

# ClassQuest: Gestión de la motivación como estrategia de aprendizaje.

## **Estudiante**

Daniel Moreno Arellano

Máster Universitario de Desarrollo de Aplicaciones para Dispositivos Móviles

M0.659 - Treball final de màster DADM

## **Consultor**

Eduard Martín Lineros

## **Profesor responsable de la asignatura**

Carles Garrigues

3 de Enero de 2020



Figura 1. Licencia Creative Commons

Este proyecto está desarrollado por **Daniel Moreno Arellano**. De la misma manera, esta obra está sujeta a una licencia **Creative Commons** del tipo Reconocimiento – No Comercial.

Puede encontrar más información en el siguiente enlace:

[Reconocimiento-NoComercial 3.0 España de Creative Commons](#)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>ClassQuest: Gestión de la motivación como estrategia de aprendizaje.</i>
<b>Nombre del autor:</b>	<i>Daniel Moreno Arellano</i>
<b>Nombre del consultor/a:</b>	<i>Eduard Martín Lineros</i>
<b>Nombre del PRA:</b>	<i>Carles Garrigues</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>10/2019</i>
<b>Titulación:</b>	<i>Máster universitario de Desarrollo de Aplicaciones para Dispositivos Móviles</i>
<b>Área del Trabajo Final:</b>	<i>M0.659 - Treball final de màster DADM</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>Gamificación, Educación, TIC</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b> Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.	
<p>La introducción de las TIC en el ámbito educativo ha permitido la aparición de nuevos retos en las metodologías docentes y en el modo alumnado se relaciona con estas. De este modo, dispositivos como las tabletas o móviles hacen presencia en el aula para formar parte de un complemento funcional más en el proceso educativo.</p> <p>El objetivo que este proyecto persigue es implementar una aplicación que permita al docente combinar las TIC con nuevas metodologías que fomenten la participación y motivación del alumnado durante el transcurso de las actividades.</p> <p>De este modo, la aplicación permitirá al alumnado la creación de un avatar personalizable que podrá ir siendo mejorado en consecuencia a los resultados que obtenga en las diferentes actividades o comportamiento en el aula. Además de un historial de sus actividades y registro de logros.</p> <p>En consecuencia, el profesor dispondrá de la gestión de las diferentes clases en las que podrá generar accesorios para determinadas actividades, bonificar o penalizar actitudes, fomentar tareas en grupo o planificar acciones que fomenten el trabajo en equipo.</p>	
<b>Abstract (in English, 250 words or less):</b>	
<p>The introduction of ICT in the educational field has allowed the emergence of new challenges in teaching methodologies and the way the student is related to</p>	

them. Consequently, devices such as tablets or mobiles are present in the classroom to be part of a functional complement in the educational process.

The objective of this project is to implement an application that allows teachers to combine ICT with new methodologies that encourage student participation and motivation during the course of activities.

Thus, the application will allow students to create a customizable avatar that can be improved accordingly to the results obtained in the different activities or behavior in the classroom. Also, check the evolution of their activities and record of achievements.

Consequently, the teacher will have a management of the different classes in which he can generate accessories for certain activities, bonus or penalize attitudes, encourage group tasks or plan actions that encourage teamwork.

# Notaciones y convenciones

A continuación, se muestra un listado de los estilos tipográficos que se detallarán en la memoria para poder distinguir su función dentro del documento.

## Título de primer nivel

### Título de segundo nivel

Texto normal

### Texto resaltado

### Referencia cruzada

Figura N. Descripción de la imagen

Tabla N. Descripción de la tabla

- Indicación de lista
- Indicación de sub lista

Línea de código

Importación/Referencia de clase en Swift

Palabra reservada en Swift

Texto en swift

Importación de pod en CocoaPods  
Código JSON

Cabecera de tabla
Celda de tabla

# Índice

<b>1. Introducción</b>	<b>8</b>
1.1 Contexto y justificación del Trabajo	8
1.2 Objetivos del Trabajo	9
1.3 Enfoque y método seguido	9
1.4 Planificación del Trabajo	10
1.4.1. Recursos de hardware	10
1.4.2. Recursos de software	11
1.4.3. Entregas establecidas (Puntos de control)	11
1.4.4. Diagrama de Gantt	12
1.5 Breve resumen de productos obtenidos	13
1.6 Breve descripción de los otros capítulos de la memoria	14
<b>2. Resto de capítulos</b>	<b>15</b>
2.1 Arquitecturas y tecnología	15
2.2 Base de datos	16
2.3 Diagrama UML	17
2.4 Perfiles de usuario	18
2.5 Casos de Uso	19
2.6 Diagrama de flujo	21
2.7 Diseño	21
2.7.1 Wireframe de baja fidelidad	21
2.7.2 Wireframe de alta fidelidad	24
2.7.3 Prototipo	28
2.7.4 Logotipo	28
2.7.5 Usabilidad	28
2.8 Instalación y preparación del entorno de trabajo	29
2.8.1 Cliente	29
2.8.2 Servidor	29
2.9 Implementación	30
2.9.1 Cliente	30
2.9.1.1 CocoaPods	30
2.9.1.2 Firebase	31

<b>2.1.1</b>	<b>Servidor</b>	<b>38</b>
<b>2.1.1.1</b>	<b>Autenticación</b>	<b>38</b>
<b>2.1.1.2</b>	<b>Persistencia</b>	<b>39</b>
<b>2.1.1.3</b>	<b>Almacenamiento</b>	<b>39</b>
<b>2.1.2</b>	<b>API's utilizadas</b>	<b>40</b>
<b>2.2</b>	<b>Juegos de pruebas / Usabilidad y UX</b>	<b>40</b>
<b>2.2.1</b>	<b>Evaluación de requisitos</b>	<b>40</b>
<b>2.2.2</b>	<b>Evaluación con usuarios</b>	<b>41</b>
<b>2.2.3</b>	<b>Manual de usuario</b>	<b>44</b>
<b>2.2.3.1</b>	<b>Iniciar sesión</b>	<b>45</b>
<b>2.2.3.2</b>	<b>Registro</b>	<b>46</b>
<b>2.2.3.3</b>	<b>Perfil</b>	<b>48</b>
<b>2.2.3.4</b>	<b>Listado de clase</b>	<b>50</b>
<b>2.2.3.5</b>	<b>Registro de actividad</b>	<b>51</b>
<b>2.2.3.6</b>	<b>Tienda</b>	<b>51</b>
<b>2.2.3.7</b>	<b>Listado de Quests</b>	<b>53</b>
<b>2.3</b>	<b>Identificación y corrección de errores</b>	<b>54</b>
<b>2.4</b>	<b>Versiones de la aplicación</b>	<b>55</b>
<b>2.4.1</b>	<b>Versión Beta</b>	<b>55</b>
<b>2.4.2</b>	<b>Versión release</b>	<b>56</b>
<b>2.5</b>	<b>Proyección a futuro</b>	<b>56</b>
<b>3.</b>	<b>Conclusiones</b>	<b>57</b>
<b>4.</b>	<b>Glosario</b>	<b>60</b>
<b>5.</b>	<b>Bibliografía</b>	<b>61</b>
<b>6.</b>	<b>Anexos</b>	<b>62</b>
<b>6.1</b>	<b>Anexo 1: Test de usuarios</b>	<b>62</b>

## Lista de figuras

Figura 1. Licencia Creative Commons	2
Figura 2. Nivel 1 Diagrama de Gantt	12
Figura 3. Nivel 3 Diagrama de Gantt	13
Figura 4. Patrón Modelo-Vista-Controlador	16
Figura 5. Diseño de la base de datos	17
Figura 6. Diseño de clases UML	18
Figura 7. Descripción ficha buyer persona	19
Figura 8. Casos de uso	20
Figura 9. Diagrama de secuencia de la app	21
Figura 10. Wireframes de baja fidelidad	24
Figura 11. Wireframes de alta fidelidad	28
Figura 12. Logotipo de la app	28
Figura 13. Organización del proyecto	30
Figura 14. Datos de la app en Google Analytics	38
Figura 15. Autenticación de usuario/a	38
Figura 16. Métodos de acceso de la app	39
Figura 17. Gestor de ficheros Cloud Storage	39
Figura 18. Resultado ejecución XCUITest	43
Figura 19. Iniciar sesión	45
Figura 20. Error de inicio	46
Figura 21. Registro de usuario 1/2	46
Figura 22. Registro de usuario 2/2	47
Figura 23. Error de registro	47
Figura 24. Unión a la clase	48
Figura 25. Perfil de usuario	48
Figura 26. Detalle del perfil	49
Figura 27. Detalle de poder/sentencia	50
Figura 28. Listado de clase	50
Figura 29. Listado de equipos	50
Figura 30. Registro de actividad	51
Figura 31. Categorías de objetos	51
Figura 32. Tienda	52
Figura 33. Confirmación de compra	52
Figura 34. Artículo comprado	52
Figura 35. Objeto ya comprado	53
Figura 36. Listado de Quests	53
Figura 37. Información de la quest	54
Figura 38. Listado de tareas	54

## Lista de tablas

Tabla 1. Fechas de entrega	12
Tabla 2. Horario laboral del recurso	12
Tabla 3. Estadísticas de personaje	46

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

En los últimos años, se ha vivido un impulso sobre el uso de tecnologías de la información y comunicación (TIC) en las aulas. De este modo, elementos cotidianos como pizarras o libros de texto han ido cediendo el relevo a pizarras digitales o tabletas, convirtiéndolos a día de hoy en un complemento funcional más en la educación.

Del mismo modo, elementos como los dispositivos móviles y el uso de aplicaciones han ganado su hueco en las aulas. A día de hoy, existen multitud de aplicaciones que permiten a docentes disponer de nuevas herramientas para hacer llegar el conocimiento a los alumnos y alumnas, consiguiendo que el sistema educativo y aprendizaje del alumnado se adapte a la nueva era digital.

De esta manera, la incorporación de un aprendizaje digital, acerca al profesor a poder ofrecer al alumnado nuevos recursos que potencien el interés por aprender, al mismo tiempo que habilitan la posibilidad de trabajar metodologías como la gamificación en un entorno más atractivo del planteamiento de clase magistral.

Por lo tanto, se puede observar la necesidad de nuevas metodologías que actúen acorde a los nuevos recursos del aula, guiando a la implementación de estrategias educativas que fomenten un aprendizaje significativo. De esta forma, nuevas preguntas deben ser respondidas: ¿Las TIC aportan un significado al aula? ¿Qué motiva al alumnado a aprender? ¿El alumnado realmente percibe en su aprendizaje una simbiosis entre TIC y metodologías?

Con estas observaciones, este proyecto pretende abordar unos de los aspectos fundamentales en la educación: el interés por el cómo se aprende [1]. De esta forma, beneficiarse de la incorporación de las TIC en el aula prestando atención al diseño de herramientas digitales que faciliten la incorporación de nuevas metodologías de aprendizaje. Ofreciendo al profesorado una app como complemento educativo que mediante el juego y la gamificación permita acercar al alumnado a las TIC durante su aprendizaje y, al mismo tiempo, mantenga la motivación y esfuerzo del alumnado.

## 1.2 Objetivos del Trabajo

El proyecto que se expone en esta memoria trata sobre el desarrollo de una aplicación móvil que incremente la motivación y participación del alumno durante el desarrollo de las actividades. Para llevar a cabo esto, se propone aplicar la gamificación como estrategia e hilo conductor en la transformación de la experiencia que vive el alumnado durante el curso escolar.

A continuación, se presentan los objetivos que deberá cumplir la versión actual del proyecto:

### Objetivos

- Implementación de una aplicación para dispositivos móviles.
- Organización de las diferentes asignaturas.
- Gestión de grupos, jugadores y clases.
- Avatar personalizable del alumnado.
- Tienda para adquirir modificaciones del avatar.
- Registro de actividad individual/grupo en la aplicación.
- Sistema de bonificación y penalizaciones:
  - En la puntuación de las entregas de actividades.
  - En el comportamiento durante la clase.
  - Implementación de un sistema de ranking.

### Objetivos ampliados

- Implementación de una aplicación multiplataforma.
- Implementación de un sistema de comunicación profesor/a-grupo.
- Mascotas para los avatares.
- Sistema de penalizaciones en la competición con otros grupos.
- Creación de retos.

## 1.3 Enfoque y método seguido

En la primera fase, se realizará un estudio de las diferentes aplicaciones existentes relacionadas con el aprendizaje y la gamificación. Debido a que el objetivo es implementar un producto nuevo que siga la línea de otros casos de éxito en el mercado para asegurar que se está ofreciendo un producto competitivo y funcional.

De este modo, este estudio servirá para plantear la situación de partida de la aplicación teniendo en consideración los puntos fuertes de otras aplicaciones pudiendo adaptar o perfilar los requerimientos propuestos entorno a la gamificación y motivación.

Teniendo en cuenta que el proyecto combinará tareas que engloban funcionalidades que dependen entre ellas, el cumplimiento y especificación de un calendario se convierte en un elemento clave durante el desarrollo del mismo. De la misma manera, se destacan una serie de puntos de control (hitos) que analicen si los objetivos propuestos se están llevando a cabo de la forma en que se planificaron o

de lo contrario se han de aplicar medidas correctivas en dicho caso. Estos puntos de control se centran en las fechas de entregas propuestas por la asignatura.

Asimismo, se realizarán revisiones periódicas para comprobar si se está llevando a cabo la planificación inicial, quedando en estudio posibles cambios en el diagrama de Gantt.

También se ejecutará un juego de pruebas el cual tiene como objetivo detectar y corregir posibles incidencias. Por lo tanto, el periodo de testeo será clave debido a que se interactúa con el sistema en tiempo real. Respecto a posibles evaluaciones con el consultor, ya que no se realizan en un lugar físico, se propone si fuese necesario, una conexión con alguna herramienta de video llamada como Skype para comprobar la correcta ejecución del sistema.

Una vez completada y alcanzada la parte de conectividad, interactividad y funcionalidad del sistema, es decir, se obtiene un sistema completo, la atención se centra en el desarrollo del apartado visual y gráfico del sistema. La estrategia que se plantea es priorizar la usabilidad, ofreciendo una interfaz clara e intuitiva, hecho que va impuesto debido a que el usuario final será un niño o niña que se inicia en las nuevas tecnologías. De este modo, se trabajará con wireframes de bajo nivel en el proceso de diseño inicial, para más tarde conseguir wireframes de alto nivel que cumplan las expectativas deseadas para el proyecto.

En definitiva, la estrategia empleada se rige por el conjunto de fases siguientes: estudio del mercado y situación, planificación, desarrollo, juego de pruebas, solución de incidencias y apartado gráfico. Se ha escogido esta metodología debido a que la parte funcional tiene prioridad ante el resto de partes; con este motivo se realiza un estudio previo de que se quiere hacer, se planifica la metodología de trabajo, revisiones, juegos de pruebas parciales y correcciones de las incidencias detectadas. De esta manera, el proyecto avanza mientras se van haciendo los controles oportunos, evitando únicamente una evaluación final del proyecto.

Además, en el tramo final para verificar que se ha realizado un buen desarrollo del diseño y la experiencia de usuario (UX), se realizará un juego de pruebas sobre el resultado obtenido elaborando test de cronología de sentimientos en la app.

## **1.4 Planificación del Trabajo**

### **1.4.1. Recursos de hardware**

Los elementos utilizados en el desarrollo del proyecto han sido los siguientes:

- **Ordenador de sobremesa:** Intel Core i5-7500 3700Mhz, 16GB RAM DDR3, NVidia GTX 1060, Windows 10 Professional 64bits. Utilizado para desarrollar todo el proyecto.
- **iPod Touch 6ª generación:** Dispositivo que permite reproducir música, vídeos, instalación de aplicaciones de la Apple Store o xCode. Utilizado para desarrollar y probar la app durante su ejecución.

