenguaje de programación ya que el desarrollo ha sido en relación a un servicio REST [6] alojado en el servidor de cierta sencillez.

PHP + PHPMailer

El lenguaje interpretado PHP [3] a lo largo de la carrera, lo he utilizado en diferentes asignaturas, pero no alcanzando la profundidad a la que se ha llegado en este proyecto.

Por tanto decir que me ha provocado la necesidad de adquirir aprendizaje más avanzado sobre el mismo, así como documentarme respecto a su extensión para envío de correos PHPMailer [4], la cual era sorpresa para mí y la consideraba necesaria para notificar a los usuarios docentes/administrador de cuando era necesario que tomaran alguna acción sobre el sistema.

HTML5 + CSS3

Aunque durante el transcurso de la titulación hemos aprendido bastante sobre ellos, ha sido necesario un repaso de conceptos básicos y algún que otro consejo a la hora de desarrollar.

A día de hoy, puedo escribir hojas de estilo con mucha más facilidad que antes de la realización de este proyecto.

JavaScript

JavaScript [5] es un lenguaje de programación que siempre me ha resultado interesante, por tanto la base era algo más sólida.

Por lo tanto de cara a ser usado en los ficheros de código para notificar de las diferentes alertas que suceden tras llevar a cabo operaciones, lo considero de vital importancia.

Vuforia

Aunque no era algo nuevo ya que lo he visto en una asignatura de la carrera, la profundización en el concepto de proyección de vídeo con controles ha provocado la tarea de aprendizaje y documentación sobre como es posible realizar dicha acción.

Tal característica ha sido incorporada en versiones recientes, ya que en su momento cuando hice uso de Vuforia [14], la proyección era de elementos en 3D sobre el patrón reconocido y no algo tan avanzado como vídeo.

Youtube-DL

Quizás algo que ha provocado el mayor quebradero de cabeza en mí. La situación previa a conocer al existencia de dicho programa de línea de comandos era la siguiente:

En la aplicación móvil los vídeos se encontraban integrados a través de una ruta dentro de los directorios de la app. Esto provocaría problemas futuros en el sentido de la escalabilidad de la app. Si actualmente la aplicación tiene un peso de 37MB, con solo un vídeo de una duración aproximada de dos minutos acabaría pesando 60MB aproximadamente.

Por tanto surge la necesidad de dar una solución totalmente distinta al método de lectura de dichos vídeos, evitando la carga de almacenamiento que esto conlleva en el teléfono del alumno.

Tras un gran período y gracias a indicaciones por parte de Guadalupe Ortiz Bellot, fui capaz de encontrar la manera y esto fue haciendo uso de “*Youtube-DL*” [21]. Esto permitiría que cuando el docente añade una nueva temática, en el servidor es descargado el vídeo correspondiente a dicha temática, siendo leído mediante buffering por la aplicación y ya que nos encontramos bajo red Wi-Fi, no ocasionaría coste alguno.

8.3.2. Conocimientos adquiridos en el ámbito personal

Además de los conocimientos técnicos adquiridos, es necesario hacer especial mención a los conocimientos no técnicos adquiridos durante el proceso del desarrollo de este proyecto, ya que también serán aplicados al ámbito profesional.

Los conocimientos adquiridos en el ámbito personal son:

- Conocer la metodología a seguir para la correcta realización de un proyecto de este tipo (Estudio de las tecnologías, análisis de la información, desarrollo, pruebas, documentación, etc).
- Saber que para conseguir un buen resultado, el tiempo que se necesita emplear es de mínimo cuatro o cinco horas al día, sinónimo de constancia.
- Conocer un poco más el departamento y en qué centran su labor investigadora.
- Aprender sobre uno de los aspectos de desarrollo que me parecen más relevantes y en auge: la realidad aumentada.
- Y, por supuesto, enfrentarme a la fase final de un proyecto personal que me propuse hace años: la titulación académica.

8.4. Mejoras y ampliaciones: trabajo futuro

Aunque, como hemos comentado en la sección anterior, los objetivos marcados al inicio del proyecto han sido satisfechos casi en un cien por cien, este tipo de proyectos puede y debe ir mejorando con el tiempo, a fin de conseguir un funcionamiento más sólido y estable y una mejor interacción con sistemas externos y con el usuario.

A continuación se enumerarán cada uno de los aspectos que deben mejorarse en iteraciones futuras del proceso de desarrollo, así como otros aspectos que permitirán que el producto sea de una calidad mayor.

8.4.1. Mejoras

Al igual que en todos los proyectos, existen varios aspectos del proyecto que nos compete que pueden ser mejorados. A continuación se exponen cada uno de ellos con detalle y se da una solución posible a éstos, creando un objetivo a tener en cuenta para otras posibles iteraciones.

Mejoras en la aplicación móvil

Debido a la envergadura del desarrollo de la aplicación móvil, se han relajado las partes de “buena estética” de la app. Con esto queremos decir que las imágenes de fondo, algunos textos y botones podrían haber sido realizados por uno mismo (siendo algunas obtenidas de fuentes de Internet), consiguiendo un mejor aspecto de homogeneidad en cuanto a colores y vistas se refiere.

