



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castellet

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO DE FINAL DE GRADO

TÍTULO DEL TFG: Desarrollo de un módulo de geocaching para una aplicación de gamificación

TITULACIÓN: Grado en Ingeniería de Aeronavegación

AUTOR: Carlos Domínguez Carreira

DIRECTOR: Miguel Valero García

FECHA: 17 de julio del 2020

Título: Desarrollo de un módulo de geocaching para una aplicación de gamificación

Autor: Carlos Domínguez Carreira

Director: Miguel Valero García

Fecha: 17 de julio del 2020

Resumen

El objetivo de este proyecto es implementar una herramienta de creación de un juego de geocaching. Para conseguir tal propósito, se ha recurrido a Classpip, que es un conjunto de proyectos destinados a desarrollar una aplicación web y móvil con el fin de gamificar la docencia.

La plataforma GitHub ha hecho posible la colaboración entre todos los participantes del proyecto, almacenando en repositorios los componentes de la aplicación. Además, Classpip se ha desarrollado sobre software libre y gracias a ello ha habido libertad total entre los integrantes de usar el programa, copiar y distribuir código e incluso modificarlo.

La aplicación web se ha desarrollado en HTML5 y JavaScript mientras que la aplicación móvil está creado con Ionic, basado en Angular, y Apache Cordova hace posible construir la aplicación en código web en vez de utilizar APIS's específicas de cada plataforma como iOS o Android.

En este documento quedan recogidas y detalladas las implementaciones que se han incorporado tanto en la web como en la aplicación móvil. También aparecen las explicaciones mínimas necesarias para la comprensión del código con la idea de allanar el camino a futuros desarrolladores que participen en el proyecto, pues esa es la clave del éxito de este concepto ya que sería imposible que un individuo abarcase tanto trabajo.

Cabe destacar que este proyecto se inicio con mi compañero Oriol Fuertes, por lo que parte de la memoria del mismo es compartida. Desarrollamos conjuntamente la web y posteriormente él se encargo de la parte de la aplicación móvil del alumno y yo la del profesor.

Title: Development of a geocaching module for a gamification application

Author: Carlos Domínguez Carreira

Director: Roc Messeguer Pallarès

Date: July 17 th 2020

Overview

The aim of this project is to implement a tool for creating a geocaching game. In order to achieve this, Classpip has been used, which is a set of projects aimed at developing a web and mobile application to make teaching more flexible.

The GitHub platform has made possible the collaboration between all the project participants, storing in repositories the application components. In addition, Classpip has been developed on free software and thanks to this there has been total freedom among the members to use the program, copy and distribute code and even modify it.

The web application has been developed in HTML5 and JavaScript while the mobile application is created with Ionic, based on Angular, and Apache Cordova makes it possible to build the application in web code instead of using platform-specific APIs like iOS or Android.

In this document the implementations that have been incorporated both in the web and in the mobile application are collected and detailed. Also appear the minimum explanations necessary for the understanding of the code with the idea of paving the way for future developers to participate in the project, since that is the key to the success of this concept since it would be impossible for an individual to cover so much work.

It should be noted that this project was started with my partner Oriol Fuertes, so part of the memory of this project is shared. We developed the website together and later he was in charge of the student's mobile application and I was in charge of the teacher's one.

ÍNDICE

INTRODUCCIÓN	7
CAPÍTULO 1. GAMIFICACIÓN	9
1.1. Definición	9
1.2. Tipos de geocachés	10
1.3. Aplicaciones.....	10
CAPÍTULO 2. CLASSPIP	12
2.1. Funcionalidades	12
2.2. Arquitectura	13
2.2.1. Classpip-mobile profesor.....	14
2.2.2. Classpip-mobile alumno	14
2.2.3. Classpip-Dashboard	14
2.2.4. Classpip-services	14
2.2.5. Classpip-website	14
2.2.6. Classpip-server.....	15
2.2.7. External tools	15
CAPÍTULO 3. OBJETIVOS Y PLAN DE TRABAJO	16
3.1. Objetivo	16
3.2. Plan de trabajo.....	17
CAPÍTULO 4. DESCRIPCIÓN DEL JUEGO Y FUNCIONALIDADES.....	18
4.1. Escenarios.....	18
4.2. Características	18
4.3. Reglas	18
4.4. Requisitos	19
4.5. Funcionalidades	19
4.5.1. Funcionalidades del Dashboard	19
4.5.2. Funciones del profesor	20
4.5.3. Funciones del alumno	21
CAPÍTULO 5. DISEÑO DE LAS PANTALLAS.....	22
5.1. Escenarios.....	22
5.2. Pantallas del profesor	28

CAPÍTULO 6. MODELO DE DATOS	32
6.1. Creación de modelos	33
6.2. Entidad-relación.....	34
6.3. Creación de relaciones	37
CAPÍTULO 7. PRUEBAS Y EVALUACIÓN	39
CAPÍTULO 8. ESTUDIO DE PRECISIÓN	43
CAPÍTULO 9. CONCLUSIONES	45
9.1. Conclusiones técnicas.....	45
9.2. Conclusiones personales	45
9.3. Propuestas futuras mejoras	46
ANEXOS	47
ANEXO I: Implementación	47
I.I. Clases	47
I.II. Componentes	48
I.III. Servicios	49
I.III.I. Peticiones a la API	50
I.III.II. Cálculos	50
I.III.III. Sesión.....	51
I.IV. Datos cuantitativos	52
I.V. Resultados	53
I.V.I. Sesión	53
I.V.II. Funcionalidades del móvil profesor	65

ÍNDICE DE FIGURAS

Fig 2.1. Arquitectura de Classpip (Fuente: Classpip)	13
Fig 5.1. Pantalla de “CONFIRMAR ACTIVAR”, “CONFIRMAR ELIMINAR” de Dashboard (Fuente: MockingBot)	22
Fig 5.2. Pantalla de “MIS ESCENARIOS” de Dashboard (Fuente: MockingBot)	23
Fig 5.3. Pantalla de “CREAR JUEGO” y “CREAR JUEGO 2” de Dashboard (Fuente: MockingBot)	24
Fig 5.4. Pantalla de “CREAR ESCENARIO” de Dashboard (Fuente: MockingBot)	24
Fig 5.5. Pantalla de “DEFINIR REGLAS” y “EDITAR REGLAS” de Dashboard (Fuente: MockingBot)	25
Fig 5.6. Pantalla de “AÑADIR UBICACIÓN” de Dashboard (Fuente: MockingBot)	26
Fig 5.7. Pantalla de “CREAR PREGUNTA” y “EDITAR PREGUNTA” de Dashboard (Fuente: MockingBot)	27
Fig 5.8. Pantalla de “MIS PREGUNTAS” de Dashboard (Fuente: MockingBot)	27
Fig 5.9. Pantalla de “JUEGOS” y “EDITAR” del profesor (Fuente: MockingBot)	29
Fig 5.10. Pantalla de “AÑADIR PUNTOS”, “Mapa” y “Coordenadas” del profesor (Fuente: MockingBot)	29
Fig 5.11. Pantalla de “REGLAS” del profesor (Fuente: MockingBot)	30
Fig 5.12. Pantalla de “CONFIRMAR ACTIVAR”, “CONFIRMAR ELIMINAR” del profesor (Fuente: MockingBot)	31
Fig 6.1. Modelos existentes en la API-REST (Fuente: localhost:3000/explorer)	32
Fig 6.2. Creación de modelos (Fuente: Visual Studio Code)	33
Fig 6.3. Creación de atributos (Fuente: Visual Studio Code)	33
Fig 6.4. Modelo PuntoGeolocalizable (Fuente: Visual Studio Code)	34
Fig 6.5. Modelos y relaciones de entidad de Geocaching (Fuente: propia)	35
Fig 6.6. Modelos y relaciones de entidad de Geocaching con modelos puente (Fuente: propia)	36
Fig 6.7. Relación entre Profesor y JuegoDeGeocaching (Fuente: Visual Studio Code)	37
Fig 6.8. Relación entre Alumno y JuegoDeGeocaching through AlumnoJuegoDeGeocaching (Fuente: Visual Studio Code)	37
Fig 6.9. Relación entre AlumnoJuegoDeGeocaching y Alumno (Fuente: Visual Studio Code)	38
Fig 7.1. Resultados acerca de la facilidad de uso (Fuente: propia)	41
Fig 7.2. Resultados acerca del diseño gráfico (Fuente: propia)	42
Fig 7.3. Resultados acerca de la valoración general (Fuente:propia)	42
Fig I.1. Clase JuegoDeGeocaching (Fuente: Visual Studio Code)	48
Fig I.2. Estilos gráficos del componente CSS de Juego (Fuente: Visual Studio Code)	48
Fig I.3. Componente HTML de Juego (Fuente: Visual Studio Code)	49
Fig I.4. Componente Typescript de Juego (Fuente: Visual Studio Code)	49

Fig I.5. URL del servicio Peticiones a la API (Fuente: Visual Studio Code)	50
Fig I.6. Ejemplos de funciones para JuegosDeGeocaching (Fuente: Visual Studio Code)	50
Fig I.7. Servicio de Cálculos (Fuente: Visual Studio Code)	51
Fig I.8. Funciones dentro del servicio de Sesión (Fuente: Visual Studio Code)	51
Fig I.9. Pantalla Recursos (Fuente: Dashboard)	54
Fig I.10. Pantalla Mis Escenarios (Fuente: Dashboard)	54
Fig I.11. Pantalla Editar Escenario (Fuente: Dashboard)	55
Fig I.12. Pantalla Agregar PuntoGeolocalizable (Fuente: Dashboard)	56
Fig I.13. Pantalla principal del Dashboard (Fuente: Dashboard)	56
Fig I.14. Pantalla Mis Grupos (Fuente: Dashboard)	56
Fig I.15. Pantalla Grupo Seleccionado (Fuente: Dashboard)	57
Fig I.16. Pantalla Editar Grupo (Fuente: Dashboard)	58
Fig I.17. Pantalla Añadir Alumno (Fuente: Dashboard)	58
Fig I.18. Mensaje alerta Eliminar Grupo (Fuente: Dashboard)	59
Fig I.19. Pantalla Juegos Activos (Fuente: Dashboard)	59
Fig I.20. Pantalla Crear Juego (Tipo) (Fuente: Dashboard)	60
Fig I.21. Pantalla Juegos Inactivos/Preparados (Fuente: Dashboard)	60
Fig I.22. Pantalla Crear Juego (Modo) (Fuente: Dashboard)	61
Fig I.23. Pantalla Crear Juego (Añadir Escenario) (Fuente: Dashboard)	61
Fig I.24. Ventana emergente Añadir escenario (Fuente: Dashboard)	62
Fig I.25. Pantalla Preguntas (Fuente: Dashboard)	63
Fig I.26. Ventana emergente Añadir Preguntas (Fuente: Dashboard)	63
Fig I.27. Pantalla Puntuaciones (Fuente: Dashboard)	64
Fig I.28. Pantalla Finalización (Fuente: Dashboard).....	64
Fig I.29. Pantalla Inicio/Credenciales erróneas (Fuente: Aplicación del profesor).....	65
Fig I.30. Pantalla Mis Grupos y pantalla con un grupo seleccionado (Fuente: Aplicación del profesor).....	66
Fig I.31. Pantalla Mis Juegos Activos e Inactivos/Preparados (Fuente: Aplicación del profesor).....	66
Fig I.32. Pantalla un juego Activo, Inactivo y Preparado respectivamente (Fuente: Aplicación del profesor)	67
Fig I.33. Pantalla de Inicio y Opciones respectivamente (Fuente: Aplicación del profesor).....	68
Fig I.34. Pantalla de Escenarios, un Escenario concreto y un punto concreto respectivamente (Fuente: Aplicación del profesor)	68

INTRODUCCIÓN

Las nuevas tecnologías son protagonistas hoy en día. Todos los ámbitos evolucionan muy rápido y el entorno se ajusta a ellas. Entre otros ámbitos, la educación también se somete a este cambio, ofreciendo nuevas alternativas y convirtiéndola en un concepto renovado.

- **¿Qué hemos hecho?**

Se ha desarrollado una aplicación en un principio con Oriol Fuertes, y después de manera individual basada en la gamificación para entornos docentes y educativos utilizando el juego geocaching. Debido a esto, todo lo referido al Dashboard se comparte con la memoria de Oriol, ya que lo hicimos entre los dos.