#### 1.4.2. Recursos de software

- **MacOS High Sierra 10.13.1:** es la decimocuarta versión de macOS (anteriormente OS X), el sistema operativo de Apple para sus ordenadores de escritorio, portátiles y servidores Macintosh. Utilizado para ejecutar el IDE de programación. Debido a una incompatibilidad de la API de Firebase con la versión máxima actualizable del SO, se actualiza a Catalina 10.15.1, tal como se detalla en el punto [2.3 Identificación y corrección de errores](#).
- **VMware Workstation 15.5 Player:** Software que proporciona virtualización de sistemas operativos. Utilizado para virtualizar el sistema operativo MacOs.
- **Xcode 11.2.1:** es un entorno de desarrollo integrado (IDE, en sus siglas en inglés) para macOS que contiene un conjunto de herramientas creadas por Apple destinadas al desarrollo de software para macOS, iOS, watchOS y tvOS. Utilizado para realizar la aplicación utilizando el lenguaje Swift y el emulador virtual.
- **Microsoft Project:** Es un software de administración de proyectos. Utilizado para la realización del Diagrama de Gantt.
- **Google Docs:** Procesador de textos utilizado para elaborar la documentación del proyecto.
- **Google Slides:** Gestor de presentaciones utilizado para realizar la presentación del proyecto.
- **Moqups.com:** Herramienta online utilizada para realizar el diseño de los wireframes de baja fidelidad.
- **Adobe XD:** Maquetador de prototipos utilizada para realizar el diseño de los wireframes de alta fidelidad y prototipo funcional.

#### 1.4.3. Entregas establecidas (Puntos de control)

El proyecto está formado por 4 PACS evaluables que se establecen para el control y desarrollo del proyecto en base a un ciclo de vida real. Además de estas entregas se presentan un conjunto de evaluaciones posteriores en base al proyecto. Estas fechas han de tenerse en cuenta en la planificación temporal del proyecto.

En la siguiente tabla se reflejan estas entregas:

Fecha de entrega	Evaluación	Descripción
09/10/2019	PAC1. Plan de trabajo	Definición del proyecto. Primera versión de la memoria

30/10/2019	PAC2. Diseño	Por especificar
11/12/2019	PAC3. Implementación	Por especificar
03/01/2020	PAC4. Entrega final	Memoria y video Última versión de la memoria
Por especificar	Presentación del proyecto	Defensa ante el tribunal

Tabla 1. Fechas de entrega

#### 1.4.4. Diagrama de Gantt

A continuación, se muestra el nivel 1 del diagrama de Gantt utilizado para planificar el proyecto:

Modo de	Nombre de tarea	Duración	Comienzo	Fin	Trabajo
▶	PAC1. Plan de trabajo	16 días	18/09/2019	09/10/2019	26 horas
▶	PAC2. Diseño	15 días	10/10/2019	30/10/2019	82 horas
▶	PAC3. Implementación	30 días	31/10/2019	11/12/2019	178 horas
▶	PAC4. Entrega final	17 días	12/12/2019	03/01/2020	48 horas
▶	Defensa del proyecto	5 días	13/01/2020	17/01/2020	1 hora

Figura 2. Nivel 1 Diagrama de Gantt

Para llevar a cabo este proyecto se asigna un único recurso que ejerce las funciones de jefe de proyecto. El horario de trabajo del recurso Daniel Moreno Arellano se detalla en:

Día de la semana	Horario
Lunes	9:00 - 13:00
Martes	9:00 - 21:00
Miércoles	9:00 - 13:00 / 17:00 - 19:00
Jueves	9:00 - 17:00
Viernes	9:00 - 13:00 / 18:00 - 21:00
Sábado	9:00 - 20:00
Domingo	9:00 - 20:00

Tabla 2. Horario laboral del recurso

En el nivel 3 de detalle se muestra el contenido de cada hito. En la siguiente figura se puede observar el conjunto de fases que la componen, temporización y número de horas:

Nombre de tarea	Duración	Comienzo	Fin	Trabajo
<b>▲ PAC1. Plan de trabajo</b>	<b>16 días</b>	<b>18/09/2019</b>	<b>09/10/2019</b>	<b>26 horas</b>
Búsqueda de información	1 día	18/09/2019	19/09/2019	8 horas
Def. del plan de trabajo	2 días	08/10/2019	09/10/2019	15 horas
Rev. del plan de trabajo	0,88 días	09/10/2019	09/10/2019	2 horas
Entrega PAC1	0,88 días	09/10/2019	09/10/2019	1 hora
<b>▲ PAC2. Diseño</b>	<b>15 días</b>	<b>10/10/2019</b>	<b>30/10/2019</b>	<b>82 horas</b>
Diseño BD	0,75 días	10/10/2019	10/10/2019	8 horas
Diseño UML	1 día	13/10/2019	13/10/2019	10 horas
Perfiles de usuario	0,5 días	15/10/2019	15/10/2019	6 horas
Casos de uso	1,5 días	16/10/2019	17/10/2019	8 horas
Diagramas de secuencia	3 días	17/10/2019	19/10/2019	8 horas
Wireframe baja fidelidad	1,38 días	20/10/2019	21/10/2019	9 horas
Wireframe alta fidelidad	2,22 días	23/10/2019	25/10/2019	12 horas
Prototipo	1 día	27/10/2019	27/10/2019	8 horas
Actualización de la memoria	2 días	28/10/2019	29/10/2019	12 horas
Entrega PAC2	1 día	30/10/2019	30/10/2019	1 hora
<b>▲ PAC3. Implementación</b>	<b>30 días</b>	<b>31/10/2019</b>	<b>11/12/2019</b>	<b>178 horas</b>
<b>Etapa 1. Prep. del entorno de trabajo</b>	<b>2 días</b>	<b>31/10/2019</b>	<b>01/11/2019</b>	<b>7 horas</b>
<b>▲ Etapa 2. Back-end</b>	<b>10 días</b>	<b>03/11/2019</b>	<b>17/11/2019</b>	<b>60 horas</b>
Diseño BD	4 días	03/11/2019	06/11/2019	8 horas
Conectividad back-end a servicios	4 días	08/11/2019	13/11/2019	29 horas
Consulta de API's	3 días	14/11/2019	17/11/2019	23 horas
<b>▲ Etapa 3. Front-end</b>	<b>10 días</b>	<b>17/11/2019</b>	<b>29/11/2019</b>	<b>75 horas</b>
Diseño Clases UML	2 días	17/11/2019	19/11/2019	11 horas
Implementación comunicación y funcionalidades con back-end	6,5 días	18/11/2019	26/11/2019	43 horas
Diseño de pantallas	4 días	26/11/2019	29/11/2019	21 horas
<b>▲ Etapa 4. Juego de pruebas</b>	<b>4 días</b>	<b>01/12/2019</b>	<b>05/12/2019</b>	<b>22 horas</b>
Identificación y corrección de errores	5 días	01/12/2019	05/12/2019	22 horas
Actualización de la memoria	3 días	09/12/2019	11/12/2019	13 horas
Entrega PAC3	1 día	11/12/2019	11/12/2019	1 hora
<b>▲ PAC4. Entrega final</b>	<b>17 días</b>	<b>12/12/2019</b>	<b>03/01/2020</b>	<b>48 horas</b>
Elaboración del manual de usuario	3 días	12/12/2019	14/12/2019	10 horas
Realización de la presentación	3 días	16/12/2019	18/12/2019	8 horas
Actualización de la memoria	5 días	23/12/2019	27/12/2019	16 horas
Preparación/Optimización del código fuente	2 días	28/12/2019	30/12/2019	13 horas
Entrega PAC4	1 día	03/01/2020	03/01/2020	1 hora
<b>Defensa del proyecto</b>	<b>5 días</b>	<b>13/01/2020</b>	<b>17/01/2020</b>	<b>1 hora</b>

Figura 3. Nivel 3 Diagrama de Gantt

En consecuencia, el tiempo total dedicado al proyecto asciende a 335 horas en las que se incluyen la defensa del mismo, las cuales se desglosan en:

Horas en días laborables: 144.66 horas

Horas en días festivos: 190.34 horas

### 1.5 Breve resumen de productos obtenidos

El proyecto se centra en el desarrollo de una aplicación móvil que permita adaptar la situación actual de las TIC con nuevas metodologías educativas otorgando una herramienta tanto a docentes como alumnado.

Por un lado, la aplicación permitirá al alumno la creación de un avatar personalizable que irá pudiendo ser mejorado mediante la obtención de experiencia y bonificaciones en relación a los resultados obtenidos en las actividades entregadas. Al mismo tiempo, podrá consultar el ranking de la clase y perfiles de sus compañeros para así observar qué accesorios han añadido a su perfil o cuáles disponen.

El alumnado también dispondrá de un historial de actividad dentro del curso y de su interacción con la aplicación.

Por otro lado, el profesor/a podrá tener un control de la evolución de los alumnos y alumnas, fomentar actividades que potencien la motivación o reforzar conceptos y actitudes. Dispondrá de organización de las diferentes clases y grupos en los que imparta materia, dónde podrá asociar actividades. También podrá crear grupos para realizar actividades competitivas.

El desarrollo de la parte de back-end utiliza el lenguaje de programación para dispositivos móviles Swift. La integración de persistencia, autenticación y almacenamiento de documentos se implementa utilizando la API de Firebase. Destacar que el tipo de base de datos escogido es Cloud Firestore debido a su ajuste con el modelo de documentos y colecciones descrito en el punto [2.2 Base de datos](#).

Además, se ha hecho uso de Google Analytics para recopilar datos de uso y comportamiento de la app, preparando así la aplicación para futuras mejoras al obtener datos de su funcionamiento desde versiones tempranas.

También se han utilizado librerías de terceros para la implementación de controles, en concreto, ScrollableSegmentedControl para la paginación de vista.

La parte de front-end, se desarrolla siguiendo el diseño propuesto en el punto [2.7.2 Wireframe de alta fidelidad](#) con el uso del Storyboard que el IDE ofrece. De esta forma, los elementos se pueden vincular con el back-end y responder a las peticiones que este solicita a la vez que da respuesta a las interacciones de los usuarios/as.

Para facilitar la integración de la aplicación con las cuentas asociadas por parte del instituto, se añade autenticación mediante email y se contempla la complejidad de utilizar un sistema de login mediante google, así centralizar y facilitar el alta de los alumnos.

Con todo lo expuesto, la aplicación deja un amplio camino de expansión, ya que existen diferentes ampliaciones que el proyecto podría presentar en una idea futura, tal como se recogen en el apartado [1.2 Objetivos del Trabajo](#) en los objetivos ampliados.

## **1.6 Breve descripción de los otros capítulos de la memoria**

En este apartado se presenta una posible descripción de los capítulos en los que se basará la memoria del proyecto:

- **Arquitecturas y tecnología:** Presentación de la arquitectura que seguirá el proyecto entorno a cliente-servidor.
- **BBDD:** Diagrama de la base de datos de la aplicación.
- **Diagramas UML:** Diagrama de clases de la aplicación.
- **Perfiles de usuario:** Especificación de las funciones y actividades que puede realizar cada perfil en la app.
- **Casos de Uso:** Casos posibles que contempla la aplicación en base a las funcionalidades descritas.
- **Diagramas de secuencia:** Mostrar cómo interactúa la aplicación con los diferentes sistemas.
- **Diseño:** Elaborar los modelados de las diferentes pantallas de la app, prototipo funcional que permita elaborar un correcto diseño y logotipo de la app.
  - Wireframe de baja fidelidad
  - Wireframe de alta fidelidad
  - Prototipo
  - Logotipo
  - Usabilidad
- **Instalación y preparación del entorno de trabajo:** Descripción de los requerimientos previos a la fase de implementación del proyecto.
  - Cliente
  - Servidor
- **Implementación:** Desarrollo del apartado técnico de la aplicación.
  - Cliente
  - Servidor
  - API's utilizadas
- **Juegos de pruebas / Usabilidad y UX:** Testeo y evaluación de la aplicación.
  - Evaluación de requisitos
  - Evaluación con usuarios
  - Manual de usuario
- **Identificación y corrección de errores:** Mención de los errores detectados durante el desarrollo del proyecto y su posterior resolución.
- **Versiones de la aplicación:** Detalle de las funcionalidades que incluye cada lanzamiento del prototipo. Se determina en base a las diferentes entregas de las PAC hasta su posterior versión completa.
- **Proyección a futuro:** Estudio de la viabilidad y futuro del proyecto fuera de un entorno académico.
- **Conclusiones**
  - Valoración Personal

## 2. Resto de capítulos

### 2.1 Arquitecturas y tecnología

El tipo de arquitectura que utilizará la aplicación es el patrón de diseño MVC: **Modelo-Vista-Controlador**.

La funcionalidad de cada capa del patrón de diseño se puede resumir en:

El **modelo (M)** será la representación de todas las entidades de nuestra aplicación, siendo estas las clases que gestionarán la persistencia de datos.

El **controlador (C)** es la capa intermedia entre el modelo y la vista. Su función es poner en comunicación la vista con el modelo y procesar responder las peticiones en ambos sentidos.

La **vista (V)** se centra en componentes que definiremos cómo la interfaz de la aplicación.

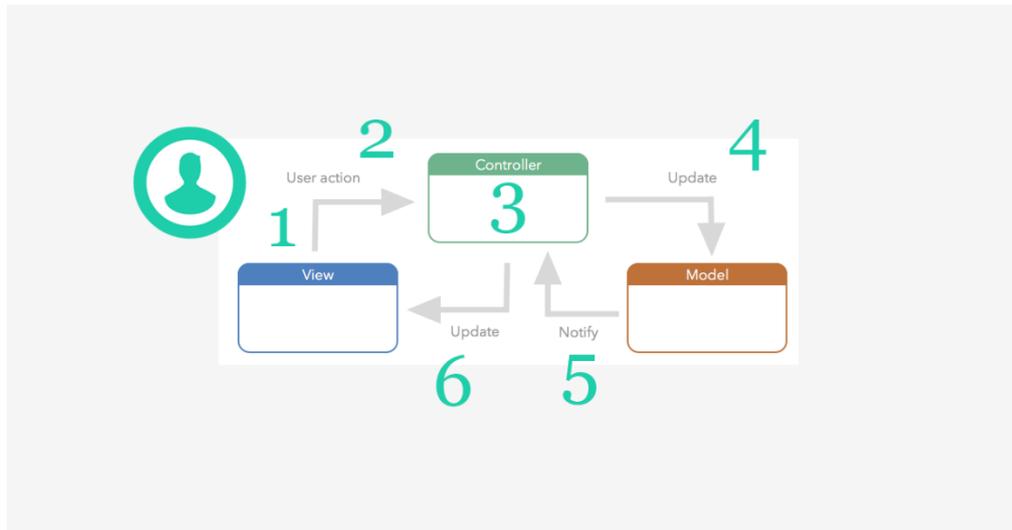


Figura 4. Patrón Modelo-Vista-Controlador

En la imagen se puede ver una representación gráfica de este patrón. A continuación, se detalla a más profundidad cómo trabajará la aplicación con el patrón de diseño:

1. El usuario/a interactúa sobre la interfaz de la app, por ejemplo, pulsando un botón.
2. La view comunica al controlador que el usuario/a ha realizado la opción de pulsar un botón.
3. El controlador ejecuta la lógica de la acción asociada.
4. Si la lógica requiere alguna modificación de datos, esta actualiza el modelo.
5. El modelo notifica de los cambios realizados al controlador.
6. En el caso que el controller haya modificado el modelo, notifica a la view para que muestre los nuevos datos al usuario/a.