Mejoras en el front-end de gestión

Aunque el front-end de la aplicación es adecuado y suficiente para mostrar al usuario el resultado de las operaciones que lleva a cabo, una mejora posible sería cambiar toda la capa de interacción con el back-end de la aplicación para usar AngularJS, un framework de Javascript en el que quiero profundizar, junto a aplicar una mejora en su estética (algo en lo que no soy habilidoso) mediante el uso de plantillas Bootstrap.

La inclusión de este framework AngularJS permitirá una comunicación Front/Back-end mucho más sencilla y estandarizada, y nos abstraerá de centrarnos en otros conceptos, permitiéndonos prescindir de ellos en iteraciones futuras del desarrollo.

Mejoras en la seguridad del panel de gestión

A pesar de que nuestro sistema se compone de una barrera de seguridad mínima como es un registro/login, quizás considero necesario en un futuro de aplicar ciertas mejoras.

Estas mejoras a considerar pueden ser por ejemplo, aplicar el uso de un protocolo SSL (conexiones seguras del tipo “<https://>”).

Mejora del plan de pruebas

El plan de pruebas de la aplicación necesita ser revisado, a fin de incluir mas pruebas de otros aspectos y algunos aspectos ya implementados, a fin de cerciorarnos a un 100 % de que su funcionamiento es correcto.

El plan es añadir algún tipo de automatización en el proceso de pruebas, a fin de hacer más cómodo y dinámico todo el proceso.

8.4.2. Ampliaciones al proyecto actual

A continuación se hará un breve repaso por cada una de las funcionalidades que añadiremos en un futuro al sistema:

Carga de preguntas de asignaturas

La carga de preguntas de las diferentes asignaturas actualmente puede ser algo llevadera en cuanto a tiempo, si se insertan una gran cantidad de preguntas por parte del docente.

En la etapa de diseño se planteó e incluso llegó a desarrollarse un proyecto en *Eclipse* [7] bajo código Java [8] para poder leer de un archivo de texto en formato GIFT (originario de la plataforma Moodle), bajo la cual realicé una petición de obtener un curso de prueba para generar preguntas, descargarlas y comprobar por mí mismo si el proyecto daba mucho problema al leer de dicho archivo.

El resultado de ello era el que se temía, en cuanto a que dificultaría al sistema de reconocer sobre qué asignatura se están insertando preguntas, así como no ser nada viable debido a que en la evaluación del alumno necesitamos tener preguntas de solo practicar, mixtas y de examen únicamente. Esto era imposible de reconocer de dicha manera, ya que no había ningún patrón posible para determinarlo.

En resumen, se debería plantear otro modo de inserción de preguntas que automatice el proceso.

Disponibilidad para iOS

El aspecto de ser viable para iOS es fundamental y debido a no disponer de mayor cantidad de tiempo, no ha sido posible realizarlo. En los inicios indagué como *Unity* [2] exporta la app para tal sistema, exportando no directamente el .apk para ser instalado, sino una serie de directorios con archivos que permitirán a un programa de desarrollo en iOS llamado “Xcode” leer el fichero con extensión .xcodeproj y construir así el archivo .ipa para instalar.

Bibliografía

- [1] *C#.* URL: https://es.wikipedia.org/wiki/C_Sharp (visitado 30-04-2018).
- [2] *Unity.* URL: <https://unity3d.com/es> (visitado 19-01-2018).
- [3] *PHP.* URL: php.net/ (visitado 03-05-2019).
- [4] *PHPMailer.* URL: <https://github.com/PHPMailer/PHPMailer> (visitado 08-10-2018).
- [5] *JavaScript.* URL: <https://es.wikipedia.org/wiki/JavaScript> (visitado 18-03-2018).
- [6] *REST.* URL: <https://www.codecademy.com/articles/what-is-rest> (visitado 18-02-2019).
- [7] *Eclipse.* URL: [https://es.wikipedia.org/wiki/Eclipse_\(software\)](https://es.wikipedia.org/wiki/Eclipse_(software)) (visitado 15-07-2018).
- [8] *Java.* URL: [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)) (visitado 16-05-2018).
- [9] *MySQL* URL: <https://es.wikipedia.org/wiki/MySQL> (visitado 16-06-2018).
- [10] *HTML5.* URL: <https://es.wikipedia.org/wiki/HTML5> (visitado 11-04-2018).
- [11] *CSS3.* URL: <https://developer.mozilla.org/es/docs/Web/CSS/CSS3> (visitado 13-07-2018).
- [12] *FortiClient VPN.* URL: <https://forticlient.com/> (visitado 19-05-2019).
- [13] *Android.* URL: <https://es.wikipedia.org/wiki/Android> (visitado 11-04-2019).
- [14] *Vuforia.* URL: <https://www.vuforia.com/> (visitado 16-02-2019).
- [15] *GitHub.* URL: <https://github.com/> (visitado 03-06-2019).
- [16] *Visual Studio.* URL: <https://visualstudio.microsoft.com/> (visitado 02-05-2019).