En este juego, los alumnos recurrirán a su ingenio y pericia para encontrar unos puntos predeterminados (mediante GPS) por el profesor previamente mediante unas pistas. Al llegar a estos puntos, los alumnos serán puestos a prueba con preguntas relacionadas con la asignatura y serán puntuados en función de sus resultados. La aplicación ofrece una serie de pantallas, escenarios, pistas y una manera de calificar al estudiante mediante este método.

La modalidad será individual, se utilizarán bonus como preguntas extra o cronometraje del tiempo empleado para incentivar y ayudar al alumno en el aprendizaje.

Toda la aplicación puede ser gestionada por los profesores. Serán capaces de modificar el escenario, añadir preguntas nuevas, nuevas pistas o incluso cambiar los juegos e itinerarios.

- **Gamificación. ¿Qué es?**

Dicha aplicación se desarrolla en una nueva sección de la herramienta Classpip. La aplicación Classpip es un conjunto de proyectos desarrollados por distintos estudiantes de la Escuela Superior de Ingeniería de Telecomunicaciones y Aeronáutica de Castelldefels (EETAC). El proyecto global trata de utilizar la gamificación en el entorno educativo, de manera que se rompan las barreras del clásico método de enseñanza y se puedan introducir las nuevas tecnologías en el ámbito.

Está compuesto por diferentes módulos que funcionan de manera autónoma. Los módulos son los siguientes:

- Módulo de Colecciones, basado en un sistema de asignación de cromos.
- Módulo de Cuestionario, proporciona la posibilidad de crear cuestionarios.
- Módulo de Puntos e Insignias, permite crear un sistema de asignación de puntos e insignias.

Por último, existe un módulo de conexiones que unifica los tres módulos mencionados anteriormente.

- **Estructura de la memoria**

La memoria de este trabajo consta de diferentes capítulos. En el primer apartado se explicarán las bases del término de gamificación y cómo se ha utilizado en nuestro proyecto Classpip en el capítulo 3.

En el capítulo 4 se definirán los objetivos y el plan de trabajo que se ha seguido, y en los siguientes la descripción del juego y las funcionalidades, el diseño de las pantallas, el modelo de datos y el diseño gráfico.

Finalmente se hablará de la implementación y sus detalles, las pruebas y evaluación realizadas para comprobar el correcto funcionamiento de la aplicación y se concluirá con unas conclusiones técnicas, personales y cómo puede afectar este proyecto y todo lo aprendido con él en un futuro.

- **Tareas**

Inicialmente el trabajo sobre el juego de geocaching iba a ser con Oriol Fuertes así que lo empezamos a desarrollar en conjunto. Debido a eso, el Dashboard es una parte en común. Cuando se nos comunicó que teníamos que hacerlo de manera individual, nos repartimos el trabajo restante desarrollando mi compañero la aplicación del alumno y yo la del profesor. A lo largo de ese periodo de tiempo aún mantuvimos reuniones conjuntas con los tutores y trabajamos en paralelo porque muchas partes estaban relacionadas.

En cuanto a las funcionalidades, el Dashboard permite al profesor la creación de juegos de geocaching y modificar sus elementos. La aplicación del alumno permite a los alumnos jugar al juego y la aplicación del profesor permite capturar coordenadas e iniciar y finalizar juego.

CAPÍTULO 1. GAMIFICACIÓN

En este capítulo se introducirán conceptos básicos sobre la gamificación y se introducirá el juego de geocaching. También se proporcionarán algunos ejemplos para facilitar la comprensión del juego.

1.1. Definición

La gamificación es una técnica de aprendizaje que aplica los elementos de diseño de juegos en un entorno educativo-profesional con el fin de conseguir mejores resultados. También se puede definir como un conjunto de actividades y procesos para resolver problemas usando o aplicando las características de los elementos del juego.

Una colección de investigaciones sobre la gamificación, como por ejemplo:

- Estudio sobre gamificación 1: Universidad de Montreal. La gamificación mejora la participación, la atención y la retención del conocimiento.
- Estudio sobre gamificación 2: Universidad de Trunoyo Madura. La gamificación es divertida y productiva.
- Estudio sobre gamificación 3: Mediterranean University y RWTH Aachen University. La gamificación mejora las notas de los alumnos.
- Estudio sobre gamificación 4: University of Maribor. La gamificación ayuda a darle un significado estimulante al acto de formarse.
- Estudio sobre gamificación 5: Autodesk Research Toronto. Ahorra tiempo y aumenta el interés.

Muestra que la mayoría de los estudios sobre la gamificación encuentran que tiene efectos positivos en las personas. Sin embargo, existen diferencias individuales y contextuales.

En nuestro caso utilizamos la gamificación a través del juego del geocaching.

El geocaching es un juego de búsqueda en el mundo real con ayuda de dispositivos GPS. Los participantes deben ir a unas coordenadas específicas a través de unas pistas y realizar unas acciones en esas coordenadas. En este caso orientamos el geocaching a entorno educativo, utilizándolo como método de gamificación.

Geocaching tuvo su origen en el grupo de noticias sci.geo.satellite-nav dedicado a los Sistemas Globales de Navegación por Satélite (GNSS).

Dave Ulmer propuso el juego *Great American GPS Stash Hunt* al resto de miembros, escondiendo el 3 de mayo un "cofre del tesoro" en los alrededores de la ciudad de Portland (Oregón) y enviando al grupo de noticias las coordenadas exactas de su ubicación (N 45° 17.460 W 122° 24.800).

1.2. Tipos de geocachés

El “caché” o “geocaché” son cada uno de los contenedores del geocaching, cuya interpretación es el “tesoro” o el “geoescondite”. Los geocachés principales son los siguientes:

- Geocaché tradicional: Es el tipo de geocaché original, y el más directo. El caché consiste en un contenedor de tamaño variable en las coordenadas indicadas. Todos los geocachés tienen un libro de registro, y los que sean más grandes pueden contener objetos de intercambio y rastreables.
- Cachés “mystery” o puzzle: Nuevos y únicos geocachés que no encajan en otras categorías. Pueden contener complicados acertijos que deben ser resueltos para averiguar las coordenadas.
- Multi-caché: Son geocachés que utilizan dos o más ubicaciones, de las que la última será un contenedor físico con un libro de registro. Normalmente en la primera etapa se recibirá una pista sobre la posición de la segunda, y así sucesivamente.

Una vez que los geocachés encuentran un geocaché, siguen estas reglas básicas:

- Se rellena un cuaderno de registro.
- Se obtiene algo del geocaché.
- Se coloca algo propio en el geocaché.
- Se devuelve el geocaché a la posición exacta y en las mismas condiciones en las que se encontró.

1.3. Aplicaciones

Hoy en día existen algunas aplicaciones que utilizan el geocaching, por ejemplo aplicaciones móviles que están al alcance de cualquier persona que posea un Smartphone. Algunos ejemplos son:



- Pokémon GO

Pokémon GO es un videojuego de realidad aumentada basado en la localización y el geocaching desarrollado por Niantic, Inc. La aplicación está disponible para dispositivos iOS y Android. El juego consiste en buscar y capturar personajes de la saga Pokémon escondidos en lugares

determinados del mundo real y luchar con ellos, e implica el desplazamiento del usuario por las calles de la ciudad para avanzar en el juego.



- Cachly

Es una aplicación de geocaching simple y poderosa para iOS que incluye muchas características avanzadas que incluyen el registro para una cuenta Geocaching, la interacción con amigos, encontrar cachés y mapas y cachés offline.



- Geocaching

Esta comunidad de búsqueda del tesoro es una muy buena herramienta. Hay millones de contenedores ingeniosamente escondidos en más de 185 países a la espera de ser encontrados. Ofrece búsqueda en vivo, comunicación con otros jugadores y registro de trazables. También ofrece una variedad Premium.

CAPÍTULO 2. CLASSPIP

Con tal de combatir la falta de entusiasmo de muchos estudiantes, se decidió crear una herramienta aplicando los métodos de la gamificación. Es por ello que se creó Classpip, que consiste en una página web y una aplicación móvil, las cuales serán usadas en clase.

Classpip es un macro proyecto que ha ido evolucionando gracias a la suma de diferentes proyectos individuales desarrollados por varios estudiantes de la EETAC realizando proyectos de fin de grado o máster.

2.1. Funcionalidades

Estos proyectos realizados han proporcionado diversas funcionalidades a Classpip como son:

- Inicio

Pantalla que ayuda a entender, mediante vídeos, el funcionamiento de la aplicación y sus funcionalidades.

- Grupos

- Profesor: Permite ver todos los grupos que imparta y permite observar y editar los detalles correspondientes de cada grupo (nombre del grupo, alumnos,...)
- Alumno: Permite observar en que grupos pertenece, generalmente asociados a las asignaturas que curse.

- Cuestionarios y preguntas

Esta funcionalidad permite al profesor crear distintos cuestionarios/preguntas para que los alumnos tengan la oportunidad de contestarlos.

- Colecciones

- Profesor: Esta funcionalidad permite al profesor crear cromos virtuales con una temática concreta (por ejemplo, jugadores de la NBA o físicos) con tal de formar álbumes. Los cromos creados se entregarán a los alumnos en forma de recompensación al lograr ciertos objetivos, permitiendo completar la colección de forma progresiva.
- Alumno: Podrá hacer un seguimiento a los cromos obtenidos y a los restantes de cada álbum.

- Puntos e Insignias

- Profesor: Similarmente a Colecciones, con la funcionalidad de Puntos e Insignias, el profesor podrá crear insignias y diferentes tipos de puntos (comportamiento, asistencia, ...), que posteriormente podrá asignar a los alumnos con tal de recompensarlos.
- Alumno: Podrá hacer el seguimiento de los puntos e insignias obtenidas.

- Competiciones

Permite crear torneos o ligas con tal de enfrentar, de forma individual o por equipos, a los alumnos de un grupo específico, con tal de recompensarlos con puntos, insignias o cromos.

2.2. Arquitectura

A lo largo de toda su existencia, Classpip ha mantenido su arquitectura intacta, con la excepción del diseño web, pues se sustituyó *BootStrap* por *Angular Material*. A continuación se muestra el esquema completo de la arquitectura de Classpip (Fig 2.1).

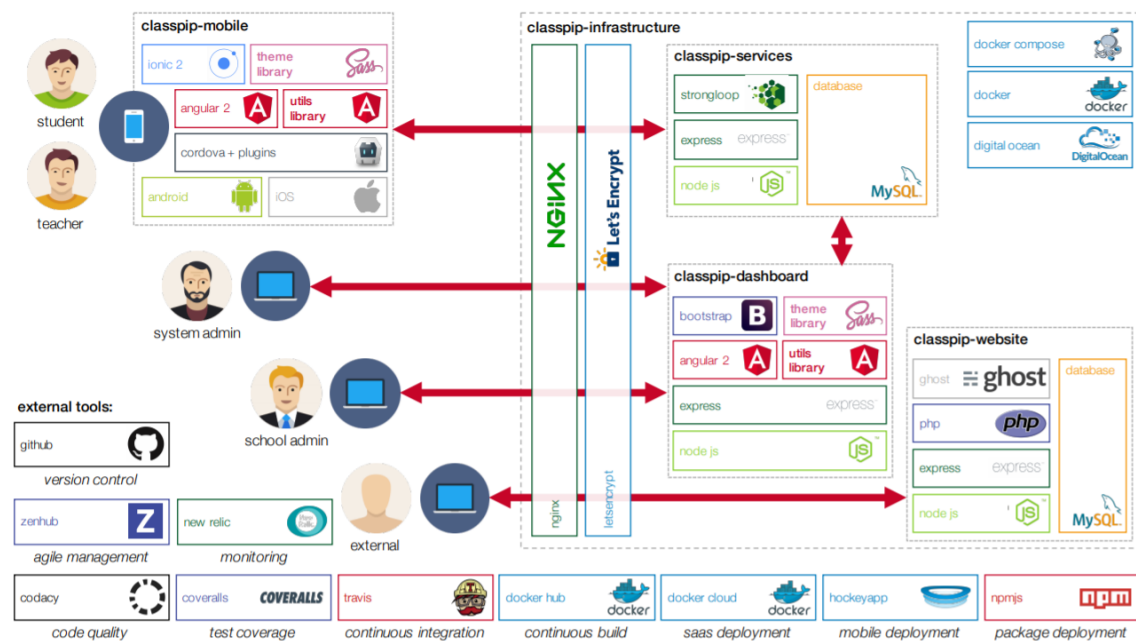


Fig 2.1. Arquitectura de Classpip (Fuente: Classpip)

Como se puede observar, está formado por 4 repositorios y un conjunto de herramientas externas. Recientemente, se ha añadido un nuevo repositorio, el classpip-server, por lo tanto, existen 5.