## 2.2 Base de datos

La información de diseño de la aplicación está almacenada en una base de datos noSQL, por lo tanto, se basa en un sistema de datos no relacionales y que no necesita seguir un esquema. A su vez, la metodología está orientada a documentos y para facilitar su administración se ha recurrido a la utilización de una interfaz gráfica. Las colecciones que componen la base de datos son las siguientes:

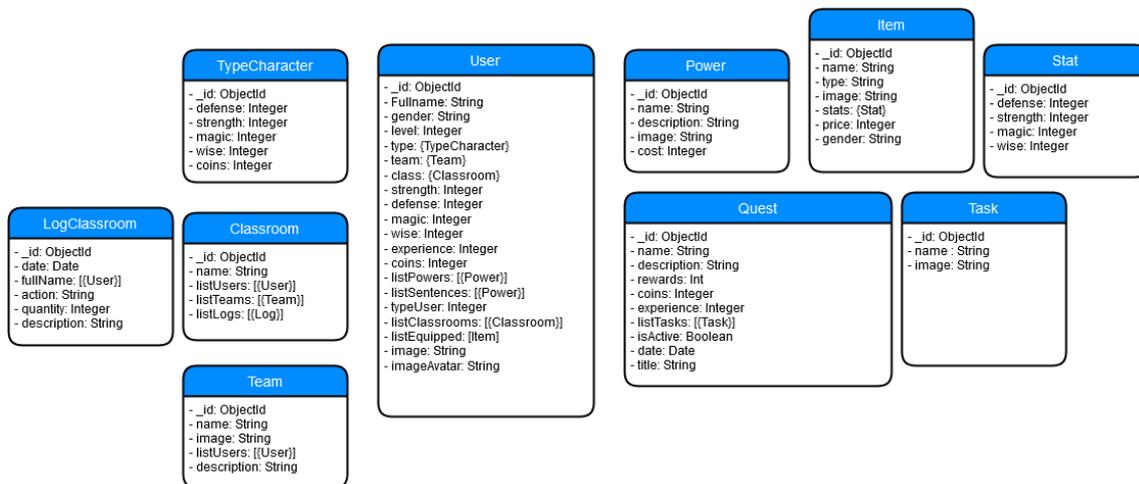


Figura 5. Diseño de la base de datos

### 2.3 Diagrama UML

A continuación, se muestra el diseño UML que seguirá la aplicación. Tal como se muestra en el esquema un usuario dispone de un conjunto de atributos. También dispone de una lista de quest a realizar, las cuáles están compuestas por tareas. Al finalizar una quest, el alumno recibirá una recompensa con la que podrá adquirir nuevos objetos en la tienda y modificar su lista de ítems. El usuario a su vez, dispone de una lista de poderes y sentencias que afectan de una manera u otra a sus atributos.

En esta primera versión, el alumno sólo puede pertenecer a una clase, aunque un profesor puede tener una lista de clases con los correspondientes alumnos en ella. Dentro de las clases se pueden crear equipos para posibles competiciones o actividades en grupo.

Destacar que para implementar una mayor escalabilidad se definen tipos Enum en el tratamiento de atributos que puedan sufrir escalabilidad o interfaces para las funciones de administración.

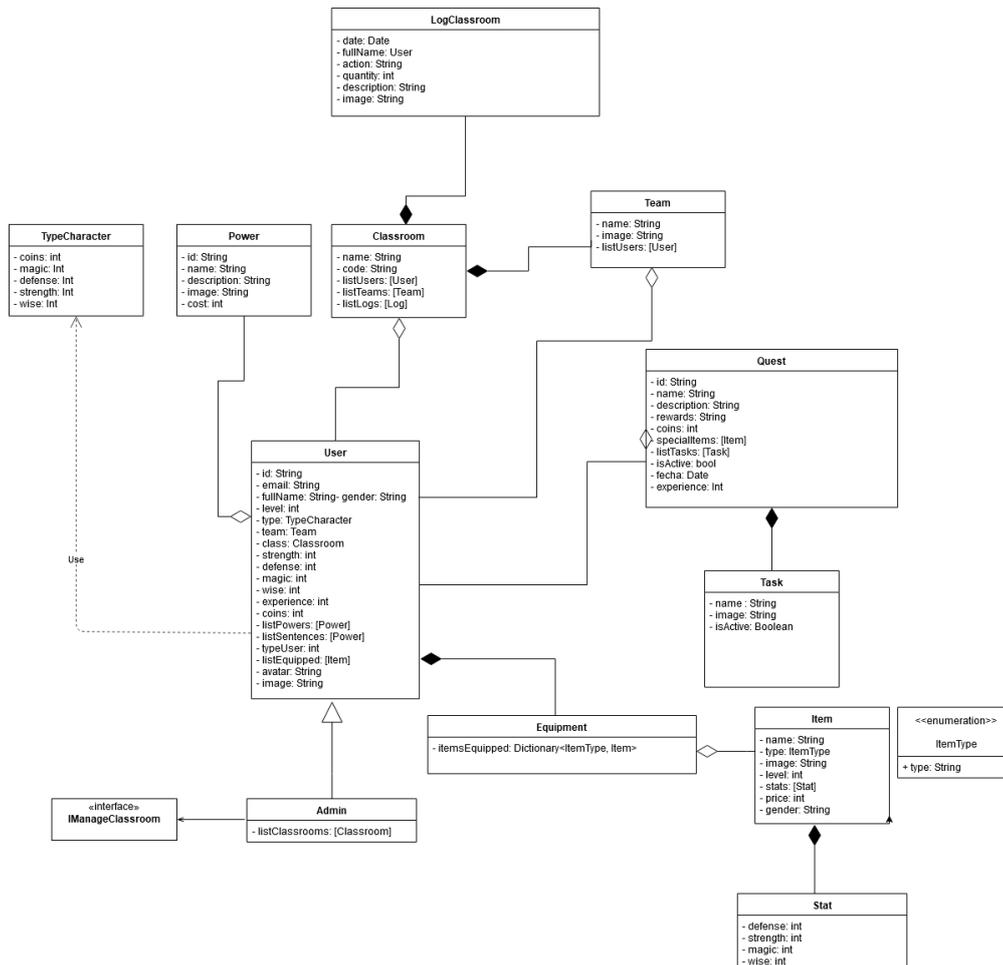


Figura 6. Diseño de clases UML

## 2.4 Perfiles de usuario

Después de un periodo de observación, se establece el público objetivo en alumnado de ciclo formativo de grado medio, concretamente en la familia de informática en ciclos de grado medio de sistemas microinformáticos y redes con edades comprendidas entre los 18 a los 23 años de ambos sexos ubicados en Barcelona con perfil tanto de estudiante que trabaja como estudiante que no trabaja.

Cualquier nacionalidad. Dominio de idioma castellano o catalán. Tipo de público activo, aficionado a las nuevas tendencias tecnológicas y habituado al uso de aplicaciones. Busca un servicio que le aporte algún valor añadido diferenciador. Prioriza el ahorro de tiempo y rapidez. Le motiva sentir control en sus acciones obteniendo los resultados que espera.

## Público Objetivo



Figura 7. Descripción ficha buyer persona

## 2.5 Casos de Uso

Debido a que no se quiere aumentar la complejidad del proyecto en base al tiempo del que se dispone, se especifica que las gestiones de un posible actor super administrador de la aplicación las realiza el actor profesor.

Del mismo modo, se asume que únicamente se define un profesor que gestione las diferentes clases, actividades, habilidades o ítems que ya se establecerán fijadas en la base de datos, quedando a futuras ampliaciones la implementación de estos requerimientos.

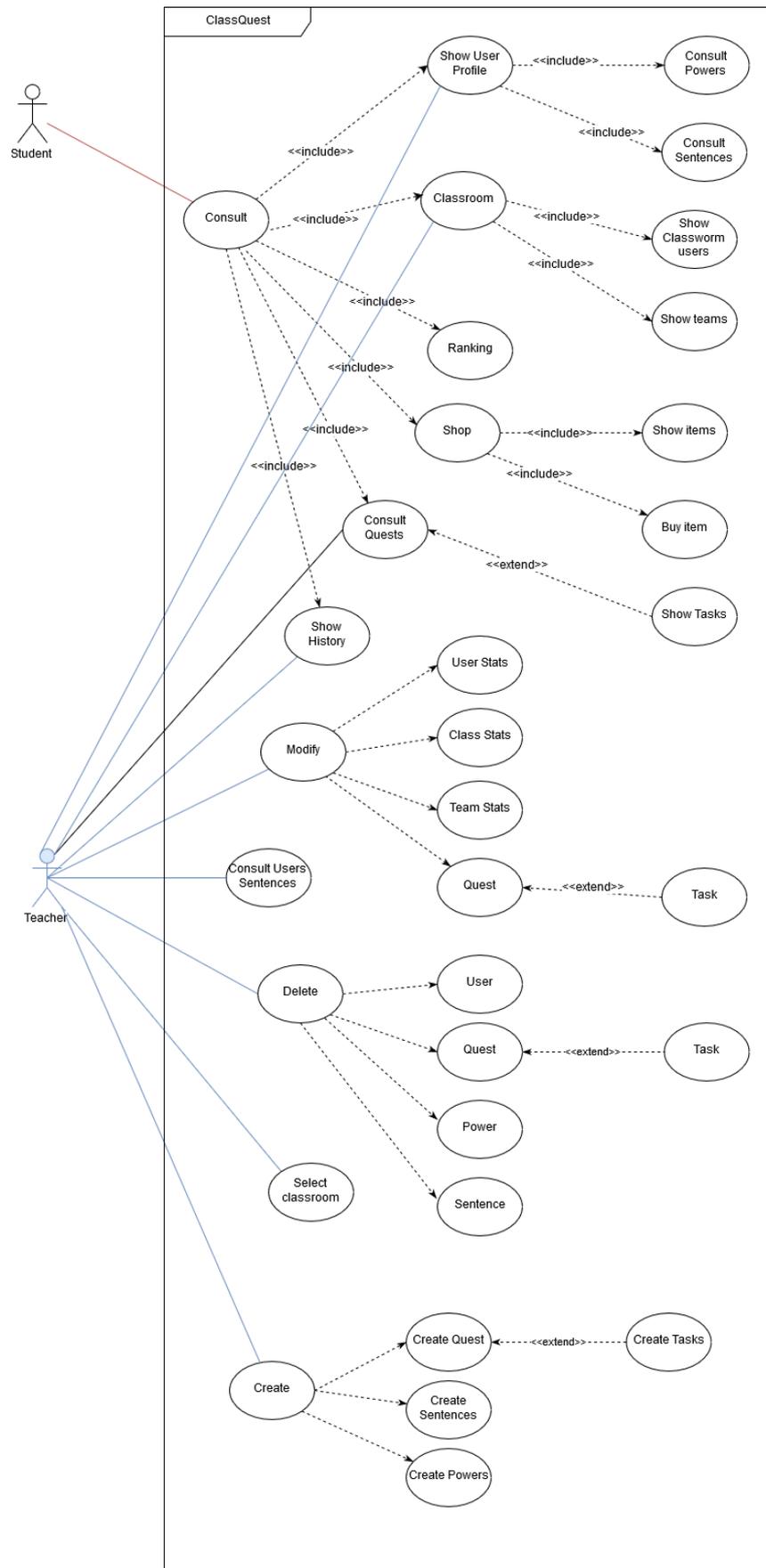


Figura 8. Casos de uso

## 2.6 Diagrama de flujo

Para representar el flujo que experimenta el alumno durante el uso de la aplicación se muestra el diagrama de flujo. Así, se puede determinar el recorrido que se produce en cada interacción con el sistema y las posibles variantes que se pueden tomar en cada pantalla.