[17] *UML*. URL: <https://www.dcc.uchile.cl/~psalinas/uml/introduccion.html> (visitado 22-11-2018).

[18] *Tomcat 8*. URL: <https://tomcat.apache.org/> (visitado 28-12-2018).

[19] *PHPMyAdmin* . URL: <https://es.wikipedia.org/wiki/PhpMyAdmin> (visitado 09-01-2019).

[20] *LDAP*. URL: https://es.wikipedia.org/wiki/Protocolo_Ligero_de_Acceso_a_Directorios (visitado 07-06-2019).

[21] *Youtube-DL*. URL: <https://yt-dl.org.github.io/youtube-dl/index.html> (visitado 07-06-2019).

Anexo A

Programas utilizados

A continuación haremos un repaso por los programas de envergadura utilizados en el proceso de desarrollo de este sistema.

Esta sección puede ser de ayuda de cara a otras personas interesadas en leer esta documentación para aclarar ciertas dudas y, quizá, encontrar alguna herramienta software que se ajuste a sus necesidades.

A.I. Eclipse IDE

Eclipse es, sin duda, uno de los entornos integrados de desarrollo más utilizados dentro del mundo de la programación.

Además de tener una licencia libre y ser gratuito, viene con un SDK (Developer Kit) que es Open Source. Ésto, unido a su facilidad de uso y sus infinitas funcionalidades para integrarse con otros sistemas a través de los plugins, explica rápidamente por qué es uno de los más usados.

La pantalla de inicio de Eclipse es la siguiente:



Figura A.1: Pantalla de Inicio

Y la vista por defecto una vez arrancado el IDE es la siguiente:

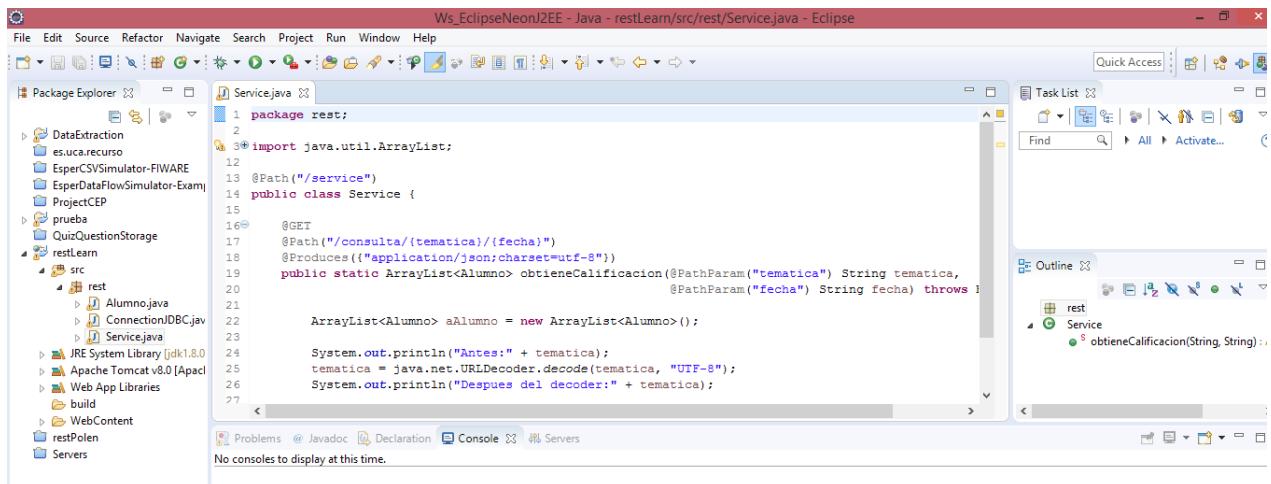


Figura A.2: Interfaz de Eclipse IDE

A.II. Advanced REST client

Es una extensión de Chrome como aplicación de escritorio, que nos permite lanzar peticiones a servicios o APIs RestFul. Puedes hacer cualquier tipo de petición además de las habituales GET, POST, PUT y DELETE. Permite pasar parámetros a las peticiones y muestra el resultado devuelto por el servicio que queremos probar.

Su funcionamiento es muy simple y no requiere ningún tipo de configuración especial.