2.2.1. Classpip-mobile profesor

Se trata de la aplicación móvil para el profesor. Desarrollada con el framework *Ionic* junto con otras herramientas como *Angular* y *Angular Material*.

Cabe destacar que *Ionic* usa Apache Cordova que nos permitirá usar en nuestro proyecto las características propias de los teléfonos móviles, como son la cámara o el GPS entre otros.

En concreto, esta aplicación tiene como objetivo que el profesor sea capaz de configurar los parámetros más importantes del juego (los estados de los juegos y los puntos geolocalizables desde su dispositivo móvil).

2.2.2. Classpip-mobile alumno

La aplicación móvil del alumno está desarrollada con el mismo framework y las mismas herramientas que la del profesor. Por lo tanto, podremos disponer de las mismas características que las del apartado anterior (GPS, cámara...).

Tiene por principales objetivos la realización del juego, contestando a las preguntas pertinentes y mostrando las alertas de proximidad establecidas.

2.2.3. Classpip-Dashboard

Esta parte se centra en la aplicación web, y está basada principalmente para los profesores pues permite configurar con mayor facilidad las características que aparecen en la aplicación móvil. Basada en *node.js*, *express*, *Angular* y *Angular Material* para la interfaz del usuario.

2.2.4. Classpip-services

Esta parte está basada en una API-REST conectada a la base de datos que permite la lectura, creación, modificación o eliminación de datos mediante direcciones html. Dicha API-REST ha sido creada gracias a *strongloop*, que está basado en *node.js* y *express*.

2.2.5. Classpip-website

Esta parte se centra en dar visibilidad del proyecto a personas externas al desarrollo mediante una página web. Además, proporciona el material necesario para iniciarse a dicho proyecto.

La página web se ha creado usando un servidor web *express* y *node.js*.

2.2.6. Classpip-server

Este nuevo repositorio, se encarga de realizar tareas de notificación entre los usuarios, sin tener que actualizar la aplicación.

2.2.7. External tools

Esta última parte recoge otros frameworks usados como pueden ser *GitHub*: un repositorio online para colgar y descargar los códigos que forman el proyecto, o *npm*: un gestor de paquetes para el entorno *node.js*, entre otros.

CAPÍTULO 3. OBJETIVOS Y PLAN DE TRABAJO

En esta parte del documento se presentarán los objetivos del trabajo de manera sintética y precisa, y también aparecerá el plan de trabajo diseñado desde un comienzo, esclareciendo que partes realizó quién.

3.1. Objetivo

El objetivo de nuestro proyecto pretende incentivar al alumno con juegos al aire libre diferentes al método clásico educativo con tal de ofrecer nuevas alternativas de aprendizaje y evaluación a los alumnos. Por ello, se pretende crear un juego de Geocaching dentro de la aplicación de Classpip. Fueron necesarios los siguientes avances:

- Implementar los modelos de datos necesarios.
- Creación de las clases en los repositorios pertinentes.
- Los alumnos deben descubrir diferentes puntos geolocalizables.
- Sumarán puntos por contestar preguntas correctamente y serán clasificados en función de ello.
- Obtendrán recompensas según su posición final en la clasificación.

El juego se basa en descubrir diferentes puntos geolocalizables propuestos mediante pistas para tener la oportunidad de sumar puntos respondiendo a una pregunta realizada en dicha ubicación. Los puntos conseguidos en cada etapa (referente a cada punto geolocalizable) serán acumulados, permitiendo así calificar al conjunto de participantes y consecuentemente, determinar el ganador al final del juego, además de recompensar a los alumnos participantes con puntos y/o cromos usando las funcionalidades ya existentes en la versión actual de la aplicación.

Para ello, será necesaria la implementación de una sección nueva en la aplicación que permita la creación de juegos de Geocaching, ofreciendo al profesor la personalización de los diversos parámetros del juego.

Además, se deberán crear todas las pantallas asociadas a la aplicación móvil para que en los teléfonos de los alumnos se permita jugar a estos juegos de forma correcta, mientras que en la aplicación móvil del profesor permita capturar puntos geolocalizables con sus coordenadas GPS y hacer el seguimiento del juego.

3.2. Plan de trabajo

A raíz de las metas anteriores, diseñamos un plan de trabajo para llevarlas a cabo de la forma más óptima posible:

- Alcanzar una base mínima de conocimientos acerca de Classpip mediante video tutoriales de Miguel en *YouTube* - *Tarea realizada individualmente*.
- Diseño de las pantallas y funcionalidades tanto del Dashboard como de las aplicaciones móviles - *Tarea realizada con Oriol*.
- Desarrollo del código Dashboard - *Tarea realizada con Oriol*.
- Desarrollo del código de la aplicación del profesor - *Tarea realizada individualmente*.
- Retoques gráficos de apariencia - *Tarea realizada individualmente*.
- Estudio precisión - *Tarea realizada individualmente*.

En un inicio, nuestros conocimientos acerca de Classpip eran nulos, así que lo primero fue adquirir una base en esta aplicación para poder empezar a entender lo que queríamos hacer. Para ello, Miguel nos proporcionó unos tutoriales en formato video en la plataforma *YouTube*, donde explicaba paso a paso las nociones básicas requeridas.

A continuación diseñamos las pantallas y las funcionalidades tanto del Dashboard como de las aplicaciones móviles. Acto seguido empezamos el desarrollo del código, primero del Dashboard y después las aplicaciones tanto del alumno como la del profesor. Cuando aseguramos su correcto funcionamiento, retocamos la parte gráfica con el propósito de mejorar su aspecto y amigabilidad.

Para finalizar comprobamos nuevamente que no hubieran aparecido errores y todo funcionara correctamente.

CAPÍTULO 4. DESCRIPCIÓN DEL JUEGO Y FUNCIONALIDADES

En este apartado se explicará el juego y sus funcionalidades. A continuación se especifican los escenarios que va a tener el juego, sus características más importantes, las reglas, requisitos y las funcionalidades de los usuarios de la aplicación.

4.1. Escenarios

Cada juego tendrá asociado un escenario. Cada escenario se caracteriza por los siguientes elementos principales: el mapa, los puntos geolocalizables y las pistas. Se debe tener en cuenta que un mismo escenario podrá ser compartido en diferentes juegos. Cuando el profesor cree un juego, deberá elegir un escenario y entonces asignará las preguntas a los puntos geolocalizables que desee (preguntas relacionadas con los contenidos de las asignaturas).

4.2. Características

- **Recorridos:** Se dispondrán de recorridos diferentes para cada alumno. Con tal de evitar malas prácticas, a cada participante se le asignarán etapas con un orden distinto.
- **Ubicaciones:** se proporcionarán pistas (una difícil y otra fácil) que conducirán a los alumnos al lugar donde se realizará la pregunta de la etapa.
- **Preguntas:** al igual que los recorridos, las preguntas y respuestas estarán desordenadas.
- **Bonus:** el alumno tendrá la opción de obtener una puntuación adicional si contesta correctamente a una pregunta extra al final de cada etapa y será penalizado en caso de errarla.

4.3. Reglas

El juego se divide en tantas etapas como puntos geolocalizables tenga el escenario elegido el profesor organizador de la prueba. Al inicio de cada etapa, a los alumnos les aparecerá la pista difícil y un botón con la opción de rendirse, que de ser utilizado mostrará la pista fácil, que prácticamente revelará la ubicación del punto a dirigirse.

El alumno incapaz de encontrar la ubicación, será penalizado con una reducción de los puntos potenciales de la pregunta puntuable en un 20%, es decir, si la pregunta vale 10 puntos y la responde correctamente, obtendría 8. No obstante, con tal de ayudar, cuando los participantes se encuentren a menos de 25 metros la ubicación de la pregunta, saltará una alerta (“Caliente, Caliente...”).

Cuando se encuentren a menos de 5 metros de la ubicación, saltará otra alerta (¡Has Llegado!) y se les formulará la pregunta. Si el alumno contesta correctamente recibirá la puntuación por respuesta correcta introducida por el profesor en el momento de crear las reglas del juego y si falla se restará la puntuación incorrecta del juego. Los participantes que hayan respondido correctamente tendrán la opción de contestar a una pregunta adicional de dificultad superior la cual recompensará o penalizará en caso de fallar, los porcentajes definidos por el profesor para las puntuaciones de las preguntas bonus.

Cada vez que termine una etapa, los puntos obtenidos serán sumados a la puntuación total del alumno, para mostrarlos en la clasificación general. Al terminar el juego, el ganador de la prueba será el que encabece dicha tabla.

4.4. Requisitos

- Estar registrado a Classpip.
- Formar parte de un grupo con juegos de geocaching activos.
- Disponer de un móvil con GPS.

4.5. Funcionalidades

A continuación se enunciarán las funcionalidades junto a una pequeña descripción tanto del Dashboard como la de la aplicación del profesor y la del alumno.

4.5.1. Funcionalidades del Dashboard

- Ver Mis Escenarios: Editar/Eliminar escenarios creados previamente.

- Crear Escenarios: se mostrará el conjunto de parámetros a completar para crear un escenario (Mapa y descripción, PuntosGeolocalizables).
- Crear Preguntas: se mostrará el conjunto de parámetros a completar para crear una pregunta (título, pregunta, temática y respuestas).
- Ver Mis Preguntas: Editar/Eliminar base de preguntas creadas previamente.
- Crear Juego (implementado dentro de Mis Grupos): Seleccionar un escenario, el conjunto de preguntas de cada etapa y las reglas.
- Lista de Juegos: Observar el estado del juego (activo, inactivo o preparado) y observar la clasificación. Dependiendo de su estado, tendrá la opción de activar, desactivar, terminar y/o eliminar el juego.

4.5.2. Funciones del profesor

- Crear Escenarios: Seleccionar conjunto de puntos geolocalizables de la prueba en el mapa.
- Ver Mis Escenarios: Editar/Eliminar escenarios creados previamente.
- Crear Preguntas: Añadir preguntas (en formato test) a la base.
- Ver Mis Preguntas: Editar/Eliminar base de preguntas creadas previamente.
- Crear Juego (implementado dentro de Mis Grupos): Seleccionar un escenario, el conjunto de preguntas de cada etapa y las reglas.
- Lista de Juegos: Observar el estado del juego (activo, inactivo o preparado) y observar la clasificación. Dependiendo de su estado, tendrá la opción de activar, desactivar, terminar y/o eliminar el juego.
- Tendrá capacidad de obtener coordenadas para los puntos geolocalizables de un escenario mediante la propia ubicación del teléfono. Además, podrá modificar el estado de los juegos (activo, inactivo o preparado).

4.5.3. Funciones del alumno

- Jugar al juego: Al inicio de cada etapa, escoger entre la pista difícil o fácil, en caso de no encontrar la ubicación y hacer uso del botón Rendirse.
- Seguir las alertas de proximidad.
- Responder a las preguntas en cada ubicación.
- Responder de forma voluntaria a la pregunta bonus en caso de acertar la pregunta propia de la ubicación.
- Observar la clasificación.

CAPÍTULO 5. DISEÑO DE LAS PANTALLAS

En este apartado se detalla el diseño inicial de las pantallas que contendrá el juego de geocaching de nuestra aplicación, con tal de tener una mejor comprensión de cómo realizar el código necesario. Como se ha mencionado anteriormente, la aplicación tendrá una versión para el profesor y otra para el alumno, permitiendo al profesor la edición del juego en cuestión.

Además también se podrá configurar el juego a través desde un portal web al que nos referiremos como “Dashboard”. En este portal el profesor podrá encontrar una forma más amigable y completa para configurar el juego a sus necesidades.

5.1. Escenarios

- Pantalla de inicio: En esta pantalla el usuario puede diferenciar dos grupos: preguntas y escenarios. En cada uno de ellos podrá visualizar los ya existentes y crear nuevos (Fig 5.1).