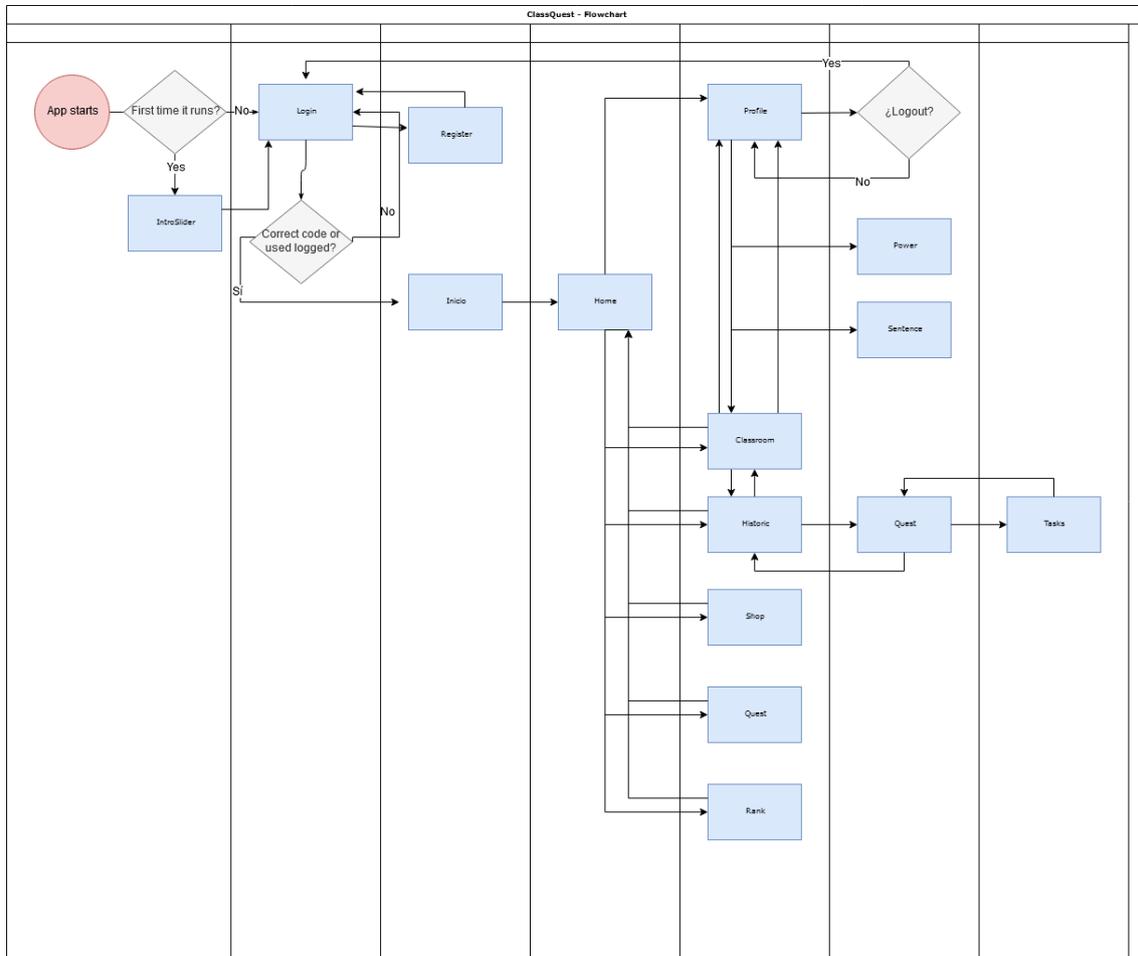


Figura 9. Diagrama de secuencia de la app

## 2.7 Diseño

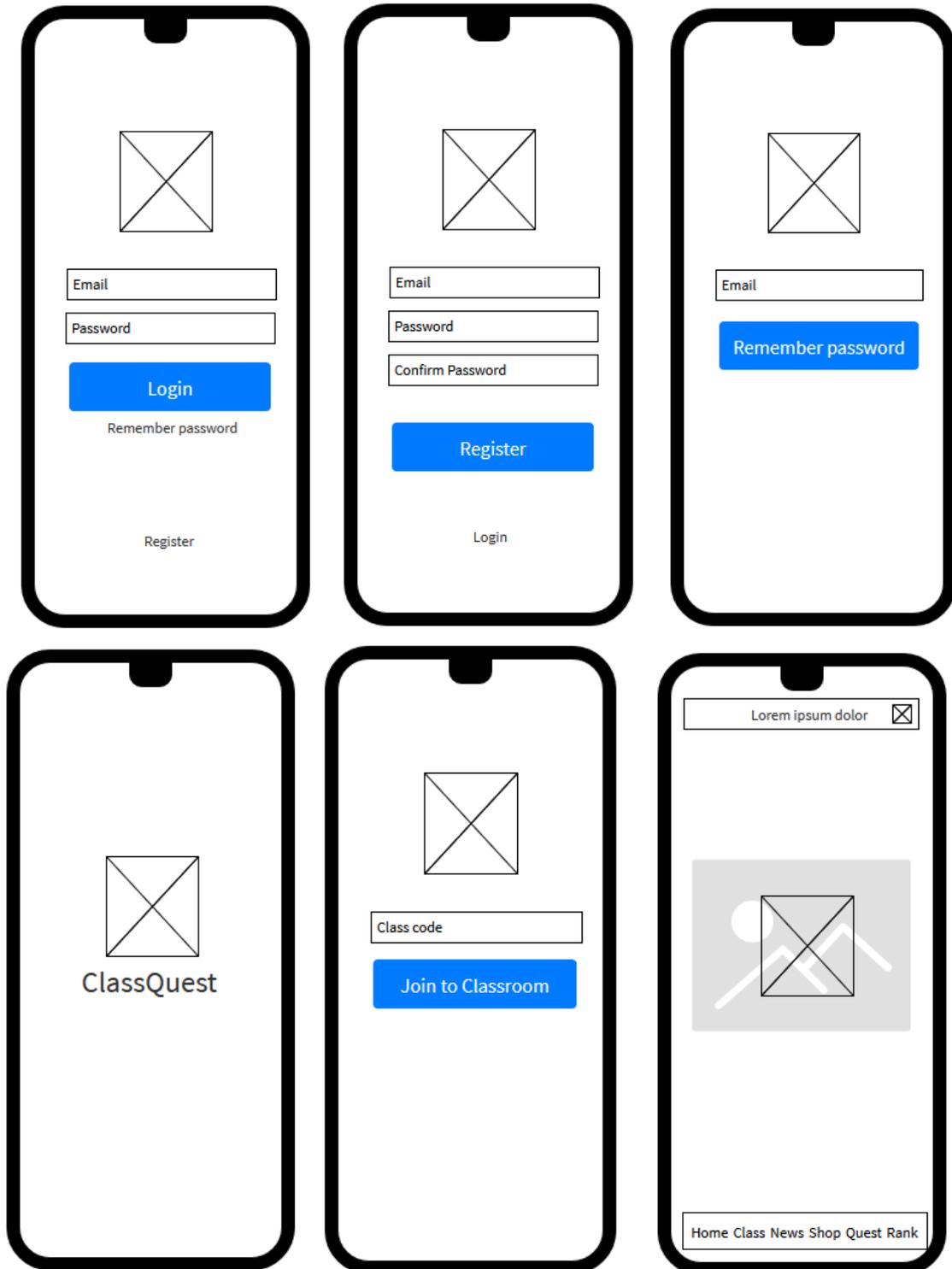
Durante la fase de desarrollo gráfico se esbozan las primeras pantallas de la aplicación y cómo se organiza el contenido en estas. Para ello se maquetan en wireframes de baja fidelidad, los cuáles darán lugar más tarde a los wireframes de alta fidelidad que son los que se utilizarán para la preparación del prototipo.

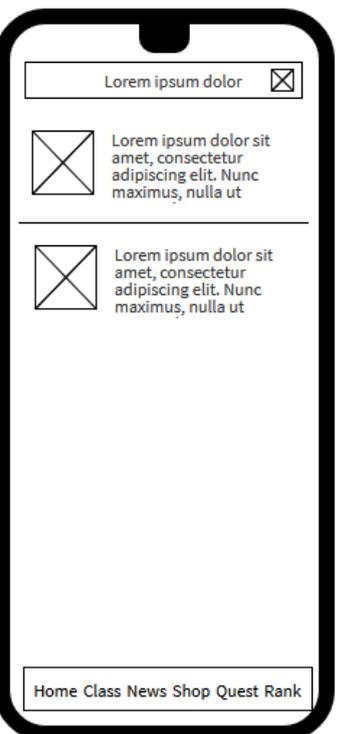
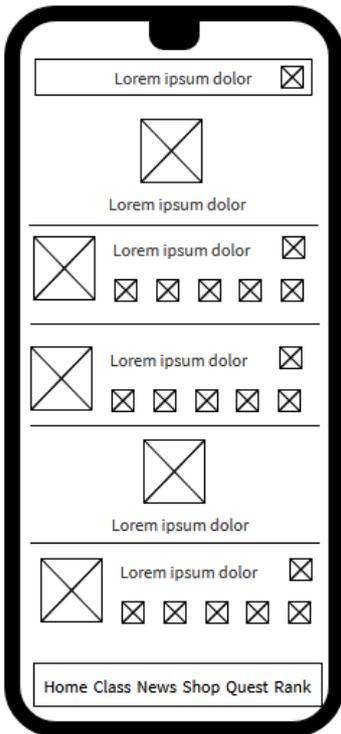
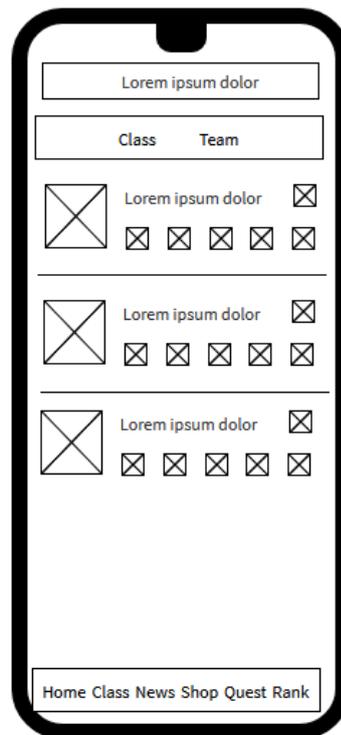
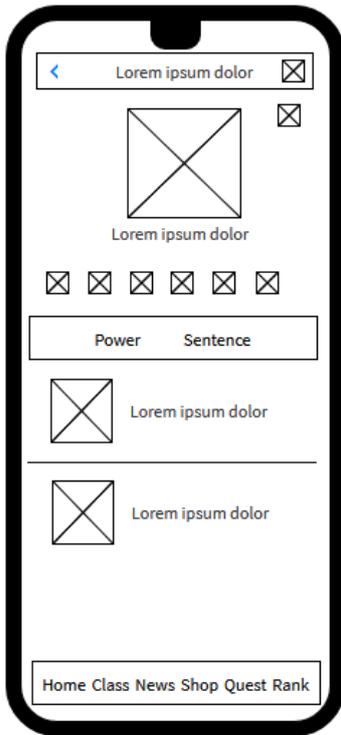
También se presenta el logo y la usabilidad presentada durante el desarrollo del proyecto.

### 2.7.1 Wireframe de baja fidelidad

La realización de los wireframes de baja fidelidad se lleva a cabo mediante el uso de la herramienta mockflow y permite una primera idea del resultado de la

aplicación. Estos wireframes se utilizan como base para crear los wireframes de alta fidelidad explicados en el apartado **2.7.2. Wireframe de alta fidelidad.**





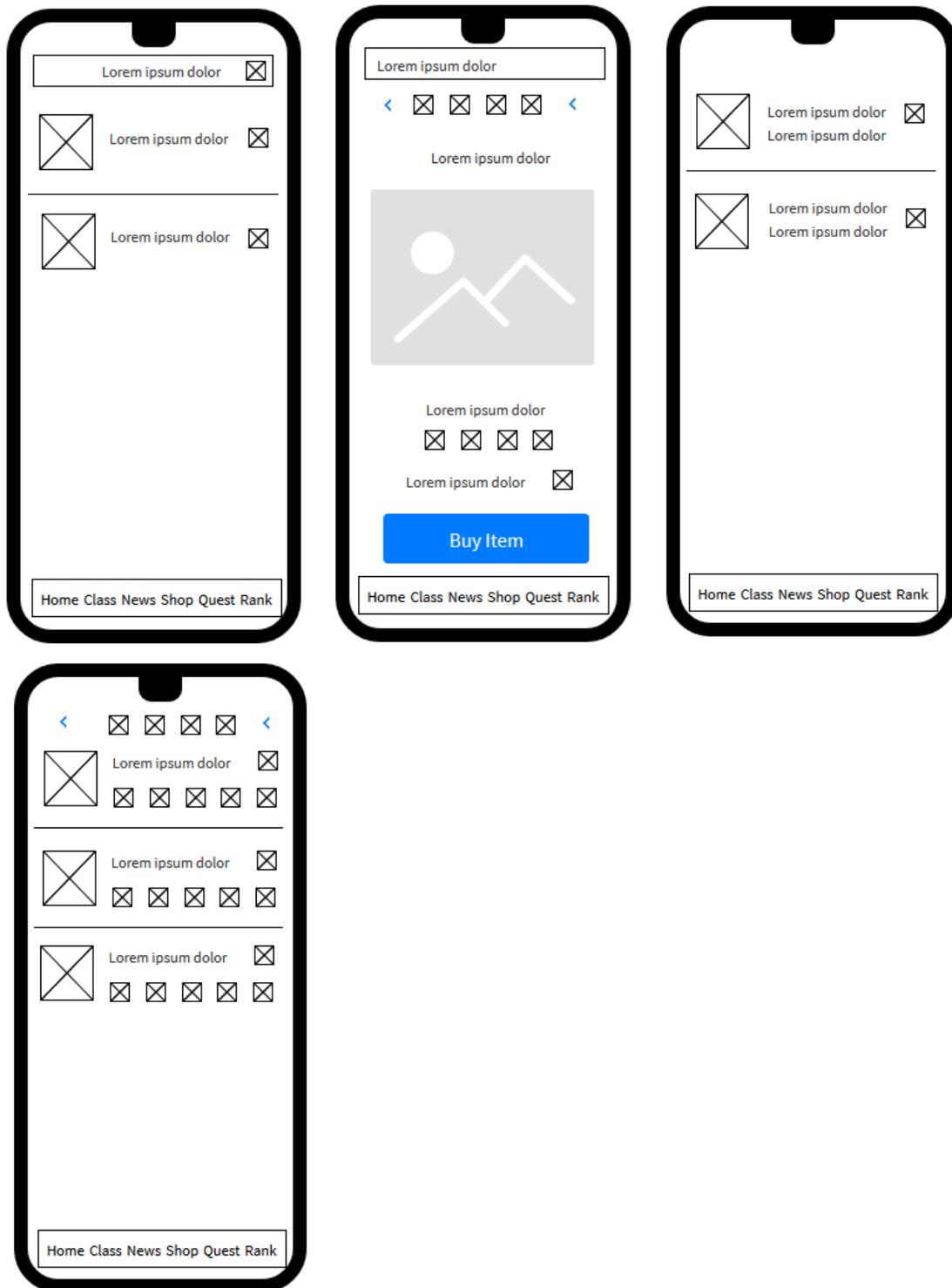
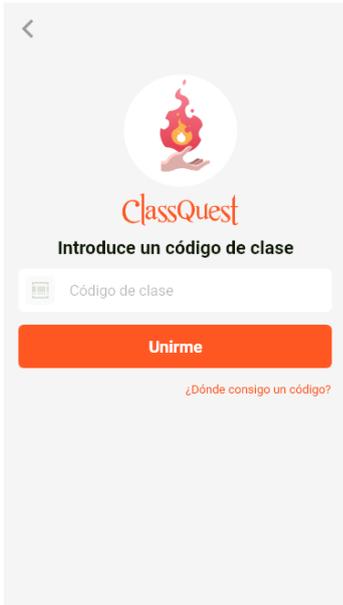


Figura 10. Wireframes de baja fidelidad

## 2.7.2 Wireframe de alta fidelidad

Una vez diseñada la estructura básica de la aplicación se define la interfaz gráfica definitiva, implementando todos los estilos, tipografías, gráficos y colores a utilizar.

Destacar que los wireframes que se muestran a continuación corresponden a la interfaz del alumno.



Perfil



**Destello de luz**

El invocador/a que utilice este poder devuelve 4 puntos de magia a un compañero/a y quién lo usa recibe 1 punto de sabiduría.

Aliado/a **+4**      Invocador/a **+1**

Cerrar sesión

Carmen Moreno Torrubiano



**Sanadora - Nivel 3**  
SMX AB - Los destructores

22 15 47 12 81 3

Poderes      Sentencias

**Destello de luz**

Dispones de 2 punto(s) de poder para gastar

Cerrar sesión

Carmen Moreno Torrubiano



Cerrar sesión

Carmen Moreno Torrubiano



**Sanadora - Nivel 3**  
SMX AB - Los destructores

22 15 47 12 81 3

Poderes      Sentencias

Estás libre de sentencias.

Cerrar sesión

Manel Sanz Torrent



**Guerrero - Nivel 2**  
SMX AB - Los destructores

44 33 24 2 10 2

Poderes      Sentencias

**Furia de fuego**

Cerrar sesión

Manel Sanz Torrent



**Guerrero - Nivel 2**  
SMX AB - Los destructores

44 33 24 2 10 2

Poderes      Sentencias

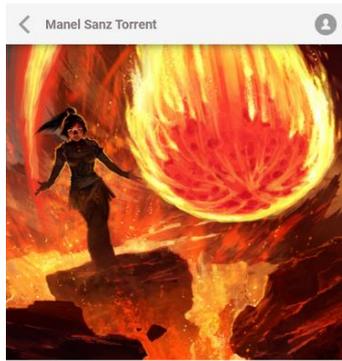
No tienes poderes para usar

Dispones de 1 punto(s) de poder para gastar

Cerrar sesión

Carmen Moreno Torrubiano

Clase	Equipo
 <b>SMX AB - Los destructores</b>	
 Carmen Moreno Torrubiano  3 Fuerza: 22 Defensa: 15 Poder mágico: 47	 Manel Sanz Torrent  2 Fuerza: 44 Defensa: 33 Poder mágico: 24



**Furia de fuego**

Tus acciones te han llevado a tener una penalización que afecta a todos los puntos de fuerza. Reduce tu fuerza -10 puntos.

**Sentencia**

 -10

Habla con el profesor/a para saber cómo librarte de la sentencia.