Pongamos nuestro ejemplo de uso para pruebas unitarias sobre dicho servicio:

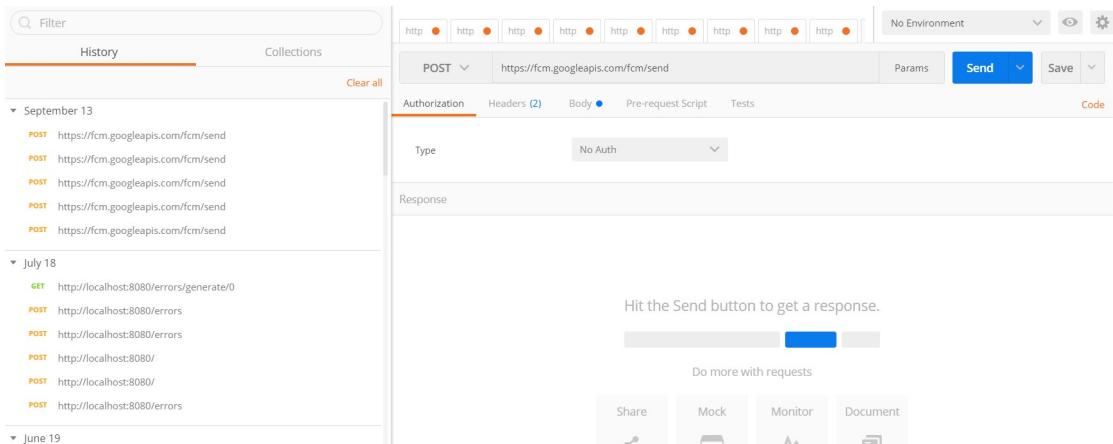


Figura A.3: Interfaz de Advanced REST client

A.III. PuTTY

PuTTY es un cliente SSH y Telnet con el que podemos conectarnos a servidores remotos iniciando una sesión en ellos que nos permite ejecutar comandos. El ejemplo más claro es cuando empleamos *PuTTY* para ejecutar comandos en un servidor VPS y así poder instalar algún programa o configurar alguna parte del servidor.

En nuestro caso ha sido utilizado para comunicarnos y gestionar el paso de todos los ficheros en lo que respecta al servidor de la universidad (back-end del servidor).

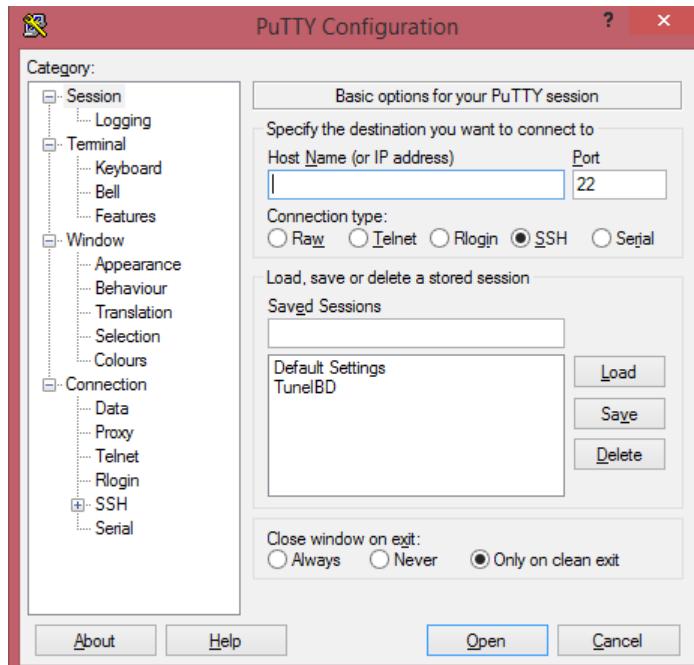


Figura A.4: Interfaz de PuTTY

A.IV. Unity

Unity [2] es un motor de videojuego multiplataforma creado por Unity Technologies. *Unity* está disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux.

La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas. A partir de su versión 5.4.0 ya no soporta el desarrollo de contenido para navegador a través de su plugin web, en su lugar se utiliza WebGL. Unity tiene dos versiones: Unity Professional (pro) y Unity Personal.

En nuestro caso usamos *Unity* versión 2018.1 (Personal).