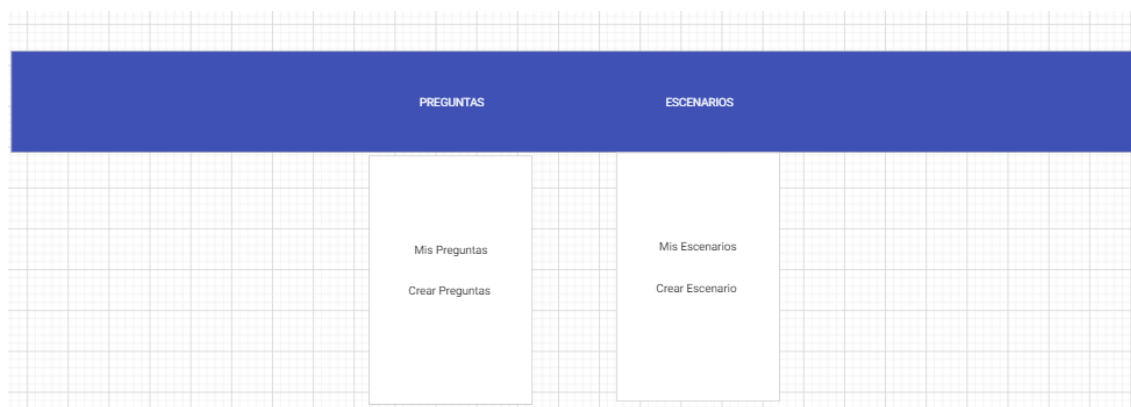


Fig 5.1. Pantalla de “CONFIRMAR ACTIVAR”, “CONFIRMAR ELIMINAR” de Dashboard (Fuente: MockingBot)

- Pantalla “MIS ESCENARIOS”: A través de la pantalla de inicio se puede llegar a la pantalla “MIS ESCENARIOS”, en la cual se podrá editar los escenarios en “EDITAR” y eliminar escenarios en “ELIMINAR” (Fig 5.2).

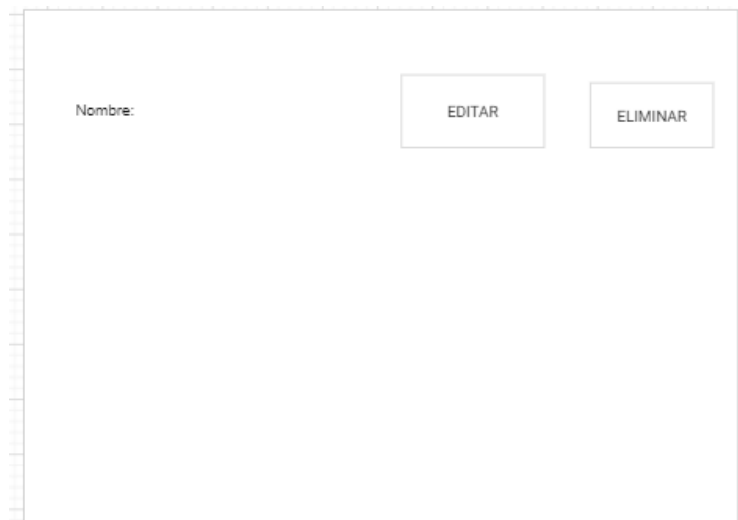


Fig 5.2. Pantalla de “MIS ESCENARIOS” de Dashboard (Fuente: MockingBot)

- Pantalla “CREAR JUEGO”: En esta pantalla el usuario será capaz de crear un nuevo juego. Podrá elegir entre distintas modalidades de juego, pero en nuestro caso siempre será el juego de geocaching. Finalmente, tendrá que pinchar en “SIGUIENTE” para proceder a la creación del juego.

Una vez allí, tendrá una nueva pantalla donde podrá definir cuantas etapas tiene el juego y las pistas difíciles y fáciles de cada etapa. Además podrá crear un escenario en la pantalla “CREAR ESCENARIO” y finalizar la creación del juego en “CREAR JUEGO” (Fig 5.3).

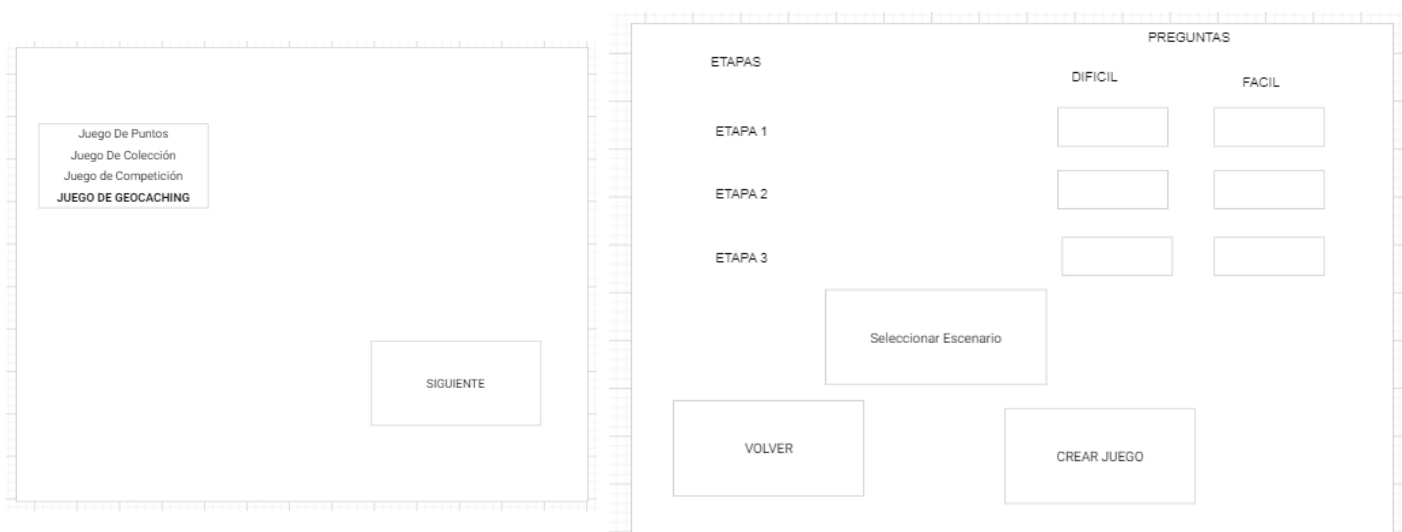


Fig 5.3. Pantalla de “CREAR JUEGO” y “CREAR JUEGO 2” de Dashboard (Fuente: MockingBot)

- Pantalla “CREAR ESCENARIO”: En la siguiente pantalla se permite al usuario crear un escenario. En la parte superior de la pantalla se debe introducir el nombre del nuevo escenario y seguidamente la ubicación en “AGREGAR UBICACIÓN” que permite la formación de etapas. Además podrán ser editadas individualmente en el botón “EDITAR” o bien ser eliminadas en “ELIMINAR” (Fig 5.4).

The image shows a wireframe of a mobile application screen for creating a scenario. At the top, there is a text label "Nombre del escenario:" followed by a rectangular input field. Below this, there is another text label "Ubicación:" followed by a button labeled "AGREGAR UBICACION". Underneath the "AGREGAR UBICACION" button, there is a table with four columns: "ETAPA 1", "LONG", "LAT", and two empty cells. To the right of the "LONG" and "LAT" columns are two buttons labeled "EDITAR" and "ELIMINAR". At the bottom of the screen, there are two large buttons: "VOLVER" on the left and "SIGUIENTE" on the right.

Fig 5.4. Pantalla de “CREAR ESCENARIO” de Dashboard (Fuente: MockingBot)

- Pantalla “DEFINIR REGLAS”: En la siguiente pantalla el usuario podrá definir las reglas del juego. Se podrá modificar los puntos por pista difícil, los puntos por pregunta, los puntos por error, el

aumento de puntos del bonus y su penalización. Finalmente se guardarán los cambios en “CREAR ESCENARIO”.

Además estas reglas podrán ser modificadas anteriormente desde la pantalla “EDITAR REGLAS”, que tendrá la misma estructura que la primera pero permitirá cambiar las reglas una vez establecidas (Fig 5.5).

The image shows two wireframe screens for rule management, both titled "REGLAS".

The left screen, "DEFINIR REGLAS", contains five input fields for the following rules:

- Puntos por pista difícil (5):
- Puntos por pregunta (10):
- Puntos por error (0):
- Aumento de puntos del bonus (50%):
- Penalización por errar el bonus (25%)

At the bottom, there are two buttons: "VOLVER" and "CREAR ESCENARIO".

The right screen, "EDITAR REGLAS", has the same layout as the left screen but with a button labeled "ACTUALIZAR ESCENARIO" instead of "CREAR ESCENARIO".

Fig 5.5. Pantalla de “DEFINIR REGLAS” y “EDITAR REGLAS” de Dashboard (Fuente: MockingBot)

- Pantalla “AÑADIR UBICACIÓN”: En esta pantalla se permite al usuario la creación de una nueva ubicación. Tiene dos opciones, a través de la selección de un punto en el mapa, o bien a través de la introducción manual de coordenadas (latitud y longitud). Además esta pantalla permite facilitar directamente las preguntas difícil y fácil de la ubicación escogida. Finalmente se guardará en “AÑADIR UBICACIÓN” (Fig 5.6).

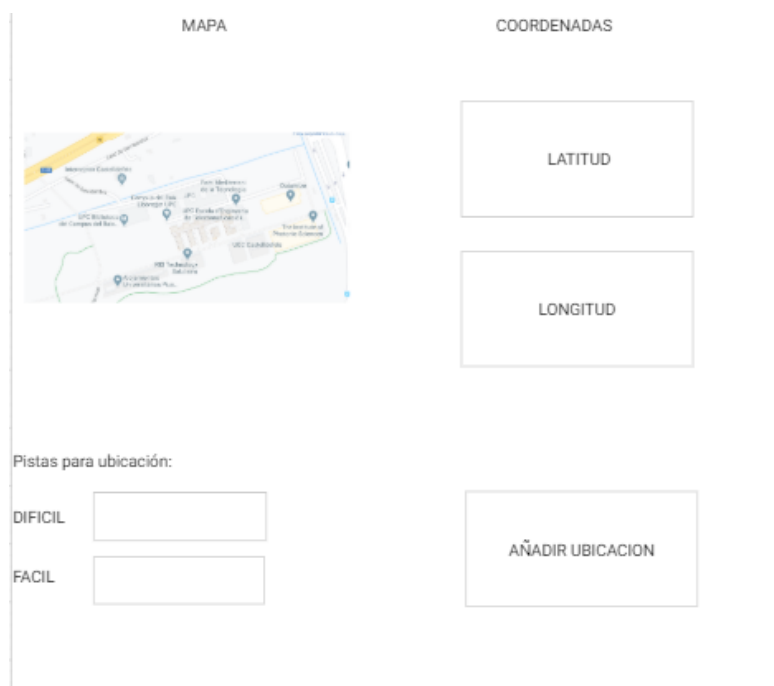


Fig 5.6. Pantalla de “AÑADIR UBICACIÓN” de Dashboard (Fuente: MockingBot)

- Pantalla “CREAR PREGUNTA”: En la pantalla “CREAR PREGUNTA” se puede definir una pregunta desde 0. El usuario debe seleccionar la dificultad de la pregunta (FACIL O DIFICIL) y la asignatura que le corresponde (MATEMÁTICAS, FÍSICA...).

Seguidamente se puede escribir el enunciado de la pregunta con sus posibles respuestas, y finalmente se selecciona la opción correcta. Después para guardar esa pregunta se debe seleccionar “AGREGAR”.

Además también se dispone de una pantalla “EDITAR PANTALLA” desde la cual se podrá editar la pregunta después de haberla guardado (Fig 5.7).

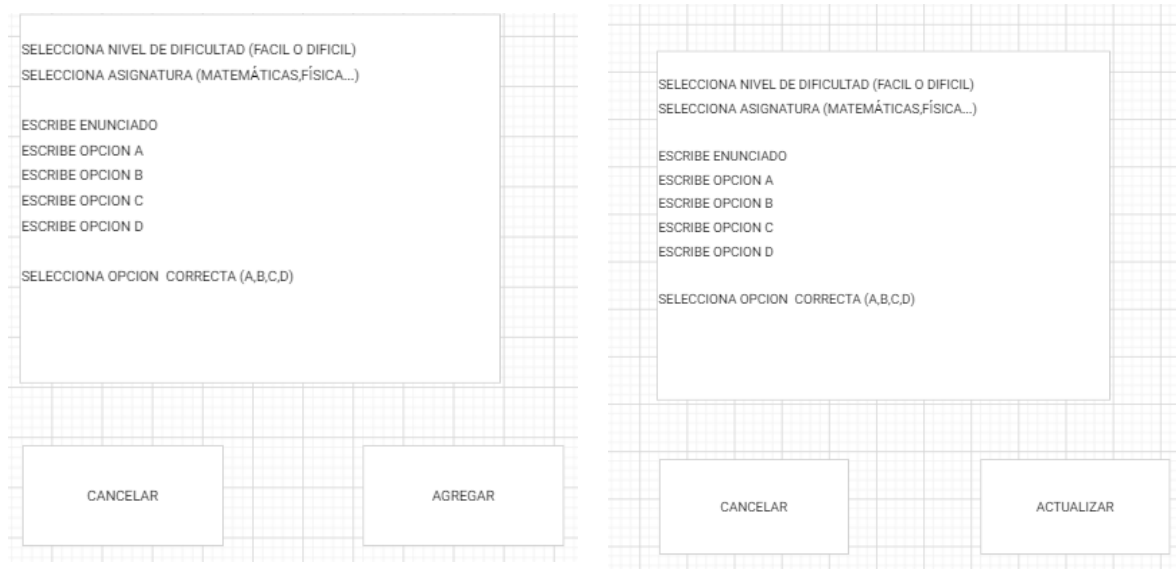


Fig 5.7. Pantalla de “CREAR PREGUNTA” y “EDITAR PREGUNTA” de Dashboard (Fuente: MockingBot)

- Pantalla “MIS PREGUNTAS”: En esta pantalla el usuario podrá ver las preguntas que ha realizado, de qué asignatura son, la dificultad que tienen y el enunciado de cada una. Además podrá eliminarlas en el botón “ELIMINAR” y editarlas en el botón “EDITAR” (Fig 5.8).