Ranking

< Fuerza **Defensa** Poder mágico >

 Carmen Moreno Torrubiano  3 Fuerza: 22 Defensa: 15 Poder mágico: 47	12  81
 Manel Sanz Torrent  2 Fuerza: 44 Defensa: 33 Poder mágico: 24	2  10

Quest

 "El terror de los atributos" 24/10/2019 - 10:00 Activa <b>Completada</b> 
 "Introducción a etiquetas" 22/10/2019 - 13:00 Inactiva <b>Incompleta</b> 

Quest



**El terror de los atributos**  
 Consigue superar esta misión y obtendrás:

Premios  
 +10

Estado: **Incompleta**

[Ver tareas](#)

El terror de los atributos

 Activitat1. HTML Calificació: 6.5 
 Activitat2. HTML i CSS Calificació: Sin calificación 

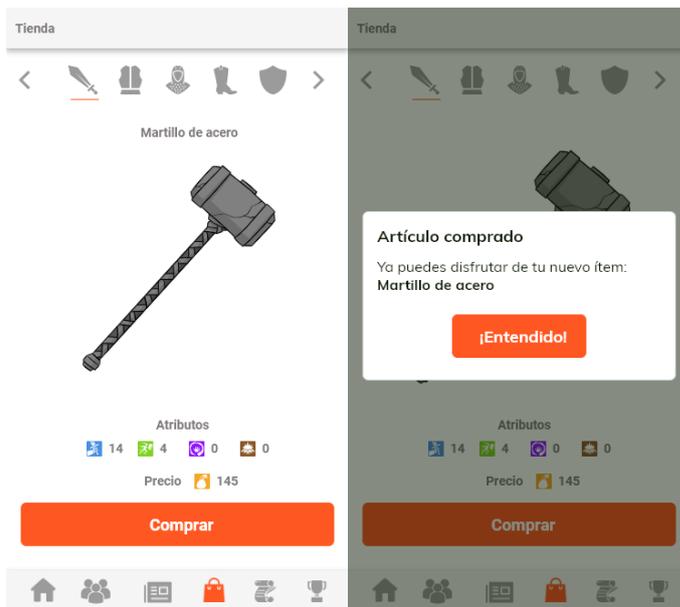


Figura 11. Wireframes de alta fidelidad

### 2.7.3 Prototipo

En el siguiente enlace se puede ver el prototipo funcional de la aplicación utilizando los wireframes de alta fidelidad.

<https://xd.adobe.com/view/ffd580b2-5495-4bb6-448e-ceb8da73fb46-89e4/>

### 2.7.4 Logotipo

Debido a que la aplicación está vinculada a generar interés en los estudiantes, se presenta un logo que identifique a un juego clásico rpg. Se escoge el diseño flat design por su simplicidad. A su vez se utilizará como icono de la app.



Figura 12. Logotipo de la app

### 2.7.5 Usabilidad

En el proceso de diseño de la interfaz se ha mantenido en constante pensamiento que el uso pueda ser llevado fácilmente por un alumno que posee ciertos conocimientos de informática, pero a su vez se está iniciando en el uso de las nuevas tecnologías. Por ello se destacan algunos valores que se han intentado respetar:

- La interfaz se ha diseñado con la intención de que resulte intuitiva y fácil de utilizar, minimizando la curva del aprendizaje.
- Los iconos se presentan con uniformidad y acompañan al texto para que resulte más sencilla su identificación.
- Se mantiene una coherencia entre las pantallas utilizando un diseño plano y colores visualmente agradables.
- Las fuentes son suficientemente grandes para aportar una fácil legibilidad en las pantallas de los actuales dispositivos móviles o tabletas.
- Se distinguen cabecera y cuerpo claramente en el diseño, utilizando la misma composición para las diferentes pantallas. De este modo, otorgando comodidad y sensación de conocer que información se encontrará debido a la coherencia de la estructura.
- Se utiliza un sistema de ventanas emergentes durante las interacciones del alumno, facilitando el feedback y la sensación de que el alumno domina la situación, conociendo en todo momento en la fase que se encuentra durante el proceso que esté realizando.
- Se informa de posibles errores o fallos durante la introducción de datos o modificación.

## 2.8 Instalación y preparación del entorno de trabajo

En esta fase se especifican los requerimientos que se deben tener en cuenta en la preparación previa al desarrollo del proyecto.

### 2.8.1 Cliente

Esta capa utiliza:

- **xCode**: IDE que permite la codificación del back-end y la creación del flujo de pantallas de la aplicación.
- **Simulador de dispositivo**: Emulación de un terminal iOS. Para las pruebas y realización de pantallas se ha utilizado un Iphone 8 Plus.
- **CocoaPods**: Gestor de dependencias que permite la integración de estas en la aplicación, permitiendo el posterior uso de la librería.

### 2.8.2 Servidor

Se define por:

- **Swift**: Lenguaje de programación orientado a objetos que permite la codificación de la aplicación.

- **Firestore:** Librería que permite gestionar la persistencia de datos, autenticación de usuarios/as, análisis de uso y almacenamiento de ficheros. A continuación, se detallan los componentes utilizados en la infraestructura.
  - **Cloud Firestore:** Almacenamiento y sincronización de datos mediante una base de datos noSQL.
  - **Authentication:** Administración de usuarios/as ofreciendo diversos métodos de autenticación.
  - **Google Analytics:** Recopila datos de uso y comportamiento de la aplicación.
  - **Cloud Storage:** Almacenamiento de contenido: imágenes, audio y vídeo.
- **ScrollableSegmentedControl:** Librería que permite emular el comportamiento del elemento UISegmentedControl aplicando un aspecto personalizado.

## 2.9 Implementación

### 2.9.1 Cliente

En primer lugar, se presenta la organización del árbol del proyecto y su funcionalidad dentro de la aplicación.

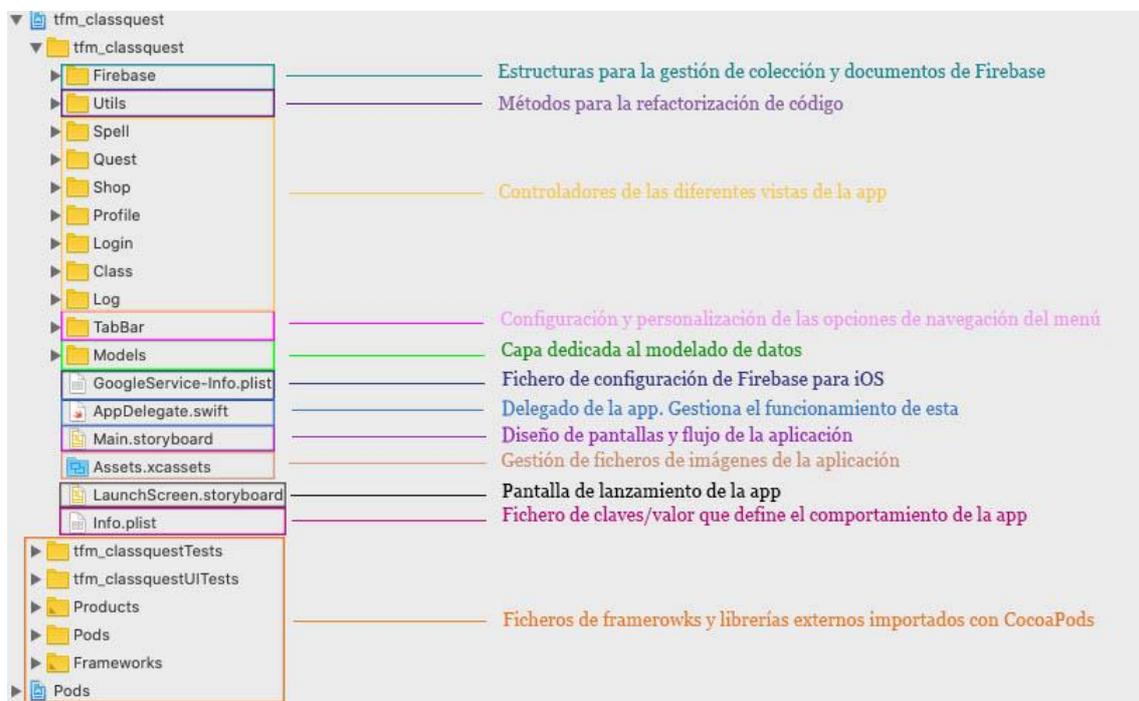


Figura 13. Organización del proyecto

La arquitectura presentada, implementa la preparación previa que se detalla en el punto [2.8 Instalación y preparación del entorno de trabajo](#) contemplado la configuración de cliente y servidor.

#### 2.9.1.1 CocoaPods

En primer lugar, para realizar la integración librerías en el proyecto, se ha gestionado haciendo uso de de CocoaPods. Este gestor de dependencias para el IDE Xcode nos permite agregar frameworks a nuestros proyectos gestionando todo directamente en un fichero PodFile.

La instalación en el sistema se realiza con el comando:

```
$ sudo gem install cocoapods
```

A partir de aquí, se puede proceder a iniciar el pod con el comando:

```
$ pod init
```

La ejecución de este comando, crea en el directorio del proyecto el documento donde se integran los nombres de los frameworks a utilizar. La configuración del fichero PodFile de este proyecto se detalla a continuación:

```
# Uncomment the next line to define a global platform for your project
# platform :ios, '13.0'

target 'tfm_classquest' do
  # Comment the next line if you don't want to use dynamic frameworks
  use_frameworks!

  # add the Firebase pod for Google Analytics

  pod 'Firebase/Analytics'
  pod 'Firebase/Auth'

  pod 'Firebase/Core'
  pod 'Firebase/Firestore'

  # Optionally, include the Swift extensions if you're using Swift.
  pod 'FirebaseFirestoreSwift'

  pod 'ScrollableSegmentedControl', '~> 1.5.0'

end
```

Finalmente, la integración de las dependencias de los frameworks descritas en el fichero se integran en el proyecto con el comando:

```
$ pod install
```

### 2.9.1.2 Firebase

La gestión de la API de Firebase permite el control de la autenticación, almacenamiento y persistencia de datos del proyecto. Para que la app sea reconocida y se integre con Firebase se han de cumplir unos requisitos:

- Requisitos mínimos:
  - Xcode 10.1 o una versión más reciente.
  - CocoaPods 1.4.0 o una versión más reciente.
  - iOS 8 o una versión más reciente.
  - Swift 3.0 o una versión más reciente.
  - Dispositivo físico o virtualizado.
  - Cuenta de Firebase console.

Una vez el entorno de trabajo cumple los requisitos, se puede proceder a realizar los pasos previos para la integración del SDK de Firebase en el proyecto:

1. Crear un proyecto de Firebase para iOS en el panel de Firebase console.
2. Registrar la app con Firebase
  - 2.1. Agregar una nueva app en el panel de Firebase.
3. Ingresar el ID del paquete de la app, en este caso: **com.dmorenoar.tfm-classquest**
  - 3.1. Información de la app (opcional)
  - 3.2. Registrar la app.
4. Agregar el archivo de configuración de Firebase "GoogleService-Info.plist" al directorio raíz del proyecto.
5. Agregar el SDK de Firebase a la app usando CocoaPods, indicando en el fichero:

```
pod 'Firebase/Analytics'
```

6. Inicializar Firebase en la app
  - 6.1. Importando el módulo de Firebase en la clase UIApplicationDelegate:

```
import Firebase
```

- 6.2. Configurar una instancia compartida de FirebaseApp en en el método `application:didFinishLaunchingWithOptions` de la clase UIApplicationDelegate:

```
FirebaseApp.configure()
```

7. Finalmente, en las clases que vayan a hacer uso de Firebase realizar el import de la librería en la cabecera:

```
import Firebase
```

Los siguientes puntos tienen en común que para su correcta integración, el proyecto debe contemplar realizar la importación del módulo Firebase y configuración de FirebaseApp del fichero UIApplicationDelegate.

## Cloud Firestore

En primer lugar, se deben agregar al fichero podFile los siguientes pod, teniendo en cuenta de agregar la extensión de Swift en el caso de que la aplicación esté utilizando este lenguaje.

Como anotación para entender el sentido de la importación de **FirestoreSwift**, informar que también se pueden programar aplicaciones en iOS haciendo uso del lenguaje Objective-C en lugar de Swift:

```
pod 'Firebase/Core'  
pod 'Firebase/Firestore'  
  
# Opcionalmente, incluimos la extensión de Swift debido al lenguaje de la app  
pod 'FirestoreSwift'
```

La inicialización de la instancia de cloud firestore se realiza añadiendo a la clase UIApplicationDelegate() la siguiente línea:

```
let db = Firestore.firestore()
```

A continuación, se detalla el CRUD, siendo este el acrónimo de "Crear, Leer, Actualizar y Borrar" (del original en inglés: Create, Read, Update and Delete).

La estructura de los datos y cómo estos se almacenan, se detalla en el punto [2.2 Base de datos](#).

Para **agregar (C)** datos, cabe destacar que no es necesario crear las colecciones o documentos previamente, al igual que el identificador, en el caso de no ser requerido. De esta forma, podemos proceder a crear un nuevo documento:

```
var ref: DocumentReference? = nil  
ref = db.collection("users").addDocument(data: [  
    "name": "Dani Moreno",  
    "email": "dmorenoar@uoc.edu"  
]) { err in  
    if let err = err {  
        print("Error añadiendo el documento: \(err)")  
    } else {  
        print("Documento añadido con el ID: \(ref!.documentID)")  
    }  
}
```

La **lectura (R)** de datos se puede realizar haciendo uso del método get, con el que se puede recuperar una colección o un documento previamente creado. En el siguiente ejemplo, se muestra cómo se recupera la colección creada en el punto anterior:

```
db.collection("users").getDocuments() { (querySnapshot, err) in  
    if let err = err {
```

```

    print("Error obteniendo el documento: \${err}")
  } else {
    for document in querySnapshot!.documents {
      print("\${document.documentID} => \${document.data()}")
    }
  }
}

```

En la obtención de datos, también se pueden realizar consultas que devuelvan como valor varios documentos que cumplan una condición determinada. De esta forma, mediante el uso de la cláusula `where()` se pueden aplicar condiciones en la lectura de datos:

```

db.collection("tasks").whereField("isActive", isEqualTo: true)
  .getDocuments() { (querySnapshot, err) in
    if let err = err {
      print("Error obteniendo el documento: \${err}")
    } else {
      for document in querySnapshot!.documents {
        print("\${document.documentID} => \${document.data()}")
      }
    }
  }
}

```

La **modificación (U)** de colecciones o documentos se puede realizar haciendo uso del método `setData()` o `updateData()`.

En la utilización del método `setData()`, destacar que durante la ejecución de la consulta, si el documento no existe se creará. De lo contrario, si el documento existe, su contenido se reemplazará con los datos proporcionados.

```

db.collection("quest").document("tasks").setData([
  "name": "La maldición de los tags",
  "isActive": "true",
  "date": Timestamp(date: Date())
]) { err in
  if let err = err {
    print("Error obteniendo el documento: \${err}")
  } else {
    print("Documento actualizado correctamente")
  }
}

```

Haciendo uso del método `updateData()`, se pueden actualizar algunos campos del documento sin reemplazo por completo:

```

db.collection("quest").document("tasks").updateData([
  "name": "La maldición de los atributos 2",

```

```

]) { err in
  if let err = err {
    print("Error modificando el documemnto: \(err)")
  } else {
    print("Documento actualizado correctamente")
  }
}

```

La **eliminación (D)** de documentos se realiza mediante el método delete().

```

db.collection("users").document("tom@gmail.com").delete() { err in
  if let err = err {
    print("Error modificando el documento: \(err)")
  } else {
    print("Documento eliminado correctamente")
  }
}

```

Del mismo modo que eliminamos un documento, también se pueden eliminar campos de estos.

```

db.collection("users").document("tom@gmail.com ").updateData([
  "name": FieldValue.delete(),
]) { err in
  if let err = err {
    print("Error borrando el campo del documento: \(err)")
  } else {
    print("Campo del documento actualizado correctamente")
  }
}

```

Para la eliminación de colecciones, Firebase no recomienda su utilización debido a que colecciones con grandes cantidades de documentos podrían generar errores de memoria insuficiente, afectando al rendimiento de la app. De esta forma, se recomienda realizar la eliminación de colecciones en un entorno de servidor de confianza.