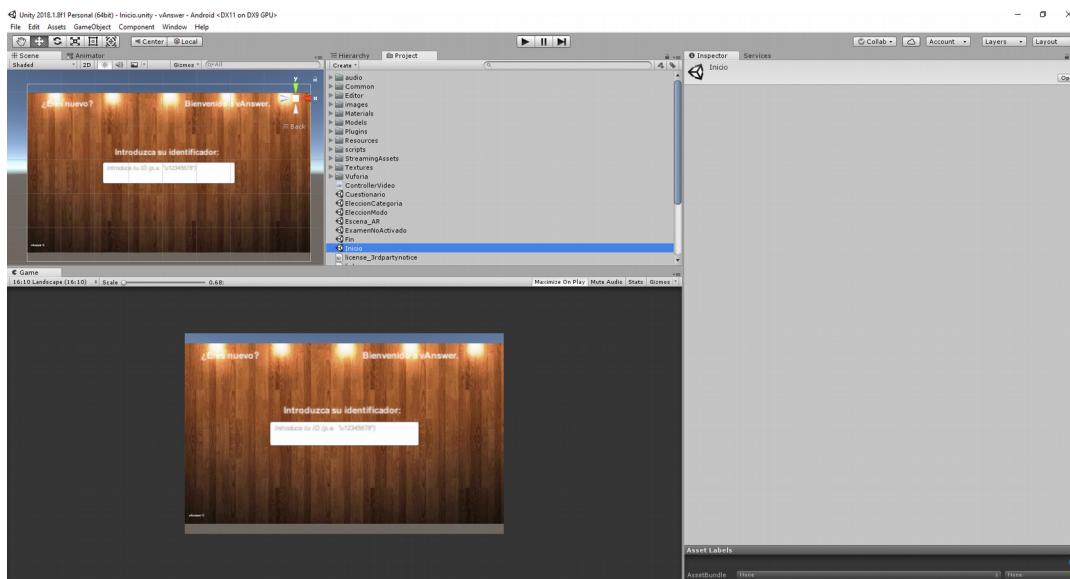


Figura A.5: Interfaz de *Unity*

A.V. MySQL Workbench 6.3 CE

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL [9].

En nuestro caso, únicamente haremos uso de ella para obtener el esquema y las relaciones de la base de datos utilizada.

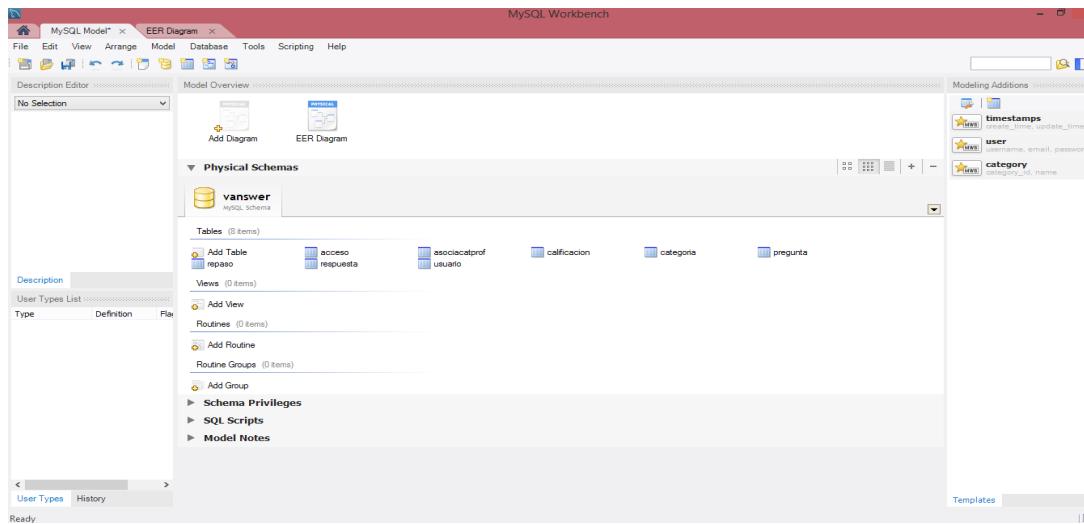


Figura A.6: Interfaz de MySQL Workbench

A.VI. Gantt Project

En nuestro lugar haremos uso del programa *Gantt Project* para realizar la planificación de las etapas fundamentales del proyecto.

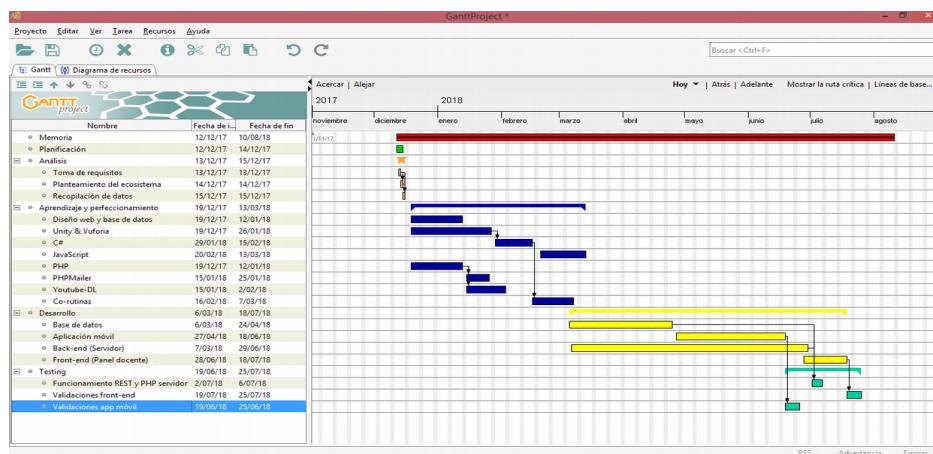


Figura A.7: Interfaz de Gantt Project

Anexo B

Manual del usuario

En este anexo haremos un breve repaso por el proceso a seguir por parte del usuario final de la aplicación para ejecutarla y visualizar sus resultados.

Requisito previo (importante)

Antes de comenzar con nuestro manejo sobre el sistema debemos de cumplir el siguiente requisito.

Estar conectado a la red de la universidad para tener acceso y permitir la comunicación con el servidor.

Para ello si nos encontramos físicamente en la universidad, no existe ningún problema ya que la comunicación funcionará.