DIFICULTAD	ASIGNATURA	ENUNCIADO	
Facil	Física	Segunda Ley de Newton	<input type="button" value="ELIMINAR"/> <input type="button" value="EDITAR"/>

Fig 5.8. Pantalla de “MIS PREGUNTAS” de Dashboard (Fuente: MockingBot)

5.2. Pantallas del profesor

- Pantalla de inicio: La pantalla de inicio del profesor será exactamente igual que la pantalla de inicio del alumno. Desde ella podrá acceder a las asignaturas, juegos y colecciones, que podrá editar.
- Pantalla “JUEGOS”: De la misma forma que el alumno, el profesor visualizará los juegos activos e inactivos. El profesor podrá editar los juegos, activarlos y hacerlos visibles, eliminarlos o añadir juegos nuevos.

También podrá acceder a la clasificación general, de la misma manera que los alumnos.

Si el profesor pincha en el botón “EDITAR”, verá una nueva pantalla donde podrá modificar la penalización o obtención de puntos en el botón “AÑADIR PUNTOS”, cambiar las reglas en “REGLAS” y volver a la anterior pantalla si pincha en “VOLVER” (Fig 5.9).

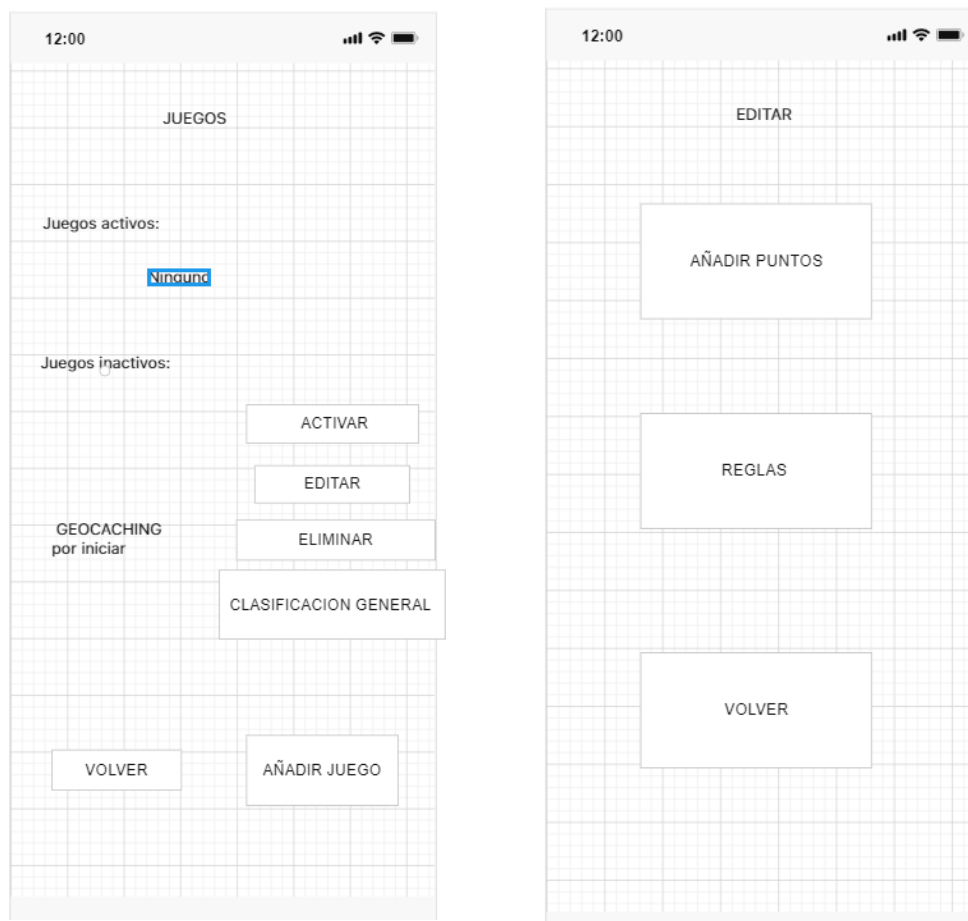


Fig 5.9. Pantalla de “JUEGOS” y “EDITAR” del profesor (Fuente: MockingBot)

- Pantalla “AÑADIR PUNTOS”: En la siguiente pantalla el profesor podrá introducir nuevos puntos a través de dos opciones: En el botón “MAPA” se abrirá un mapa donde indicando el punto deseado se guardará como una nueva coordenada, o bien en el botón “COORDENADAS” se introducirá la latitud y la longitud de dicho punto (Fig 5.10).

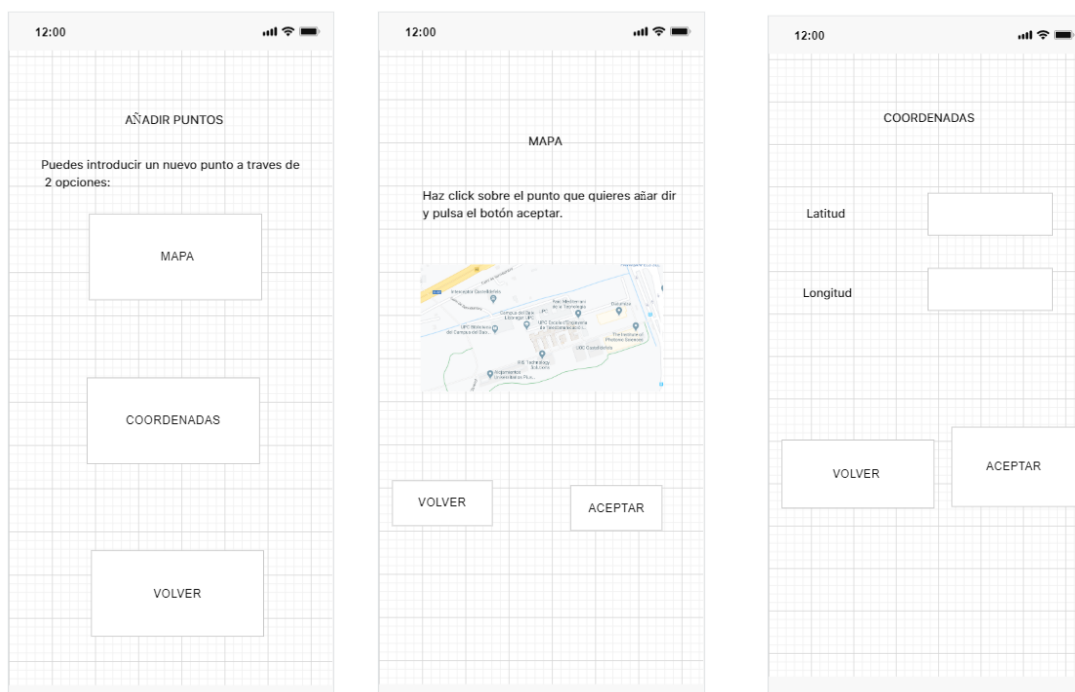


Fig 5.10. Pantalla de “AÑADIR PUNTOS”, “Mapa” y “Coordenadas” del profesor (Fuente: MockingBot)

- Pantalla “REGLAS”: En esta pantalla el profesor podrá editar diferentes parámetros entorno a las reglas del juego geocaching. Podrá establecer el número de puntos por pista difícil, por pregunta correcta, por error cometido y el aumento y penalización en porcentaje del bonus.

Para guardar los cambios tendrá que pinchar el botón “ACEPTAR”, o bien “VOLVER” si no quiere guardar las modificaciones (Fig 5.11).

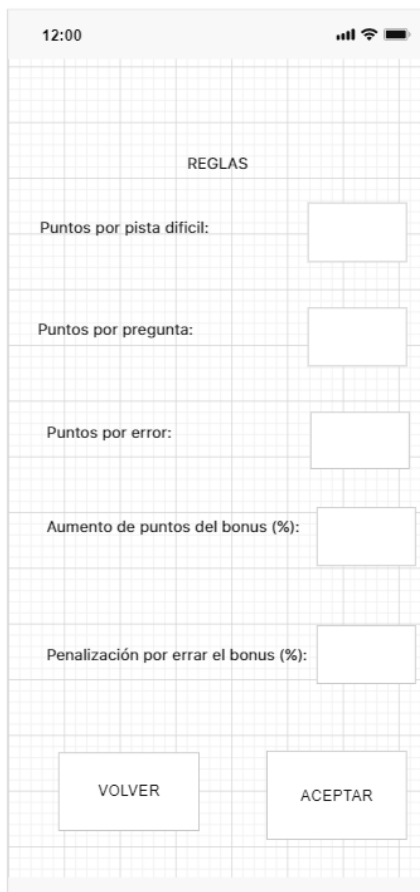


Fig 5.11. Pantalla de “REGLAS” del profesor (Fuente: MockingBot)

- Pantalla “CONFIRMAR ACTIVAR” y “CONFIRMAR ELIMINAR”: En caso de que el profesor desee activar o bien eliminar un juego, se le abrirá una de estas dos pantallas para confirmar la decisión tomada. En caso de que esté seguro del cambio pinchará el botón “SI”, y el botón “NO” en caso de que no quiera hacer dicha modificación (Fig 5.12).

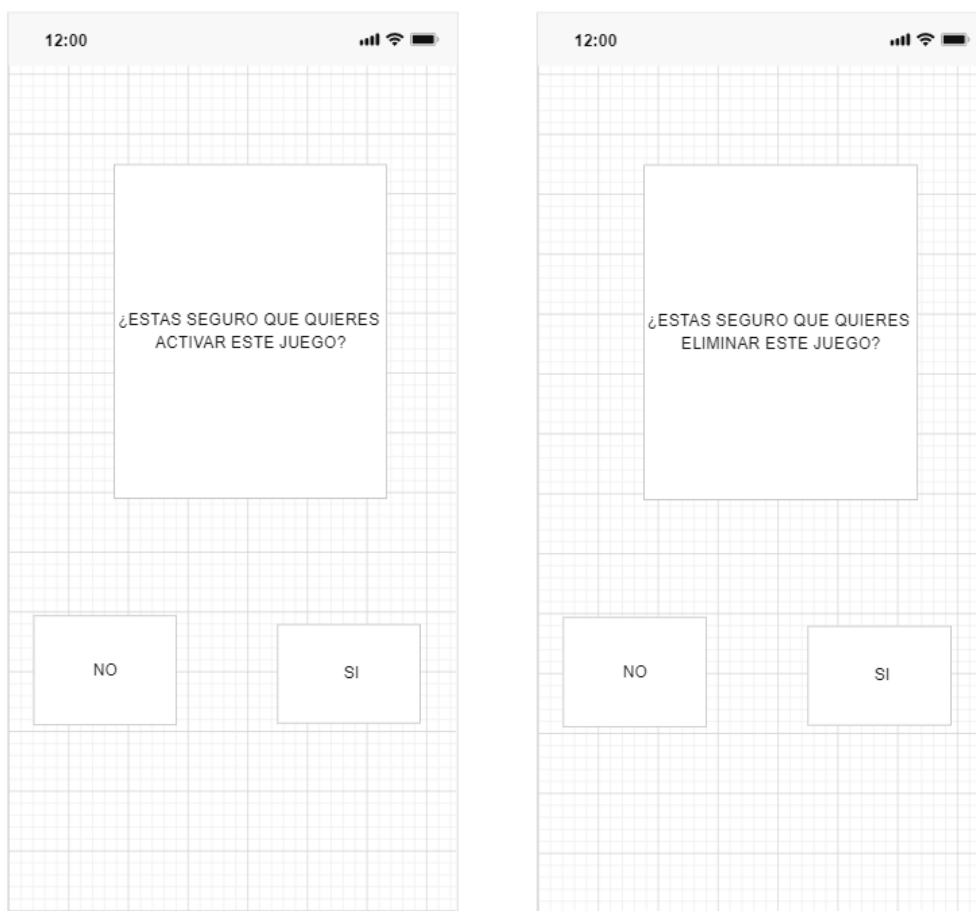


Fig 5.12. Pantalla de “CONFIRMAR ACTIVAR”, “CONFIRMAR ELIMINAR” del profesor (Fuente: MockingBot)

- Pantalla “CLASIFICACIÓN GENERAL”: La pantalla clasificación general del profesor será idéntica a la del alumno, mostrándole los mismos aspectos ya explicados anteriormente.