## Authentication

Para poder realizar autenticación con usuarios se han de añadir las siguientes dependencias en el fichero Podfile:

```

pod 'Firebase/Auth'

```

Una vez inicializado el SDK de Firebase se puede proceder a realizar el registro de nuevos usuarios/as mediante correo electrónico y contraseña, recogiendo los datos de registro. La aplicación, al mismo tiempo que realiza el registro crea un documento en la colección Users.

Haciendo uso del método `createUser` se puede realizar el alta:

```
Auth.auth().createUser(withEmail: email, password: password) { authResult, error in
  // ...
}
```

El inicio de sesión de usuarios registrados con anterioridad se realiza mediante el correo electrónico y contraseña. El método `signIn()` gestiona si el login ha tenido éxito:

```
Auth.auth().signIn(withEmail: email, password: password) { [weak self] authResult, error in
  guard let strongSelf = self else { return }
  // ...
}
```

Durante el uso de la aplicación, en determinadas pantallas se requiere hacer uso de la información del usuario/a que ha iniciado sesión. Se puede establecer un objeto como método de escucha que controle el estado de la autenticación en la carga de las diferentes vistas, concretamente en el método `viewWillAppear()`:

```
handle = Auth.auth().addStateDidChangeListener { (auth, user) in
  if let user = user {
    // The user's ID, unique to the Firebase project.
    let name = user.name
    let email = user.email
    let profileImage = user.profileImage
    // ...
  }
}
```

También, se puede gestionar la desvinculación del objeto haciendo uso del método `viewWillDisappear()` en el controlador de la vista:

```
Auth.auth().removeStateDidChangeListener(handle!)
```

## Cloud Storage

Para permitir que la aplicación contemple el almacenamiento se debe añadir al fichero `Podfile` el siguiente pod y proceder a la integración tal como se comenta en el punto [2.9.1.1CocoaPods](#).

```
pod 'Firebase/Storage'
```

Posteriormente, se debe incluir en el fichero `UIAppDelegate`, concretamente en la función `application:didFinishLaunchingWithOptions`, la siguiente línea:

```
let storageRef = Storage.storage()
```

A continuación, para realizar la subida de un archivo se debe crear una referencia que permite establecer la ubicación donde se ubicará el archivo.

```
let imagesRef = storageRef.child("imagenes")
```

Posteriormente, se puede proceder a subir ficheros. Las siguientes líneas indican un ejemplo:

```
// Datos en memoria
let data = Data()

// Creamos una referencia al directorio que queremos guardar las imágenes
let imagenesRef = storageRef.child("imagenes")

// Realizamos la subida del fichero
let uploadTask = imagenesRef.putData(data, metadata: nil) { (metadata, error) in
  guard let metadata = metadata else {
    // Gestionamos si encontramos un error
    return
  }

  // Metadata contiene metadatos de archivo como tamaño, tipo de contenido..
  let size = metadata.size

  // También se puede acceder a la URL de descarga después de la subida
  imagenesRef.downloadURL { (url, error) in
    guard let downloadURL = url else {
      // Gestionamos si encontramos un error
      return
    }
  }
}
```

## Google Analytics

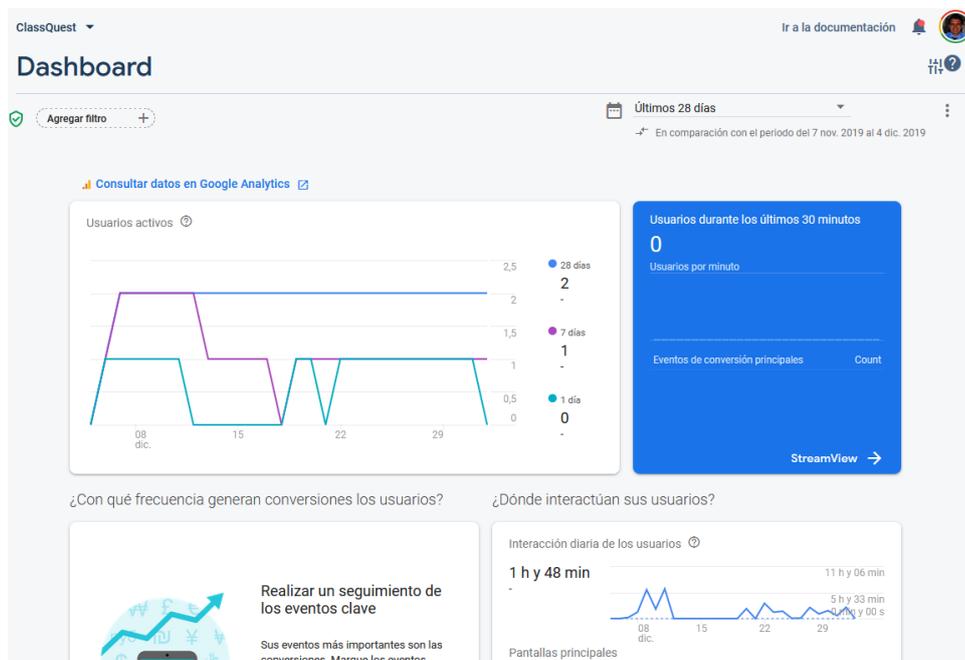
Debido a la envergadura del proyecto, no se entra en profundidad en analizar el comportamiento y uso de la aplicación, pero se muestra las gráficas recogidas durante el desarrollo de la misma y se establece como una proyección a futuro.

Se ha de pensar que gestionar un análisis de la aplicación puede permitir observar el comportamiento que realizan los usuarios/as de esta y establecer estrategias que potencien todavía más la aplicación y su uso.

Para empezar a trabajar y controlar estos datos se ha de incluir en el PodFile el siguiente pod:

```
pod 'Firebase/Analytics'
```

Finalmente, al realizar la integración en el proyecto se puede empezar a consultar los datos y uso de la app.

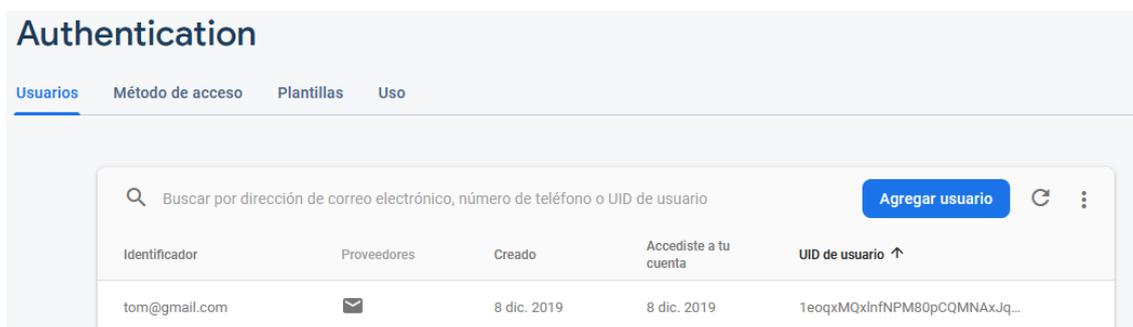


## 2.1.1 Servidor

El siguiente apartado se comenta como se recibe y gestiona la información en la administración de Firebase Console, siendo esta la que ofrece los servicios los cuáles hace uso la aplicación.

### 2.1.1.1 Autenticación

La autenticación implementa los métodos de acceso mediante cuentas de correo electrónico. En su versión inicial se activa la autenticación mediante correo electrónico y en versiones posteriores se contempla la autenticación mediante Google.



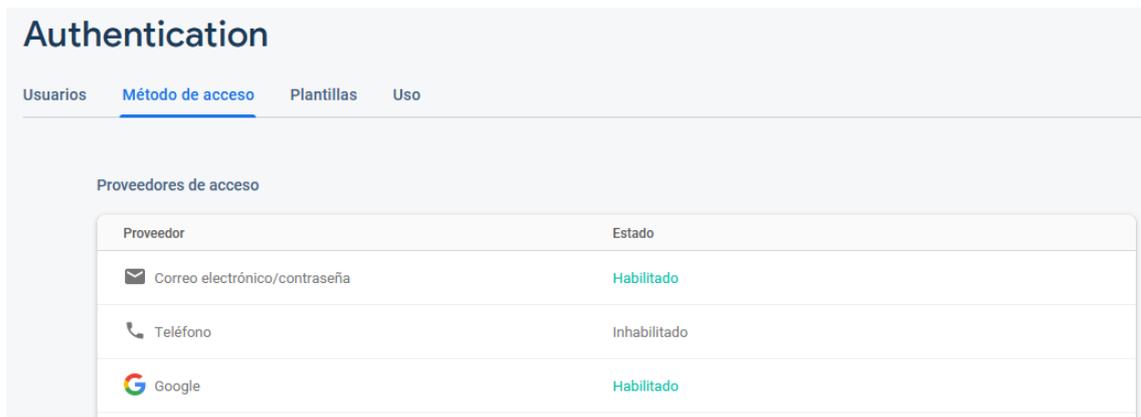


Figura 16. Métodos de acceso de la app

### 2.1.1.2 Persistencia

La app gestiona la persistencia haciendo uso de Firebase, concretamente el tipo de base de datos Cloud Firestore. Se ha optado por hacer uso de este tipo de base de datos en lugar de Realtime Database debido a que el planteamiento de la arquitectura de la base de datos se orienta a colecciones de documentos, en lugar de un árbol JSON como hasta entonces ofrecía Firebase.

De esta forma, Cloud Firestore se beneficia de todas las ventajas que ofrece Realtime Database a la vez que aporta un modelo de datos más simple y permite que los datos complejos y jerárquicos de la app sean más simples de tratar, adaptándose de una forma más natural a ClassQuest. Esto permite que ciertas colecciones como, por ejemplo, la colección User, integren dentro de sí misma un conjunto amplio de subcolecciones dentro de documentos.

### 2.1.1.3 Almacenamiento

En primer lugar, se debe habilitar en Firebase Console la función Storage y las políticas de seguridad de la app. A partir de este punto, ya se puede proceder a gestionar el almacenamiento.

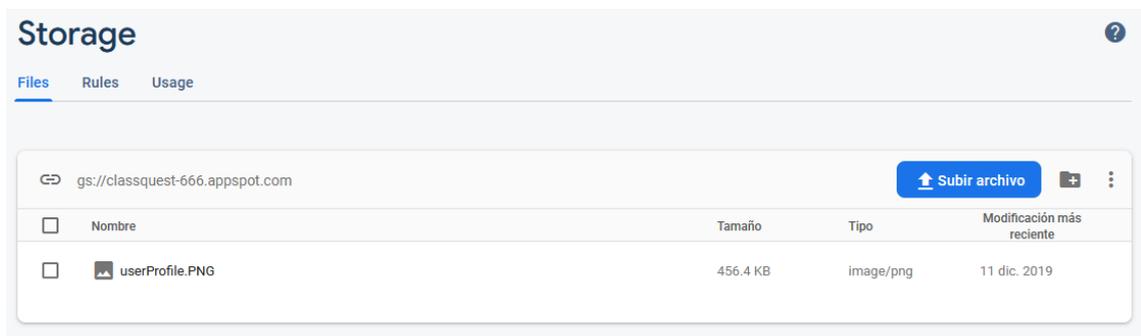


Figura 17. Gestor de ficheros Cloud Storage

## 2.1.2 API's utilizadas

Se utiliza la API de Firebase para beneficiarse de los diferentes servicios que ofrece: autenticación, almacenamiento, persistencia, etc. De esta forma, el conjunto de funcionalidades permite configurar y compilar la app estableciendo seguridad y escalabilidad. La configuración del SDK se detalla en el punto [2.8 Instalación y preparación del entorno de trabajo](#) y posteriormente, en el punto 2.9 Implementación la utilización y implementación de estos servicios en el proyecto.

Para realizar una fácil gestión de las dependencias del proyecto, se ha utilizado CocoaPods como gestor de contenido. De esta forma, se consigue un control de las librerías o frameworks en un único fichero de configuración.

En relación al aspecto visual, se ha utilizado el repositorio de la librería ScrollableSegmentedControl que permite configurar y diseñar modificaciones de elementos nativos presentes en Swift como el segmentedControl. Así, obviar el uso de este componente y gestionar las mismas funcionalidades con la librería de terceros.

Finalmente, la realización del testing en la interfaz de usuario de la app han sido realizadas con el framework XCUITest. De este modo, poder interactuar con la aplicación realizando test propios e incluso generar test automáticos.

## 2.2 Juegos de pruebas / Usabilidad y UX

### 2.2.1 Evaluación de requisitos

Su objetivo es comprobar que la plataforma que se desarrolla cumpla todos los requisitos especificados en el proyecto y a su vez que estos funcionan de la forma esperada. Para evaluar estos se procede a realizar el siguiente juego de pruebas:

#### **Configuración del SDK y comunicación con la API de Firebase**

Para realizar esta evaluación se procede a crear un proyecto en la plataforma Firebase Console. Una vez creada la aplicación y realizadas las importaciones que se detallan en el punto [2.9.1.2 Firebase](#) para establecer la configuración del SDK, se descarga el fichero GoogleService-Info.plist y se inicia el test de verificación de comunicación de la API. Una respuesta positiva por parte de la API, valida la creación del proyecto en el servidor y el uso de los servicios.

#### **Registro de usuarios/as y autenticación**

Se realiza la configuración de métodos de acceso a la aplicación estableciendo el login con correo electrónico en el panel de Authentication. A posterior, se realizar un registro de alumno/a indicando su correo electrónico y contraseña, verificando la creación del usuario en el panel de Authentication de Fireabase Console.

La autenticación se lleva a cabo ingresando en la pantalla de Login el correo electrónico y contraseña previamente creadas, verificando que el método de signIn no informa de error y verifica la cuenta del alumno/a.

## Funcionalidades CRUD

La realización de funcionalidades CRUD se lleva a cabo haciendo uso de los servicios de persistencia de Firebase tal cómo se detalla en el punto [2.9.1.2 Firebase](#).

### Correcto funcionamiento de avisos del sistema

Utilizando la clase UIAlertController, se establecen los diferentes mensajes que darán retroacción a las acciones del usuario/a en la aplicación.

Se realizan diferentes acciones que muestren estos mensajes, por ejemplo:

- Autenticación con un correo electrónico no registrado.
- Registro con campos vacíos.
- Adquirir un ítem sin monedas suficientes.

### Compra de artículos y asignación al avatar

Se realiza una compra de un artículo en la tienda y se procede a comprobar que en Firebase Storage el usuario dispone de la imagen del artículo. A su vez, se comprueba que el usuario almacene en la base de datos el ítem seleccionado perteneciendo al tipo de ítem. Finalmente, se comprueba mostrando la pantalla de perfil del usuario y comprobando visualmente que el ítem se le ha asociado al avatar.

## 2.2.2 Evaluación con usuarios

La evaluación de la aplicación se lleva a cabo haciendo uso de dos tipos de test: evaluación heurística de usuarios y testeo con el framework XCUITest.

Por un lado, a parte de la evaluación de requisitos del punto anterior, se realizan de test con el framework XCUITest permitiendo desarrollar pruebas de la interfaz de usuario (UI) emulando la interacción de los usuarios con la aplicación. Con ello, se establecen un conjunto de pruebas de la UI automatizadas.

En este punto, se realizan dos test:

- Login correcto y acceso al perfil
- Login correcto e intento de compra de un ítem previamente comprado.