Si no es así, procedemos a realizar lo siguiente ya usemos la app móvil o el portal de gestión:

Conectarse a la siguiente dirección y descargar la versión de la aplicación para Windows/Android:

<http://forticlient.com/#download>

Una vez descargado instalar la aplicación con los valores por defecto, excepto en la pantalla:



Figura B.1: Instalación FortiClientVPN (I)

En la que seleccionaremos la opción “**VPN Only**” para evitar la instalación de otros componentes no necesarios.

Una vez instalada la aplicación tendremos el siguiente icono:



Configuración de la aplicación

Una vez instalada la aplicación la ejecutamos y obtendremos la siguiente pantalla:

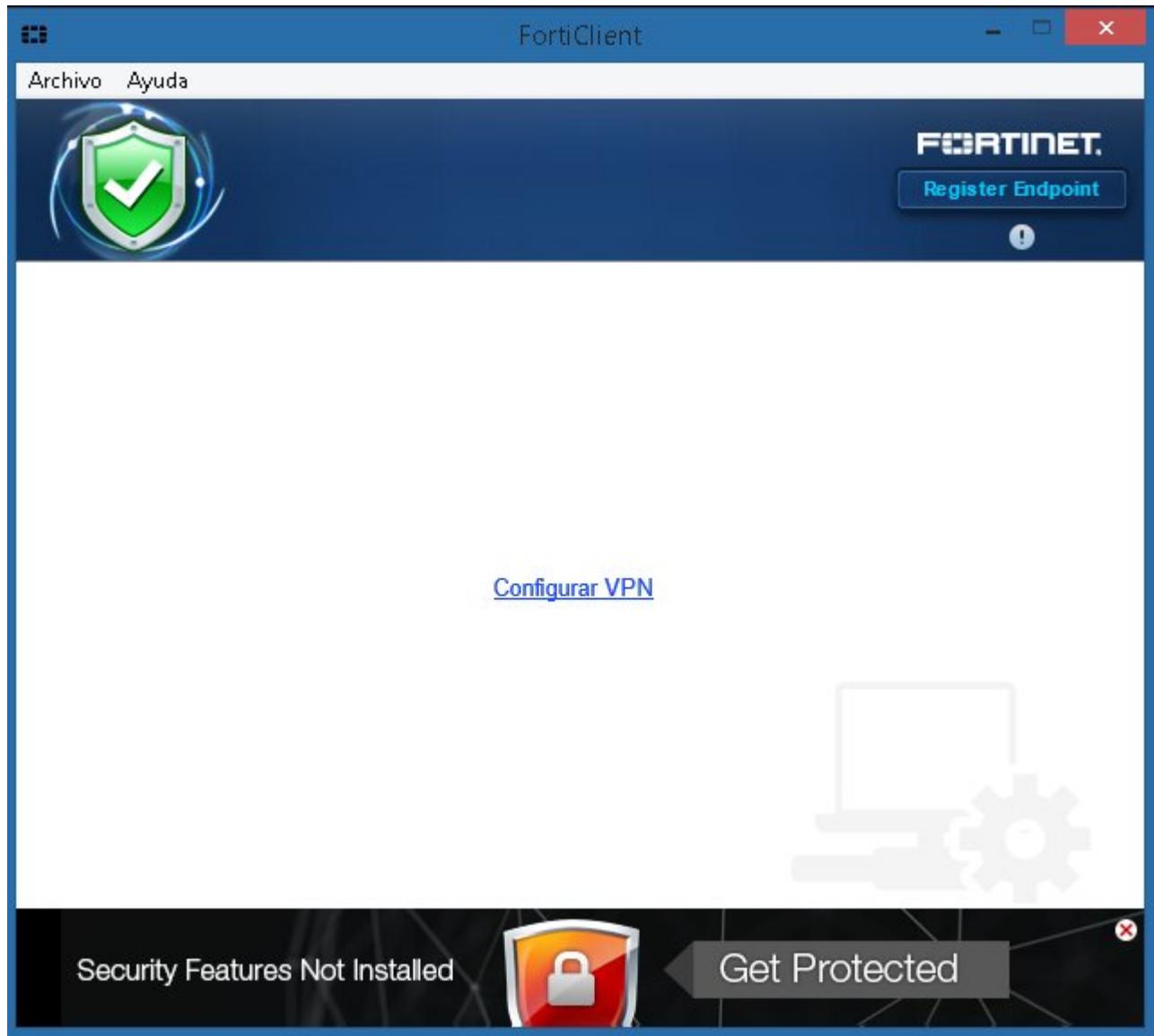


Figura B.2: Interfaz configuración VPN

Pulsaremos en la opción “**Configurar VPN**” e indicaremos los parámetros de configuración:



Figura B.3: Configuración avanzada VPN

Los parámetros que debemos indicar son los siguientes:

Nombre de Conexión: UCA

Gateway Remoto: vpn.uca.es

Nombre de Usuario: ahí indicaremos el identificador de usuario **uDNI** usado para acceder a los servicios académicos, campus virtual, ... o bien la dirección de correo electrónico.