CAPÍTULO 6. MODELO DE DATOS

Una vez determinado el diseño inicial de las pantallas visto en el capítulo anterior, procede desarrollar el código necesario para que el juego de geocaching ideado funcione de la forma deseada.

Para ello, deberemos partir del repositorio existente *classpip-services*, el cual se hace responsable de almacenar toda la información útil y permitir la comunicación entre los diferentes componentes del software mediante una interfaz de programación de aplicaciones (API), más concretamente una API-REST, pues se usan URL-web.

La API usada está formada por un conjunto de modelos simples que engloban la totalidad de las funcionalidades actuales de Classpip, tal y como muestra la siguiente imagen (Fig 6.1).

LoopBack API Explorer		Token Not Set	accessToken	Set Access Token
Album	Show/Hide	List Operations	Expand Operations	
AlbumEquipo	Show/Hide	List Operations	Expand Operations	
Alumno	Show/Hide	List Operations	Expand Operations	
AlumnoJuegoDeColeccion	Show/Hide	List Operations	Expand Operations	
AlumnoJuegoDeCompeticionLiga	Show/Hide	List Operations	Expand Operations	
AlumnoJuegoDePuntos	Show/Hide	List Operations	Expand Operations	
AsignacionEquipo	Show/Hide	List Operations	Expand Operations	
AsignacionPuntoJuego	Show/Hide	List Operations	Expand Operations	
Coleccion	Show/Hide	List Operations	Expand Operations	
Cromo	Show/Hide	List Operations	Expand Operations	
EnfrentamientoLiga	Show/Hide	List Operations	Expand Operations	
Equipo	Show/Hide	List Operations	Expand Operations	
EquipoJuegoDeColeccion	Show/Hide	List Operations	Expand Operations	
EquipoJuegoDeCompeticionLiga	Show/Hide	List Operations	Expand Operations	
EquipoJuegoDePuntos	Show/Hide	List Operations	Expand Operations	
Grupo	Show/Hide	List Operations	Expand Operations	
HistorialPuntosAlumno	Show/Hide	List Operations	Expand Operations	
HistorialPuntosEquipo	Show/Hide	List Operations	Expand Operations	
Imagen	Show/Hide	List Operations	Expand Operations	
Insignia	Show/Hide	List Operations	Expand Operations	
Jornada	Show/Hide	List Operations	Expand Operations	
JornadaDeCompeticionLiga	Show/Hide	List Operations	Expand Operations	
Juego	Show/Hide	List Operations	Expand Operations	
JuegoDeColeccion	Show/Hide	List Operations	Expand Operations	
JuegoDeCompeticion	Show/Hide	List Operations	Expand Operations	
JuegoDeCompeticionLiga	Show/Hide	List Operations	Expand Operations	
JuegoDePuntos	Show/Hide	List Operations	Expand Operations	
Liga	Show/Hide	List Operations	Expand Operations	
Matricula	Show/Hide	List Operations	Expand Operations	
Nivel	Show/Hide	List Operations	Expand Operations	
Partido	Show/Hide	List Operations	Expand Operations	
Profesor	Show/Hide	List Operations	Expand Operations	
Punto	Show/Hide	List Operations	Expand Operations	
ReglaAutomaticaColeccion	Show/Hide	List Operations	Expand Operations	
ReglaAutomaticaCompeticion	Show/Hide	List Operations	Expand Operations	
ReglaManualColeccion	Show/Hide	List Operations	Expand Operations	
ReglaManualCompeticion	Show/Hide	List Operations	Expand Operations	

Fig 6.1. Modelos existentes en la API-REST (Fuente: localhost:3000/explorer)

No obstante, para el módulo de Geocaching, se necesitan usar solamente los modelos siguientes:

- Profesor.
- Alumno.
- Grupo.

Además, se deberán crear los siguientes modelos:

- Escenario.
- Punto Geolocalizable.
- JuegoDeGeocaching.
- Pregunta (usado de manera conjunta junto con el módulo de Cuestionarios, creado por otro desarrollador).

6.1. Creación de modelos

Para crear los modelos, será necesario seguir los siguientes pasos, usando como ejemplo la creación del modelo PuntoGeolocalizable.

- > Entrar en *classpip-services*.
- > Bajarse la versión más actual del proyecto mediante *git pull origin master*.
- > Abrir el terminal y ejecutar *lb model* con tal de iniciar la creación del modelo.
- > Seguir los pasos tal y como se indica en la imagen (Fig 6.2).

```
? Especifique el nombre de modelo: PuntoGeolocalizable
? Seleccione el origen de datos al que conectar PuntoGeolocalizable: baseDatosEnMemoria (memory)
? Seleccione la clase base del modelo PersistedModel
? ¿Exponer PuntoGeolocalizable mediante la API REST? Yes
? Formato plural personalizado (utilizado para crear el URL de REST): PuntosGeolocalizables
? ¿Modelo común o sólo servidor? común
```

Fig 6.2. Creación de modelos (Fuente: Visual Studio Code)

- > Acto seguido, habrá que implementar los atributos de dicho modelo (Fig 6.3).

```
Especifique un nombre de propiedad vacío cuando haya terminado.
? Especifique el nombre de propiedad: Latitud
? Tipo de propiedad: string
? ¿Es necesario? Yes
? Valor predeterminado [dejar en blanco para ninguno]:
```

Fig 6.3. Creación de atributos (Fuente: Visual Studio Code)

> Al terminar de añadir propiedades, deberemos dirigirnos al fichero punto-geolocalizable.json con tal de observar la correcta creación (Fig 6.4).

```
common > models > {} punto-geolocalizable.json > ...
1  {}
2  "name": "PuntoGeolocalizable",
3  "plural": "PuntosGeolocalizables",
4  "base": "PersistedModel",
5  "idInjection": true,
6  "options": {
7    "validateUpsert": true
8  },
9  "properties": {
10   "Latitud": {
11     "type": "string",
12     "required": true
13   },
14   "Longitud": {
15     "type": "string",
16     "required": true
17   },
18   "PistaFacil": {
19     "type": "string",
20     "required": true
21   },
22   "PistaDificil": {
23     "type": "string",
24     "required": true
25   }
26 },
27 "validations": [],
28 "relations": {},
29 "acls": [],
30 "methods": {}
31 }
```

Fig 6.4. Modelo PuntoGeolocalizable
(Fuente: Visual Studio Code)

6.2. Entidad-relación

Como muchos de estos modelos individuales están interrelacionados unos a otros, es necesario establecer qué relaciones existen entre ellos. Para ello, se define:

- Entidad: Objeto del mundo real que queremos almacenar datos. Cada entidad posee atributos propios que la definen. Además, deberá poseer un atributo único con tal de poder ser seleccionado de manera inequívoca.
- Relación: Asociación entre entidades para establecer su dependencia. Existen los siguientes tipos:
 - Relación 1:1: Las entidades se asocian una a una.
 - Relación 1:N/N:1: Una entidad está asociada a N entidades / N entidades están asociadas a una única entidad.
 - Relación N:M: N entidades se pueden relacionar a M entidades.

A continuación, se muestra y describe el diagrama de entidad-relación (Fig 6.5) necesario para el módulo de Geocaching.

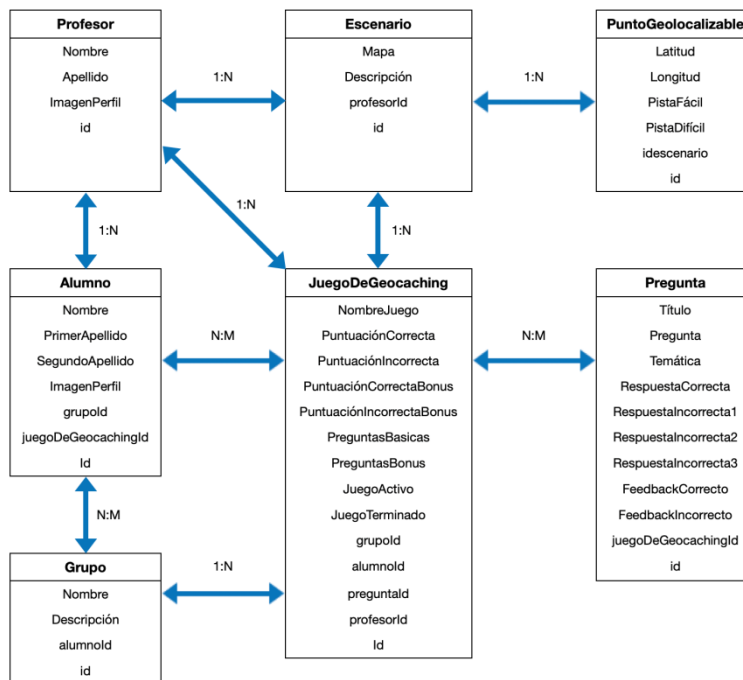


Fig 6.5. Modelos y relaciones de entidad de Geocaching (Fuente: propia)

Para el modelo Profesor, los atributos son: Nombre, Apellido, ImagenPerfil e id (que hace referencia al atributo clave). Además, observamos la relación que posee con el modelo Escenario y Juego que en ambos casos es 1:N, es decir, cada Profesor tiene asociados numerosos Escenarios y/o Juegos.

Para el modelo Escenario, los atributos son: Mapa, Descripción, idprofesor (con mismo valor que el atributo id del modelo Profesor) e id. Sus relaciones con PuntoGeolocalizable y Juego son 1:N, por lo que cada escenario puede contener varios PuntosGeolocalizable y cada Escenario puede ser reusado en diferentes juegos.

Para el modelo PuntoGeolocalizable, los atributos son: Latitud, Longitud, PistaFacil, PistaDifícil, id e idescenario (con tal de poder asignar el PuntoGeolocalizable a un Escenario específico). Este modelo tiene asociada una única relación con Escenario de 1:N, tal y como se ha descrito anteriormente.

Para el modelo Alumno, los atributos son: Nombre, PrimerApellido, SegundoApellido, ImagenPerfil e id. Este modelo posee una relación N:M con Juego, pues varios alumnos pueden tener disponer de varios juegos.

Para el modelo JuegoDeGeocaching, los atributos son: NombreJuego, PuntuaciónCorrecta, PuntuaciónIncorrecta, PuntuaciónCorrectaBonus, PuntuaciónIncorrectaBonus, PreguntasBasicas, PreguntasBonus, JuegoActivo, JuegoInactivo, idprofesor, idescenario e id. Es el modelo con mayor número de relaciones, pues tiene asociadas las relaciones 1:N con Profesor, Grupo y Escenario y relaciones N:M con Alumno y Pregunta.

Para el modelo Pregunta, los atributos son: id, Titulo, Pregunta, Temática, RespuestaCorrecta, RespuestaIncorrecta1, RespuestaIncorrecta2, RespuestaIncorrecta3, FeedbackCorrecto, FeedbackIncorrecto. Tiene una relación N:M con JuegoDeGeocaching, pues varios JuegoDeGeocaching distintos pueden tener múltiples Pregunta distintas.

Para las relaciones N:M se emplea el uso de “modelos puente”, tal y como se muestra a continuación (Fig 6.6).