El primer test, realiza un acceso con un usuario válido y a continuación, accede a la información de su perfil.

```
func testValidLoginSuccessTapSettings(){  
  
    let validEmail = "dani@gmail.com"  
    let validPassword = "123456"  
  
    let app = XCUIApplication()
```

```

app.launch()

let usernameTextField = app.textFields["Email"]
XCTAssert(usernameTextField.exists)

usernameTextField.tap()

usernameTextField.typeText(validEmail)

let passwordTextField = app.secureTextFields["Contraseña"]
XCTAssert(passwordTextField.exists)

passwordTextField.tap()
passwordTextField.typeText(validPassword)

app.children(matching: .window).element(boundBy: 0).
  children(matching: .other).element.
  children(matching: .other).element.
  children(matching: .other).element.
  children(matching: .other).element.tap()

app.buttons["Entrar"].tap()
app.buttons["userSettings"].tap()
}

```

El segundo test, realiza un acceso a la aplicación, se dirige a la pestaña de tienda y realiza la compra de un ítem.

```

func testValidLoginSuccessErrorBuyingItem(){

  let validEmail = "dani@gmail.com"
  let validPassword = "123456"

  let app = XCUIApplication()
  app.launch()

  let usernameTextField = app.textFields["Email"]
  XCTAssert(usernameTextField.exists)

  usernameTextField.tap()

  usernameTextField.typeText(validEmail)

  let passwordTextField = app.secureTextFields["Contraseña"]

```

```

XCTAssert(passwordTextField.exists)

passwordTextField.tap()
passwordTextField.typeText(validPassword)

app.children(matching: .window).element(boundBy: 0).
children(matching: .other).
element.children(matching: .other).
element.children(matching: .other).
element.children(matching: .other).
element.children(matching: .other).element.children(matching: .other).element.tap()

app.buttons["Entrar"].tap()

app.tabBars.children(matching: .button).element(boundBy: 3).tap()
app.tables.cells.containing(.button, identifier:"Comprar").
children(matching: .button).matching(identifier: "Comprar").
element(boundBy: 1).tap()

app.alerts["Ítem ya comprado"] .scrollViews.otherElements.
buttons["¡Entendido!"] .tap()
}

```

Finalmente, podemos observar el resultado positivo de la ejecución de los test:

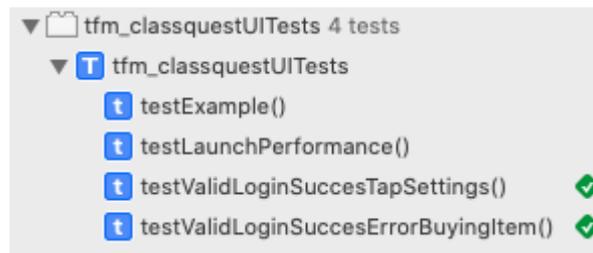


Figura 18. Resultado ejecución XCUITest

Por otro lado, la realización de los test con usuarios se usarán test de usabilidad basándose en los principios que establece Jakob Nielsen y Rolf Molich [10] para detectar de una forma rápida problemas en la usabilidad. Para ello, se hace uso de la plantilla elaborada por Deniese Pierotti, de Xerox Corporation [11].

A modo resumen, las observaciones de los resultados del test arrojan buenos resultados respecto a la sensación que experimenta el usuario dentro de la aplicación. Se puede observar que el entorno le resulta agradable, el uso de colores que resaltan los elementos activos, iconos activos o inactivos y la información que se muestra en las diferentes pantallas se mantienen coherente durante el uso de la aplicación.

También, se destaca una buena retroacción ante una acción haciendo uso las alertas.

El test indica, que la aplicación no distingue entre usuarios expertos o novatos, ni facilita configuración de idiomas. Del mismo modo, no ofrece atajos de teclado para interacción o personalización de las opciones para el usuario.

Los resultados completos del test con usuarios se pueden consultar en el documento [Anexo 1. Test con usuarios.pdf](#)

### **2.2.3 Manual de usuario**

#### **¿Qué es ClassQuest?**

ClassQuest, es una app que permite al estudiante crear su propio personaje e ir mejorando los objetos que este dispone durante la realización de las actividades de clase. Estas actividades, otorgarán recompensas que podrán ser canjeadas con monedas en la tienda por nuevos objetos o facilitarán poderes al usuario que podrá utilizar durante el desarrollo del curso.

La aplicación muestra las tareas disponibles en el módulo, y las tareas asociadas a estas, además de las recompensas.

Del mismo modo, que el usuario puede obtener poderes, el profesor/a también le puede aplicar sentencias por un comportamiento incorrecto durante el curso. Estas sentencias le aplicarán alguna penalización y podrá librarse de ella realizando las tareas que el profesor/a le proponga.

A su vez, el alumnado podrá observar el avance del resto de compañeros de clase y observar que objetos han comprado para su personaje. Todo esto, puede controlarse mediante el registro de actividad de la clase.

#### **¿Cómo compilar ClassQuest?**

##### **Xcode:**

Para abrir la aplicación en Xcode realizar doble clic en el fichero `tfm_classquest.xcworkspace`.

Proceder a seleccionar el emulador, el utilizado en las pruebas ha sido iPhone 8 Plus, o si se dispone de un dispositivo físico seleccionarlo en la lista.

##### **Dispositivo móvil (Requiere cuenta de Apple Developer).**

En el caso de disponer de una cuenta de Apple Developer, los pasos son:

Para realizar la instalación de un fichero `.ipa` en un dispositivo móvil seguir el siguiente tutorial:

1. Instalar iTunes en el ordenador.
2. Hacer doble clic en el fichero `.ipa`.

3. En la categoría “Ajustes” de iTunes hacer clic en “Aplicaciones”
4. En el recuadro de la aplicación, presionar el botón “Instalar”
5. Cuando finalice la instalación, se podrá ejecutar la aplicación en el dispositivo móvil.

### ¿Cómo administrar ClassQuest?

La administración de la app se realiza utilizando la consola de [Firebase](#). Las opciones que se requieren conocer para el funcionamiento se detallan en el punto [2.9.1.2. Firebase.](#)

### ¿Cómo utilizar la aplicación?

A continuación, se detalla el funcionamiento de las diferentes ventanas de la aplicación y su uso.

#### 2.2.3.1 Iniciar sesión

En primer lugar, para realizar el inicio de sesión, el usuario debe haberse registrado en la aplicación, de lo contrario no se podrá acceder.

La pantalla de inicio de sesión permite:

- **(1) Recordar contraseña:** en el caso de no recordar la contraseña, esta se puede restablecer mediante el reenvío de un correo electrónico de recuperación.
- **(2) Registro:** si el usuario no dispone de cuenta, esta opción le permite crear una.
- **(3) Iniciar sesión:** Si el usuario dispone de cuenta, puede introducir su email y contraseña para entrar en la aplicación.



Figura 19. Iniciar sesión

En el caso de que la contraseña no sea correcta, el usuario no exista o algún campo esté vacío, la app notificará al usuario con un mensaje de alerta.

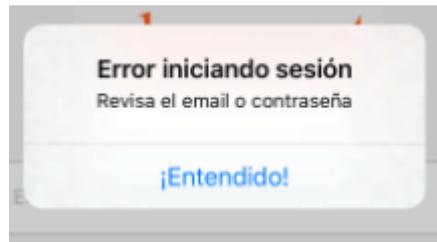


Figura 20. Error de inicio

### 2.2.3.2 Registro

El registro permite que un usuario forme parte de la base de datos de la aplicación y pueda hacer uso de sus funcionalidades. Para crear una cuenta, en la ventana principal se debe realizar un clic en la opción “No tienes cuenta? Regístrate ahora”.

En la siguiente pantalla el usuario debe escribir su nombre, escoger el género de su personaje y la raza. Únicamente se puede escoger un género y una raza, de lo contrario el sistema informa con una ventana de error, al igual que si se deja un campo vacío.



Figura 21. Registro de usuario 1/2

Según la raza escogida el personaje disfrutará de unas estadísticas para los valores: fuerza, defensa, magia, sabiduría y monedas.

Raza	Fuerza	Defensa	Magia	Sabiduría	Monedas
Elfo/Elfa	10	10	18	25	50
Mago/Maga	5	5	50	35	100
Guerrero/Guerrera	60	35	3	10	80

Tabla 3. Estadísticas de personaje

En realizar clic en Siguiente, se podrece a rellenar el campo email y contraseña para acceder al sistema.

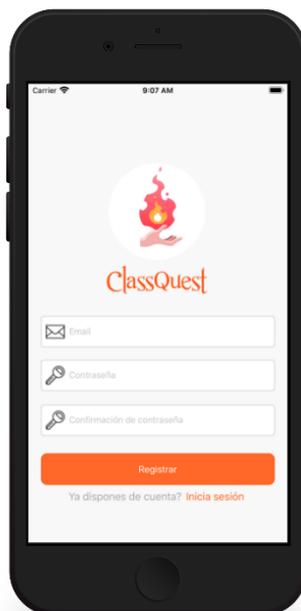


Figura 22. Registro de usuario 2/2

El email se utilizará para iniciar sesión, y los campos de contraseña deben coincidir. En caso de que algún campo este vacío, existe el email en la base de datos o las contraseñas no coincidan se muestra un mensaje de advertencia.

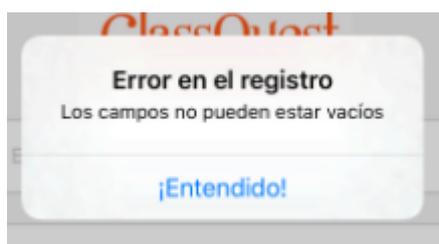


Figura 23. Error de registro

Al finalizar el registro, se redirige al usuario a la página de Inicio de sesión para que pueda introducir su correo electrónico y contraseña definidas en el paso anterior. En este punto, dado que el usuario previamente no dispone de una clase asociada para empezar a participar, debe solicitar al profesor el código de asignación de la clase.



Figura 24. Unión a la clase

Una vez introducido un código correcto, de lo contrario se informa con un mensaje de error, el usuario quedará inscrito a la clase y podrá empezar a configurar su perfil.

### 2.2.3.3 Perfil

La página de perfil muestra al usuario su personaje con los ítems equipados que adquirió previamente en la tienda.



Figura 25. Perfil de usuario

En esta pantalla, se puede acceder a:

- **(1) Detalle del perfil:** El usuario podrá consultar sus estadísticas, lista de poderes, sentencias, clase a la que pertenece y grupo.
- **(2) Perfil:** En el caso de que se encuentre en otra opción, haciendo clic en el icono de la casa puede volver a su perfil.
- **(3) Listado de clase:** Esta opción muestra el listado de compañeros/as de la clase y los grupos, si es que los hay.
- **(4) Registro de actividad:** Esta opción muestra un registro de la actividad que han llevado a cabo los usuarios con la aplicación.
- **(5) Tienda:** En esta opción, el usuario puede adquirir nuevos objetos para su personaje.
- **(6) Listado de Quest:** Se muestra una lista de las tareas, activas o inactivas, y la recompensa asociada a estas. Las tareas incluyen el listado de tareas que las componen.

El punto **(1) Detalle del perfil**, se muestra a continuación:



Figura 26. Detalle del perfil

En esta pantalla, se puede cerrar sesión (1), consultar listado de poderes (2), consultar listado de sentencias (3) o directamente consultar en que consiste el poder o sentencia eligiendo de la lista el que se quiera consultar (4).

Si el usuario escoge la opción (1), será redirigido a la pantalla de inicio de sesión. En el caso de escoger la opción 4, se le mostrará la página que detalla el poder o sentencia escogido:



Figura 27. Detalle de poder/sentencia

En esta pantalla se especifica en que consiste o beneficia dicho poder o sentencia y además se muestra el coste de uso.

### 2.2.3.4 Listado de clase

En este punto se muestran los usuarios que forman parte de la clase del usuario que ha iniciado sesión (1), y también los grupos formados en clase con los alumnos que tiene cada grupo (2).

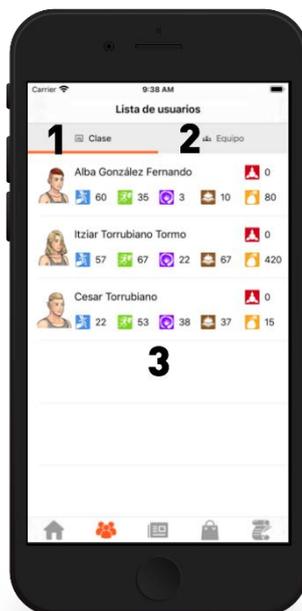


Figura 28. Listado de classe

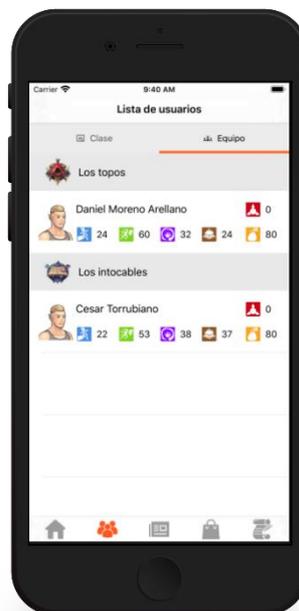


Figura 29. Listado de equipos

El punto (3) permite que si se selecciona un usuario de la lista se pueda acceder a su perfil para consultar los objetos que ha comprado y consultar el detalle de su perfil para ver sus estadísticas o lista de poderes y sentencias.

### 2.2.3.5 Registro de actividad

El registro de la actividad informa a los usuarios de la actividad que ha sucedido en la clase, ya sea la compra de un nuevo objeto o si se ha añadido un nuevo usuario a la clase.

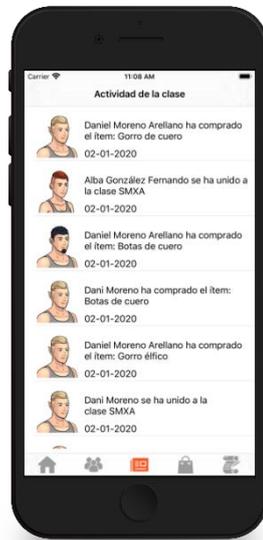


Figura 30. Registro de actividad

### 2.2.3.6 Tienda

En este apartado, se muestran las diferentes categorías de las que se disponen objetos a comprar. Al realizar una selección en la barra superior en el icono que identifica cada ítem se puede acceder a la lista de objetos disponibles para esta categoría. La lista permite un desplazamiento horizontal dónde se muestran más categorías de objetos. En total existen seis categorías: armaduras, botas, cascos, pantalones, escudos y espadas.

Además, en la barra de navegación se le informa al usuario de la cantidad de monedas disponibles para la compra.

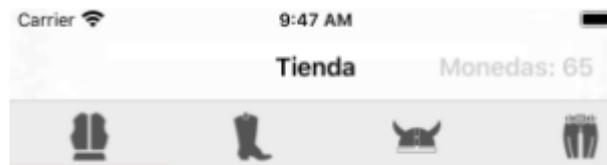


Figura 31. Categorías de objetos

Una vez seleccionada una categoría se muestra el listado de objetos disponibles. Para listar los objetos se puede deslizar la pantalla con un movimiento horizontal.



Figura 32. Tienda

Los objetos muestran las diferentes propiedades que otorgan y su coste. Estas propiedades permiten al usuario incrementar los valores que previamente ha adquirido por el tipo de raza.