Luego pulsaremos **Aplicar** y ya se puede iniciar la conexión a VPN.

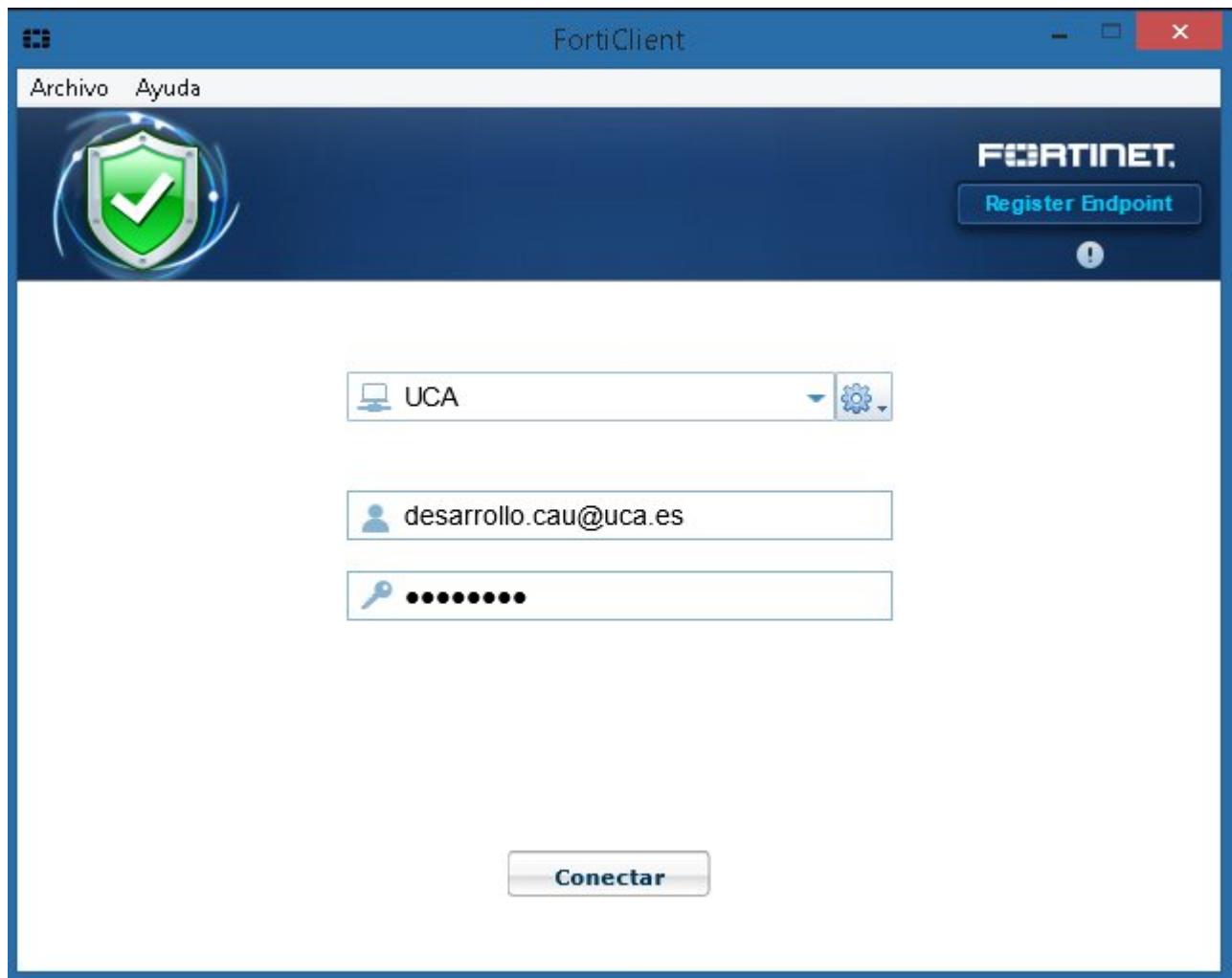


Figura B.4: Conexión a VPN

Indicaremos el identificador de usuario o email y la contraseña, luego pulsaremos el botón de **Conectar** y se realizará la conexión.



Figura B.5: Conectado al VPN.

Una vez conectada aparecerá un ícono de color verde para indicar éxito en la conexión.

B.I. Rol administrador

Previo a todo lo que procedamos a realizar, debemos cerciorarnos de que nos encontramos bajo la red de la universidad o de lo contrario no tendremos la capacidad para manipular nada en el sistema.

1. Accedemos a la dirección: ucalearn.uca.es/admin
2. Nos encargamos de iniciar sesión con las credenciales de admin y acceder al formulario de gestión (esta redirección se hace automática al tener éxito el log-in). Si olvida la contraseña existe una opción para el restablecimiento de la misma enviándose un enlace a su correo. La gestión del alta de docentes la realiza el administrador (vansweruca@gmail.com).d
3. En el formulario se le permite vincular y desvincular profesores de asignaturas, en el caso que estos sean dados de baja o en el caso de que un profesor coordinador haya generado una temática y existan más profesores asociadas a ella.