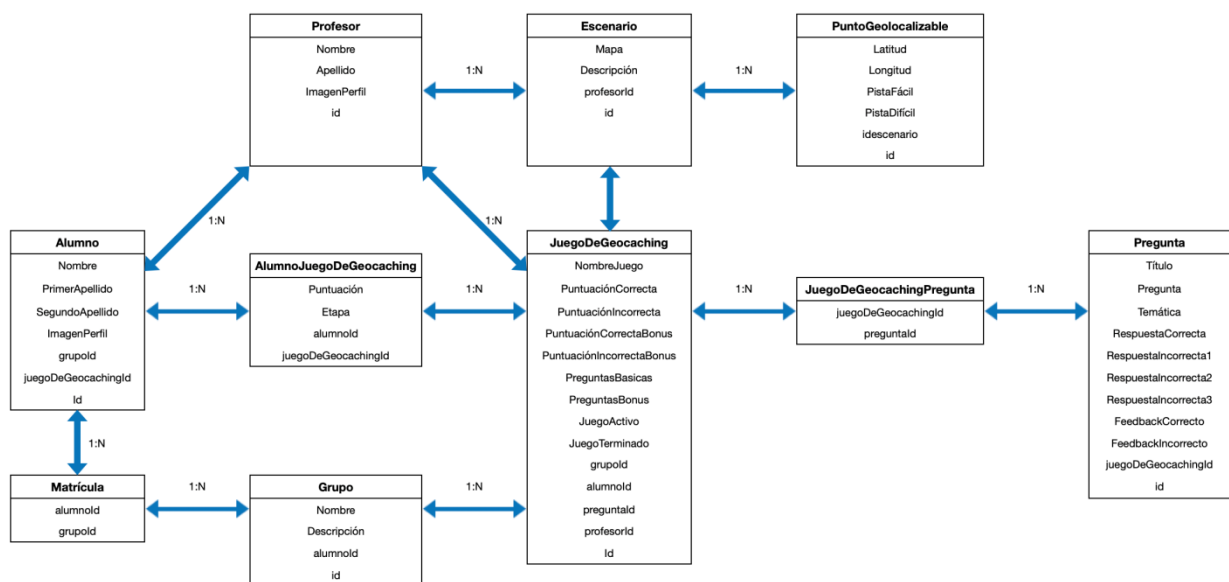


Fig 6.6. Modelos y relaciones de entidad de Geocaching con modelos puente (Fuente: propia)

En la imagen, observamos como gracias a la implementación de dos nuevos modelos llamados AlumnoJuegoDeGeocaching y JuegoDeGeocachingPregunta, somos capaces de convertir una N:M en dos 1:N, facilitando así las asociaciones entre modelos. Por lo tanto, este último esquema será el vigente.

El modelo `AlumnoJuegoDeGeocaching` contiene los atributos: `idjuego`, `idalumno`, `Puntos` y `Etapa`. Permitiendo que este modelo puente haga el seguimiento de cada alumno N durante cada etapa de cualquier juego M.

Finalmente, el modelo `JuegoDeGeocachingPregunta` contiene los atributos: `idjuego`, `idpregunta`, los cuales permitirá relacionar cada pregunta concreta a un juego de geocaching concreto.

6.3. Creación de relaciones

Para crear dichas relaciones en el repositorio de services, hará falta seguir los siguientes pasos:

>En el terminal, ejecutar `lb relation` para iniciar el proceso de creación.

>Para una relación 1:N, seguir los pasos mostrados con en la siguiente imagen (Fig 6.7), ejemplificados con la relación `Profesor` y `JuegoDeGeocaching`.

```
? Seleccione el modelo desde el que crear la relación: Profesor
? Tipo de relación: has many
? Seleccione un modelo con el que crear una relación: JuegoDeGeocaching
? Especifique el nombre de propiedad para la relación: juegosDeGeocachingloopback-datasource-juggler deprecated El método Scope "getAsync()" está en
desuso, utilice "find()" en su lugar. ../../../../../../usr/local/lib/node_modules/loopback-cli/node_modules/generator-loopback/lib/helpers.js:132:
37
? Especifique el nombre de propiedad para la relación: juegosDeGeocaching
? Opcionalmente, especifique una clave foránea personalizada: profesorId
```

Fig 6.7. Relación entre `Profesor` y `JuegoDeGeocaching` (Fuente: Visual Studio Code)

> Para las relaciones N:M, se deberá tener en cuenta el modelo puente cuando pida elegir un modelo definido. Para la relación `Alumno` y `JuegoDeGeocaching`, se debería realizar de la siguiente forma (Fig 6.8):

```
? Seleccione el modelo desde el que crear la relación: Alumno
? Tipo de relación: has many
? Seleccione un modelo con el que crear una relación: JuegoDeGeocaching
? Especifique el nombre de propiedad para la relación: JuegosDeGeocachingloopback-datasource-juggler deprecated El método Scope "getAsync()" está en
desuso, utilice "find()" en su lugar. ../../../../../../usr/local/lib/node_modules/loopback-cli/node_modules/generator-loopback/lib/helpers.js:132:
37
? Especifique el nombre de propiedad para la relación: JuegosDeGeocaching
? Opcionalmente, especifique una clave foránea personalizada: juegoDeGeocachingId
? ¿Requiere un modelo definido? Yes
? Elija un modelo definido: AlumnoJuegoDeGeocaching
? Permitir anidar la relación en las API REST: Yes
```

Fig 6.8. Relación entre `Alumno` y `JuegoDeGeocaching` through `AlumnoJuegoDeGeocaching` (Fuente: Visual Studio Code)

> Crear las relaciones: Alumno has many JuegosDeGeocaching through AlumnoJuegoDeGeocaching, mediante la clave juegoDeGeocachingId. JuegoDeGeocaching has many Alumno through AlumnoJuegoDeGeocaching, mediante la clave alumnoid.

> Para AlumnoJuegoDeGeocaching, el modelo puente, se deberán realizar relaciones de BelongsTo, tanto para el modelo Alumno como para el modelo JuegoDeGeocaching, tal y como se muestra (Fig 6.9)

```
? Seleccione el modelo desde el que crear la relación: AlumnoJuegoDeGeocaching
? Tipo de relación: belongs to
? Seleccione un modelo con el que crear una relación: Alumno
? Especifique el nombre de propiedad para la relación: (alumno) loopback-datasource-juggler deprecated El método Scope "getAsync()" está en desuso,
utilice "find()" en su lugar. ../../../../../../usr/local/lib/node_modules/loopback-cli/node_modules/generator-loopback/lib/helpers.js:132:37
? Especifique el nombre de propiedad para la relación: alumno
? Opcionalmente, especifique una clave foránea personalizada: alumnoid
```

Fig 6.9. Relación entre AlumnoJuegoDeGeocaching y Alumno (Fuente: Visual Studio Code)

CAPÍTULO 7. PRUEBAS Y EVALUACIÓN

En este apartado se recogen todas las pruebas a las que han sido sometidas las aplicaciones web y móvil con el fin de asegurar la robustez y el correcto funcionamiento. Además, también se ha hecho una evaluación a diferentes individuos y se muestran sus valoraciones.

Pruebas

Una vez completos los repositorios se subieron a GitHub haciendo un *git push* y para comprobar que todo funcionara correctamente se ejecutó una batería de pruebas para asegurar su robustez. A continuación se expone un listado con las funciones incorporadas y su funcionalidad.

En el Dashboard:

- Crear escenario: Permite crear con éxito un escenario. Para ello, obliga a asignar un nombre al mapa y una breve descripción, de no hacerlo un pequeño mensaje en rojo aparecerá advirtiéndolo.
- Crear punto geolocalizable: a diferencia de los campos de mapa y descripción comentados en el apartado anterior, este campo no será obligatorio. No obstante, para generar un punto geolocalizable y asociarlo a ese escenario únicamente será necesario darle un nombre, unas coordenadas (longitud y latitud), y asociarles las preguntas pertinentes (fácil y difícil). También aparecerá justo en el margen inferior una tabla con un resumen de los puntos agregados. A nivel de robustez cabe destacar que por ejemplo, en el caso de abandonar la ventana una alerta emergente aparecerá advirtiendo al usuario que no se guardarán los cambios. Es muy importante tener en cuenta, que las coordenadas deben ser introducidas en formato decimal, en caso contrario la aplicación móvil sería incapaz de calcular la distancia.
- Mis escenarios: permite ver una lista con los escenarios creados. Al pulsar sobre cualquiera, se muestra un desplegable con las opciones de eliminar y editar. En el caso de la opción eliminar, aparece un mensaje de confirmación y en el caso de el botón editar, conduce a una pantalla donde podremos cambiar el nombre y la descripción del escenario, así como sus puntos (tanto añadir como eliminar como editar).
- Lista de juegos: lista donde se muestran todos los juegos de geocaching creados. Se clasifican en tres grupos: activos, inactivos y preparados. Cuando un profesor crea un juego de geocaching este aparecerá como preparado. En el momento que el profesor desee empezar el juego, desde juegos preparados

pulsara sobre el juego y visualizará un botón para activarlo. Por último, si desea terminar un juego activo, pulsara sobre este y verá un botón de finalizar que pulsará. Independiente del estado en el que se encuentre un juego, una clasificación donde se recogen las puntuaciones de los alumnos se mostrará.

-Crear juego: se encuentra justo al lado de la pestaña lista de juegos. Para crear un juego de geocaching será necesario asignarle un nombre, seleccionar el tipo de juego de geocaching, indicar si se desea un modo de juego individual o por equipos, se asocie un escenario disponible y se añadan las preguntas tanto básicas como de bonus. Por último se deben establecer las reglas, en este caso las puntuaciones, tanto las que bonifican como las que penalizan, y confirmar la creación del juego. Al finalizar la creación se mostrara un mensaje de confirmación y si se cancela, también aparecerá una alerta para asegurar la decisión.

En la aplicación del profesor:

- Mis escenarios: pantalla donde se muestran los escenarios disponibles. Al pulsar sobre alguno de ellos, se mostrará otra pantalla con los puntos asociados al escenario y sus datos (nombre y coordenadas).

- Capturar coordenadas: botón que substituirá las coordenadas del punto por las correspondientes a la posición actual. Para llegar esta opción, se deberá entrar en mis escenarios y luego escoger el escenario donde se encuentre el punto que se quiere modificar.

- Mis juegos: pantalla donde se muestran todos los juegos clasificados por su estado (activo, inactivo y preparado).

- Juego activo: al pulsar sobre un juego de geocaching activo, se mostrará una pantalla con la clasificación del juego, con las puntuaciones y las etapas realizadas. Además se dispondrá de un filtro para buscar a un alumno en concreto y un botón rojo para finalizar el juego. Se muestra una alerta por pantalla avisando del proceso llevado a cabo correctamente.

- Juego inactivo: al pulsar sobre un juego de geocaching inactivo, se mostrará una pantalla con la clasificación del juego, como en el caso anterior, pero esta vez sin el botón lógicamente.

- Juego preparado: al pulsar sobre un juego de geocaching preparado, se mostrará una lista con los alumnos participantes y un botón en verde con la opción de activar el juego. Se muestra una alerta por pantalla avisando del proceso llevado a cabo correctamente.

Como conclusiones a este apartado, anotar que la aplicación del profesor funciona según lo exigido. Se ha procurado hacer la aplicación lo más amigable e intuitiva posible, con el fin de hacerla atractiva y divertida.

Evaluación

Acto seguido se ha procedido a hacer una encuesta a cinco personas de mi entorno sin ningún tipo de iniciación en Classpip, y se ha recogido en forma de gráfico sus opiniones acerca del nivel de dificultad.

La prueba consistió en un recorrido completo tanto por el Dashboard como por la aplicación, siguiendo como pauta las pruebas a las que fueron sometidas ambas plataformas. Al finalizar las pruebas, los participantes fueron sometidos a una encuesta sobre sus opiniones acerca de la facilidad de uso, el diseño gráfico y la valoración general que habían experimentado.

Entre los usuarios han coincidido en que el Dashboard les resultó más complicado claramente. Es lógico debido a que es más completo y está diseñado para recoger todas las opciones posibles de modificación. No obstante, en la aplicación del profesor, donde solo aparecen las opciones esenciales para el funcionamiento del juego, casi todos los encuestados fueron capaces de realizar todas las funciones con relativa facilidad.