Para realizar la compra de un artículo se debe presionar el botón "Comprar". En este momento, se le pregunta al usuario una alerta para que confirme la compra:



Figura 33. Confirmación de compra

En caso positivo se procede a comprobar que dispone de las monedas suficientes para realizar la compra y si es así se le notifica que ya puede disfrutar de su nuevo ítem con un mensaje:

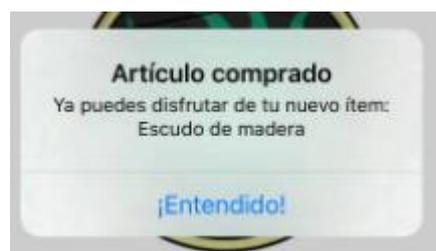


Figura 34. Artículo comprado

En el caso de que el personaje lleve equipado el ítem se le muestra un mensaje informando que ya dispone de este objeto:

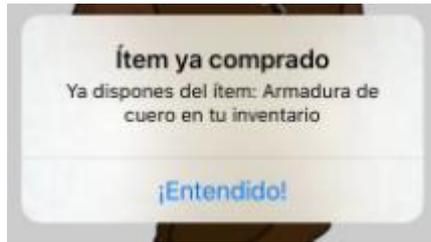


Figura 35. Objeto ya comprado

### 2.2.3.7 Listado de Quests

En esta pestaña se muestran las actividades disponibles en el aula, fecha de creación y si se encuentran activas o inactivas.

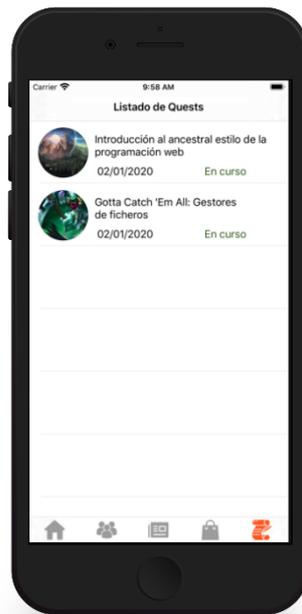


Figura 36. Listado de Quests

Al realizar una selección de una de las actividades se muestra la información de la misma:



Figura 37. Información de la quest

La información detalla la descripción de la actividad, la recompensa en monedas y experiencia. En esta pantalla se dispone del botón “Ver Tareas” que muestra una lista de las tareas que la componen:



Figura 38. Listado de tareas

### 2.3 Identificación y corrección de errores

En esta fase se procede a explicar las acciones correctivas llevadas a cabo en base a los errores o dificultades detectadas durante la implementación o evaluaciones de la aplicación con el objetivo de conseguir una versión completamente funcional del sistema.

## **Incompatibilidad de la versión de xCode con la API de Firebase**

Debido a una incompatibilidad de los requisitos de la API con la versión máxima actualizable del SO High Sierra para xCode, se procedió a actualizar a Catalina 10.15.1. De este modo, se permite instalar la versión 11.2.1 de xCode y cumplir los requisitos previos.

## **Limitación de pestañas en la barra de navegación**

En la ejecución de la fase 2 en el diseño del prototipo se diseñó la barra de navegación con un total de seis elementos. Durante la fase de implementación, se detectó que Swift no permite incorporar 6 elementos en la barra de menús haciendo uso de la clase UITabBarController. Dado que la implementación de un sexto elemento, realizaba cambios visuales demasiado significativos (generaba un desplegable de opciones) se decide no implementar la funcionalidad de ranking y para ello limitar en el listado de clase la visualización de los atributos de los usuarios y grupos.

## **Puntos de poder en las sentencias y poderes**

En la versión del prototipo se proponía usar puntos mágicos para el consumo de poderes/sentencias o adquisición de estos. Después de realizar, el test con usuarios se pudo ver que todo esto se podía gestionar directamente con las monedas del usuario, de esta manera concentrar el progreso de los usuarios/as en un solo recurso.

## **Sentencias y poderes**

En el planteamiento inicial, las sentencias y poderes combinaban la interacción entre los compañeros, pero más tarde durante la realización de la evaluación con usuarios se rediseño el sistema para focalizar estas sentencias y poderes en una interacción con el profesor y el modo en que se desarrolla la clase. De este modo, el profesor se presenta como un elemento más activo en el aula y la relación con los estudiantes. Además, este planteamiento también permite que se pueda mantener la relación entre iguales, pudiendo establecer poderes o sentencias que requieran la colaboración entre el alumnado.

## **2.4 Versiones de la aplicación**

### **2.4.1 Versión Beta**

Esta versión corresponde a la etapa inicial de la aplicación:

- **Autenticación y registro:** Implementación del sistema de inicio de sesión y registro.
- **Reestructuración del código y arquitectura del proyecto:** Primera refactorización del código y jerarquía del proyecto para facilitar su desarrollo.
- **Primera versión de la base de datos:** Codificación de las colecciones y documentos ejemplo para las pruebas CRUD.
- **Sistema de notificaciones:** Integración de sistema de alertas para retroacción de los usuarios/as.

- **Prototipado de la aplicación:** Diseño y codificación de las pantallas y flujo de la aplicación.

#### 2.4.2 Versión release

Esta versión corresponde a la etapa final de la aplicación en la que se incluyen todos los objetivos del proyecto, a excepción del ranking:

- **Reenvío de contraseña:** Implementación del sistema recordar contraseña.
- **Reestructuración del código y arquitectura del proyecto:** Refactorización final del código y jerarquía del proyecto para facilitar su desarrollo.
- **Versión final de la base de datos:** Codificación de las colecciones y documentos ejemplo para las pruebas CRUD.
- **Versión final de la jerarquía de persistencia:** Creación del árbol de organización de ficheros en Firebase Storage.
- **Registro con raza:** Creación del registro escogiendo raza para el personaje y asociación de características de este al usuario.
- **Detección y corrección de errores:** Se añaden las correcciones de todos los errores detectados.
- **Manual de usuario:** Se entrega el manual de uso de la app y se detalla el proceso para su compilación y tratamiento de dependencias.

#### 2.5 Proyección a futuro

##### **Creación de una administración web para que el profesor se aísle de dominar Firebase.**

En esta versión, la app únicamente ofrece gestión mediante la consola de Firebase. Este factor, hace que los profesores deban tener conocimientos de esta para realizar la gestión del aula.

Esta ampliación, propone gestionar una administración web que facilite la administración de la app y pueda extender su uso a otros ámbitos de la docencia fuera del perfil informático.

##### **Gestión de retos**

Esta mejora podría aportar a la aplicación movilidad, es decir, la utilización de códigos QR y la lectura por parte de la app, podría plantear que el alumnado tenga que descubrir ciertos códigos que le pueden reportar: nuevos ítems, poderes o incluso información para resolver actividades.

Otro punto importante, es la utilización en excursiones o actividades del centro. Acostumbra a darse que las faltas de asistencia se convierten en un punto difícil de tratar. Así, ofrecer un sistema de objetos especiales durante el desarrollo de la salida, podría fomentar la asistencia o participación en la actividad.

##### **Combinación con otros módulos y profesores/as.**

En esta versión, la app únicamente puede trabajar con una clase, aunque se contempla a nivel de estructura de base de datos el incorporar diferentes aulas.

La incorporación de esta mejora, permitiría que el alumnado utilice sus personajes en diferentes materias y así poder realizar tareas conjuntas con otros profesores, aplicar penalizaciones o que los alumnos vayan obteniendo mayor número de recompensas.

### **Multiplataforma**

Gestionar la aplicación en diferentes dispositivos permitiría extender el uso dentro del aula. De este modo, una implementación para otros SO cómo Android, facilitaría la utilización de los propios dispositivos del alumnado.

### **Multiidioma**

Otro punto a mejorar que debería llevarse a cabo en la app, especialmente importante en expandir su uso, es incorporar la posibilidad de escoger un idioma con el que presentar la información, ya que todos los textos se han codificado en castellano.

### **Diseño adaptativo**

Cómo suele ser habitual con las versiones de dispositivos y sistemas operativos de los terminales Apple, las aplicaciones deben acomodarse visualmente a las diferentes pantallas o funciones específicas del SO. Por ello, uno de los puntos que se puede tomar como ampliación es realizar la adaptación visual de la app para diferentes dispositivos más allá del iPhone Plus 8 con el que ha sido diseñada.

### **Copias de la base de datos**

Al mismo tiempo, se podría considerar la realización de copias de seguridad de la información de la BD en Firebase, permitiendo así poder recuperar, en caso de pérdida, el mayor volumen posible de datos almacenados.

## **3. Conclusiones**

Una vez concluido el proyecto es el momento de realizar un análisis de todo el trabajo llevado a cabo durante su desarrollo.

Para empezar, este proyecto me ha resultado apasionante y enriquecedor tanto en la parte tecnológica cómo en la personal. Si bien es cierto que en cada fase se han dado problemas que han requerido mi total atención, con esfuerzo y constancia se han conseguido ir resolviendo satisfactoriamente.

El planteamiento inicial de utilizar las tecnologías Firebase era totalmente desconocido por mi parte, del mismo modo que lo era su integración con Swift. Aunque se han requerido muchas horas de aprendizaje y dedicación he disfrutado gratamente.

Construir este proyecto desde sus inicios cómo prototipo e ir programándolo para convertirlo en una app real y llevarla a su ejecución ha sido una ardua tarea. De la misma manera, ver cómo poco a poco la app ha ido tomando forma y dominaba la integración y funcionamiento de Firebase, ha resultado reconfortante y desde luego ha merecido la pena.

Durante el diseño he podido aplicar las técnicas que aprendidas en el máster para la elaboración de todos los wireframes y realización del prototipo, además de definir la aplicación orientada al target del tipo de usuario. Obviamente, serán muchos los aspectos a mejorar, sin duda, pero creo que he conseguido generar una plataforma visualmente agradable e intuitiva.

Durante la fase de implementación y primer arranque de las diferentes pantallas con la base de datos me resulto más compleja de lo que pensaba, ya que al no haber tenido contacto previamente con bases de datos noSQL tuve dificultades. Del mismo modo, encontré que, durante la etapa de implementación, ciertos puntos cómo el incluir una sexta pestaña no se iba a visualizar correctamente, y en lugar de favorecer a la app, la iba a penalizar funcionalmente, por ello descarté implementar un ranking.

Ciertamente, el nivel de exigencia para este proyecto ha sido alto, aunque haber seguido las pautas estipuladas en el diagrama de Gantt y los consejos por parte del consultor, me han ayudado a poder seguir de una forma un poco más cómoda la planificación. Del mismo modo, haber seguido correctamente la planificación me ha permitido dar solución a errores inesperados, además de poder enfrentarme a la etapa inicial de aprendizaje de Firebase con Swift con cierta calma controlada, y así poder de una forma ágil volver a reubicarme hacia la planificación inicial.

En este proyecto se puede observar que existen diversas líneas de futuro que ClassQuest podría seguir para mejorar. Aspectos cómo la realización de retos, pueden aportar aire fresco a la forma en que los alumnos utilizan la app. Otros aspectos como un sistema de administración web para el profesorado extenderían su uso a familias de otros ciclos formativos o incluso a otras etapas educativas. Del mismo modo, que su programación para multi-plataforma extendería la aplicación a un sector de usuarios más amplio.

Sin duda, son muchas las opciones y oportunidades de ampliación que ClassQuest presenta, motivo que hace que encuentre este proyecto tan interesante.

Personalmente, estoy muy satisfecho con el resultado obtenido, dado que considero que he podido dar respuesta de forma muy completa a todos los objetivos que me marqué al inicio, y he sabido readaptar las necesidades del proyecto a los usuarios/as.

Con todo esto, haciendo una reflexión puedo observar que me llena de energía afrontar nuevos proyectos y el desarrollo de aplicaciones móviles es uno de los ámbitos que más me llena, además de poder llevar todo este aprendizaje al aula y aportar un toque de calidad profesionalmente para que mis alumnos puedan beneficiarse de todo esto.

En conclusión, me llena de orgullo haber sido capaz de afrontar las situaciones críticas y de tensión mostrándome resolutivo, y ver que con esfuerzo y dedicación uno puede conseguir cualquier reto que se proponga.

## 4. Glosario

**TIC:** Tecnologías de la Información y de la Comunicación

**app:** Aplicación

**UX:** Experiencia de usuario.

**IDE:** Entorno integrado de desarrollo

**API:** Aplicación de Programación Informática

**Wireframe:** Esquema de página o plano de pantalla.

**Rpg:** Juego de rol, del inglés *role-playing game*.

**Flat design:** Diseño de interfaz de usuario minimalista.

**CRUD:** Acrónimo de Crear, Leer, Actualizar y Borrar, en inglés (Create, Read, Update, Delete).

**Feedback:** método de control de sistemas, en el cual los resultados obtenidos de una tarea o actividad son reintroducidos nuevamente en el sistema.

**BD:** Abreviatura de Base de datos.

**Front-end:** Corresponde a la parte del software que interactúa con el o los usuarios.

**Back-end:** Corresponde a la parte que procesa la entrada de peticiones por parte del usuario.

**HTML:** Siglas en inglés de *HyperText Markup Language*, hace referencia al lenguaje de marcado para la elaboración de páginas web.

**JSON:** Acrónimo de *JavaScript Object Notation*, es un formato de texto ligero para el intercambio de datos.

**APP:** *Abreviatura* de la palabra inglesa *application*, suele referirse a aplicaciones para teléfonos móviles inteligentes.

**Target:** público objetivo al que se dirige el producto.

**Quest:** Misión, búsqueda, ir en busca...

**Enum:** Enumeración de un tipo específico de datos.

## 5. Bibliografía

- [1] Alsawaier, Raed. *The Effect of Gamification on Motivation and Engagement*. International Journal of Information and Learning Technology. (2017).
- [2] Learning by Playing: Video Games in the Classroom. nytimes.com. 2010. Consultada: 8 de Octubre de 2019. Disponible en: <https://www.nytimes.com/2010/09/19/magazine/19video-t.html>
- [3] Game icons. Game-icons.net. 2019. Consultada: 28 de Octubre de 2019. Disponible en: <https://game-icons.net/>
- [4] Flat icons. flaticon.com. 2019. Consultada: 28 de Octubre de 2019. Disponible en: <https://www.flaticon.com/>
- [5] Devianart. devianart.com. 2019. Consultada: 25 de Octubre de 2019. Disponible en: <https://www.deviantart.com>
- [6] Unity. assetstore.com. 2019. Consultada: 22 de Octubre de 2019. Disponible en: <https://assetstore.unity.com/>
- [7] Cocoapods. cocoapods.org. 2019. Consultada: 9 de Noviembre de 2019. Disponible en: <https://cocoapods.org/>
- [8] Firebase. firebase.google.com. 2019. Consultada: 14 de Noviembre de 2019. Disponible en: <https://firebase.google.com/>
- [9] Github. github.com. 2019. Consultada: 27 de Noviembre de 2019. Disponible en: <https://github.com/GocePetrovski/ScrollableSegmentedControl>
- [10] Molich, Rolf. Nielsen, Jakob. *Improving a Human-Computer Dialogue*. (1990).
- [11] Etidaten. Etidaten.fh. Pierotti, Deniese. Heuristic Evaluation. Consultada: 22 de Octubre de 2019. Disponible en: <https://cutt.ly/Me5Sv64>
- [12] Instituto nacional de estadística. Ine.es. Consultada 28 de Diciembre de 2019. Disponible en: <http://bit.ly/2QhrX7X>

## 6. Anexos

### 6.1 Anexo 1: Test de usuarios

Debido a que el test se presenta en diseño horizontal y un diseño vertical dificulta su entendimiento, se respeta la estructura de la memoria y se adjuntan los resultados en un documento anexo.

El documento se puede consultar en [Anexo1. Test de usuarios.pdf](#).