B.II. Rol docente

Previo a todo lo que procedamos a realizar, debemos cerciorarnos de que nos encontramos bajo la red de la universidad o de lo contrario no tendremos la capacidad para manipular nada en el sistema.

1. Accedemos a la dirección: ucalearn.uca.es/admin
2. Nos aparecerá el portal de registro/log-in del profesorado. Si nos registramos se envía un correo al administrador (en este caso vansweruca@gmail.com) encargado de dar de alta a un profesor. Una vez dado de alta, podrá iniciar sesión y acceder al formulario (esta redirección se hace automática al tener éxito el log-in). Si olvida la contraseña existe una opción para el restablecimiento de la misma enviándose un enlace a su correo.
3. En el formulario se le permite introducir temáticas con su vídeo, activar/desactivar examen de una asignatura, cargar preguntas de una temática para ver, editar y borrar, consultar calificaciones de un alumno, introducir nuevas preguntas V/F y preguntas de múltiples respuestas.

B.III. Rol alumno

1. Iniciamos la aplicación FortiClient VPN una vez instalado el juego en nuestro dispositivo (vAnswer). Si queremos practicar desde casa deberemos usar FortiClient VPN para conectarnos a la VPN de la uca (vpn.uca.es) con nuestras credenciales con las que accedemos al campus. Otorgamos permisos a la app a acceder a la cámara y a nuestros ficheros multimedia.
2. Damos al botón comenzar, y posteriormente deberemos introducir nuestro identificador de usuario. Se autentica al alumno mediante el LDAP [20] de la universidad.
3. Tras ello, elegimos modo de juego. De los tres modos de juego posibles, tenemos lo siguiente:
 - 3.1. Si elegimos el modo de juego "Modo práctica - Completo" realizamos preguntas de tipo aprendizaje que no contará para nota, sobre una asignatura que posteriormente elegiremos.
 - 3.2. Si elegimos el modo de juego "Modo práctica – Repasar" realizamos preguntas que previamente hallamos fallado sobre dicha asignatura elegida. Una vez se aciertan en el modo práctica dichas preguntas, son limpiadas de nuestro banco de preguntas falladas.

Si no existen preguntas a repasar se nos mostrará una pantalla de completado y un botón para volver a la elección de modo y temática.
 - 3.3. Si elegimos el modo de juego "Modo Evaluación" realizamos una serie de preguntas del tipo examen (no todas las de una asignatura realizadas en el modo práctica caerán, para sorprender al alumno), cuya calificación será almacenada en la base datos y posteriormente consultada por el profesor en cuestión. Si el examen de dicha asignatura no está activado por el profesor, el alumno no podrá realizarlo, y se le mostrará una pantalla correspondiente para que vuelva a elegir modo y temática.
4. Una vez elegido el modo, elegiremos la asignatura sobre la que queremos jugar (temática).
5. Se nos cargará la escena de realidad aumentada donde debemos enfocar la cámara a un patrón de imagen correspondiente para que el vídeo se proyecte en realidad aumentada, con controles de “play” y “pause”, como ya comentamos previamente.
6. Cuando terminemos de ver el vídeo, pulsamos el botón realizar cuestionario y comenzarán nuestras preguntas del tipo V/F y respuestas múltiples. Cuando no queden preguntas aparecerá la escena Game Over, permitiendo volver a jugar o salir del juego.

Anexo C

Descarga del proyecto

Se encuentra disponible para su libre uso, tal y como se indica mediante el uso de la licencia utilizada GNU Public License, en un repositorio, el proyecto con todas las utilidades necesarias, así como con los ficheros que se encuentran alojados en el servidor para futuras consultas o cambios.

Enlace a la descarga del proyecto:

<https://github.com/vansweruca/TFG-vAnswer>

Si nos hallamos en una máquina con distribución de Linux, podemos descargarlo directamente mediante el siguiente comando en una ventana en la Terminal:

```
$ git clone https://github.com/vansweruca/TFG-vAnswer.git
```

Debido a la gran cantidad de ficheros en la carpeta del proyecto *Unity* [2] se adjunta un fichero de texto con un enlace a MEGA para dicha descarga externa.

Los ficheros alojados en el servidor hacen uso de Tomcat8 [18] y PHPMyAdmin [19] en cuanto al servicio REST [6] y base de datos, respectivamente.

Los ficheros de código que proporcionan toda la comunicación necesaria se encuentran alojados en el directorio de la máquina Ubuntu de la universidad en: /var/www/html (Se adjuntan copias de los mismos).

Anexo D

GNU Free Documentation License

Copyright ©

Version 1.3, 3

November 2008

2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of

the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, TeXinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public. A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language.

(Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public. It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles. You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization. “Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

12. HOW TO USE THIS LICENSE FOR YOUR DOCUMENTS

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright $\$c$ YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.