Resultado sobre la facilidad de uso (Fig 7.1):

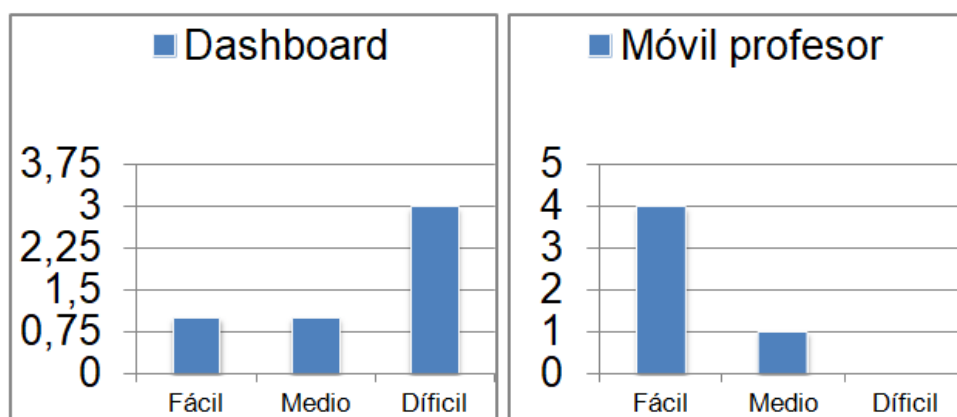


Fig 7.1. Resultados acerca de la facilidad de uso (Fuente: propia)

Resultado sobre el diseño gráfico (Fig 7.2):

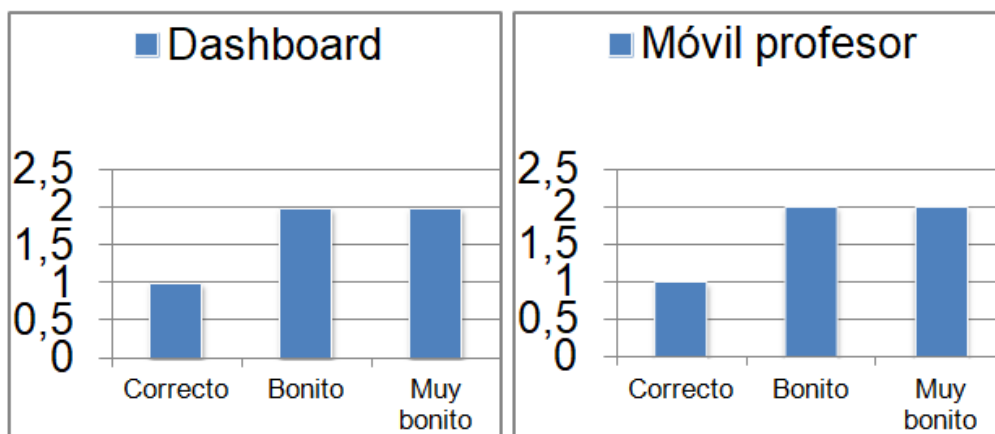


Fig 7.2. Resultados acerca del diseño gráfico (Fuente: propia)

Resultado sobre la valoración general (Fig 7.3):

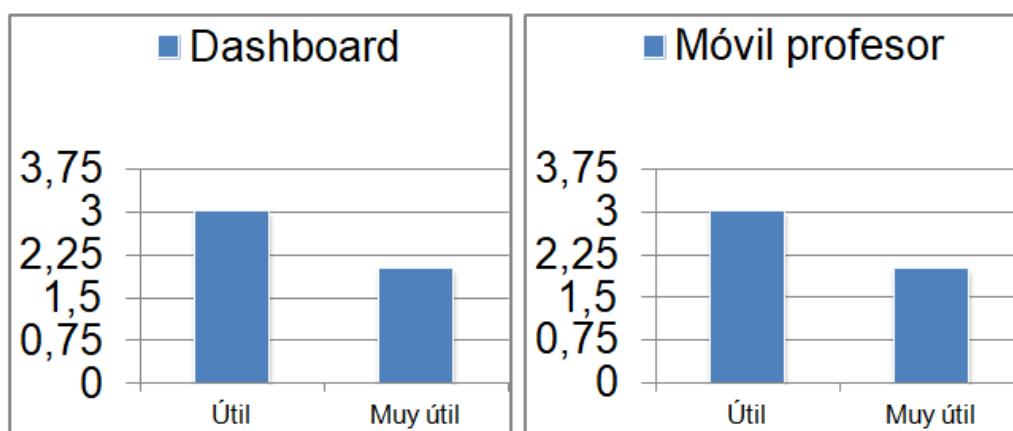


Fig 7.3. Resultados acerca de la valoración general (Fuente:propia)

CAPÍTULO 8. ESTUDIO DE PRECISIÓN

Para poder garantizar que el juego de geocaching fuese robusto y fiable, era necesario comprobar que las coordenadas insertadas manualmente por un profesor, no representaran una posición real muy alejada de la que luego interpretaría el GPS del móvil del alumno. Si esto no fuese así, se podría dar el caso que el profesor indicara unas coordenadas con la voluntad de que los alumnos acudieran a un punto en concreto y luego, por un error de precisión en el GPS, los alumnos fueran a otro lugar imprevisible.

Para ello se ha llevado a cabo un estudio de precisión basado en lo siguiente. Se han realizado dos estudios por separado, uno en un espacio cerrado (un parking) y otro en un espacio abierto (un descampado). Para cada uno de los casos se han realizado tres pruebas, utilizando tres distancias distintas, y se ha recogido la distancia que detectaba el móvil para así poder compararlas.

En primer lugar, se accedió al Dashboard con las credenciales del profesor y se creó un escenario y se asoció a este dos puntos geolocalizables. Después se ejecutó la aplicación para el móvil del profesor en un móvil Android personal con el propósito de capturar las coordenadas y sustituirlas en uno de los puntos de nuestro escenario. Por otro lado, acudimos a *GoogleMaps* y se recogieron las coordenadas de unos lugares escogidos estratégicamente y se substituyeron en el otro punto de nuestro escenario.

En un descampado, con el cielo descubierto y sin ningún tipo de obstáculo físico entre mi dispositivo móvil y el satélite GPS, se empezó el experimento. Se procedió a activar el juego puesto que después de crearlo su estado es juego preparado. Se hizo una simulación del juego y se aprovecho el hecho de tener que estar próximo al punto para poder responder a la pregunta mientras se visualizaba el parámetro de la distancia al punto en tiempo real que se había programado anteriormente.

No quedaba más que acercarse andando a cada uno de los puntos y comprobar si coincidían dichas distancias y ya además, se aprovechó para comprobar también si funcionaban las alertas correspondientes de acercamiento.

La precisión alcanzada fue todo un éxito para ambas situaciones, aproximadamente de un par o tres de metros. Los resultados fueron tan prometedores que probé de bajar la distancia mínima de la última alerta "Has llegado" de cinco a tres metros, y volví a repetir la operación funcionando esta óptimamente. Añadir que es más óptimo recoger las coordenadas con el móvil del profesor ya que se obtienen más decimales y experimentalmente, en el caso de *GoogleMaps* y cuando la alerta fue configurada a un metro, fue laborioso encontrar el punto. Cabe destacar que el sistema siempre mostraba variaciones constantes en una posición fija de un metro, con lo cual la alerta a esa distancia no tendría sentido ponerla.

Para analizar el caso en un espacio cubierto, se escogió como lugar el parking de un centro comercial. Se repitió el experimento como en el caso descubierto y se anotaron los resultados obtenidos, que sorprendentemente fueron muy similares al anterior estudio. En las tablas siguientes se recogen los datos obtenidos (Tabla 8.1, Tabla 8.2).

Distancia real	Distancia móvil	Alerta
10 ft (3 m)	[1.2, 3.8] m	1 m
5 ft (1.5 m)	[2.05, 2.6] m	1 m
3 ft (< 1 m)	[0.7, 1.76] m	1 m

Tabla 8.1. Tabla resultados espacio descubierto

Distancia real	Distancia móvil	Alerta
10 ft (3 m)	[0.8, 4.5] m	1 m
5 ft (1.5 m)	[2.12, 2.35] m	1 m
3 ft (< 1 m)	[0.87, 1.98] m	1 m

Tabla 8.2. Tabla resultados espacio cubierto

Como conclusión, es cierto que el sistema parece comportarse de manera más precisa en un espacio descubierto, no obstante la diferencia es mínima con lo que no creo que hubiera impedimento en utilizar la aplicación en lugares cerrados. En cuanto a la distancia de alerta, yo recomendaría entre tres y cuatro metros, ya que a esa distancia el sistema es fiable en cualquier escenario y es una distancia prudente como para no confundir lugares y así poder alcanzar el objetivo sin problemas.

Creo importante considerar también que mis medios fueron limitados (el experimento se realizó con un dispositivo móvil antiguo), con esto quiero expresar que si la aplicación requiriera más precisión se podría mejorar con un presupuesto mayor. En cualquier caso, para lo que está destinada y ha sido pensada creo que la aplicación es funcional con cualquier tipo de dispositivo móvil a nuestro alcance

CAPÍTULO 9. CONCLUSIONES

9.1. Conclusiones técnicas

El objetivo era desarrollar un modulo para el proyecto de Classpip sobre un juego de geocaching. Empecé desarrollando las ideas y escribiéndolas y luego tuve que implementarlas. Durante el proyecto, fuimos decidiendo que opciones implementar y de qué manera hacerlo y en general considero que se han cumplido las funciones generales. Es cierto que ha habido ideas que al final no he incorporado por diferentes razones, como por ejemplo conseguir hacer funcionar el server para actualizar las aplicaciones en tiempo real o el análisis para observar como se ha desarrollado el juego (ver las estadísticas de las preguntas, pistas...), pero como he dicho antes lo las funciones imprescindibles han sido implementadas con éxito. Visualmente también se han hecho mejoras a lo largo del trabajo, como puede ser darle colores a botones o cambiar tablas de posición, aunque siempre es mejorable como todo. A nivel general estoy contento porque creo que he cumplido el objetivo principal que me propuse.

9.2. Conclusiones personales

Al inicio del trabajo tengo que reconocer que me sentí un poco abrumado cuando tuvimos las primeras conversaciones con Miguel y Roc. Es cierto que nos advirtieron que la curva de aprendizaje era lenta al principio pero luego se aceleraba rápidamente. Empecé viendo videos que introducían a Angular y los primeras impresiones fueron un poco difíciles pero a medida que pasó el tiempo empecé a coger agilidad y mejoró la situación mejoró

A lo largo del proyecto he experimentado todo tipo de situaciones, tanto de satisfacción cuando conseguía arreglar un error en el código cuando llevaba días tras él, como de frustración cuando en algunos momentos no conseguía hacer lo que me estaba proponiendo o estaba muy apretado por la carga de trabajo debido a las demás asignaturas. Con eso me refiero a que a parte de los conocimientos que me he visto obligado a adquirir en programación en un lenguaje que no había visto nunca, o a desarrollar Angular con el que tampoco había trabajado, también este trabajo me ha ayudado a aprender a enfrentarme a situaciones en las cuales he tenido que gestionar la presión y el trabajo en momentos difíciles.

Otro punto importante que me gustaría destacar, es el trabajo cooperativo que hemos tenido entre los dos profesores que me han tutelado y Oriol con quien comparto parte de la memoria), que ha sido eficaz pero complicado y más en estos tiempos de COVID. No obstante creo que es un punto muy a favor de cara a un futuro de teletrabajo y telemático que nos espera.

Por último, también me ha agradado poder participar en el proyecto de Classpip en el cual han participado tantos estudiantes y algunos profesores, todos ellos coordinados y construyendo una plataforma de gamificación práctica e innovadora.

9.3. Propuestas futuras mejoras

Una de las cosas que más me llamo la atención de Classpip fue el dinamismo y la cooperación que tiene, y creo firmemente que tiene mucho potencial y proyección en un futuro próximo. En cuanto a mi modulo concretamente, opino como con el proyecto en general, así como lo he hecho yo cualquier otro compañero podrá recoger mi trabajo y mejorarlo o modificarlo para conveniencia de todos. Y todo eso es posible gracias a la herramienta GitHub, que permite compartir repositorios con facilidad y de manera organizada una vez que aprendes a utilizarlo.

ANEXOS

ANEXO I: Implementación

En este proyecto, se usa el estilo de arquitectura software Modelo Vista Controlador (MVC), cuya característica es que separa en tres componentes distintos los datos de la aplicación (Modelo), la interfaz del usuario (Vista) y la lógica de control (Controlador).

- Modelo: se encarga de los datos, consultas, búsquedas...
- Controlador: Se encarga de solicitar los datos al modelo y enviarlos a la vista, es decir, hace de intermediario entre el modelo y la vista.
- Vista: Se encarga de la representación visual de los datos.

El ciclo que realiza el MVC es:

1. Usuario realiza petición mediante algún mecanismo propio de la vista (ejemplo, clicar botón).
2. Controlador captura la petición del usuario.
3. Controlador llama al modelo.
4. Modelo interactúa con la base de datos y devuelve la información al controlador.
5. Controlador recibe la información y se la comunica a la vista.
6. Vista procesa y muestra la información de forma visual al usuario.

I.I. Clases

En cada repositorio, se crean clases que son usadas para que la aplicación sea capaz de interpretar los modelos, para ello, los mismos atributos entre ambos son coincidentes.

Todas las clases se encuentran en src/app/clases, en formato TypeScript (TS) (Fig I.1).


```

export class JuegoDeGeocaching {
  NombreJuego: string;
  PuntuacionCorrecta: number;
  PuntuacionIncorrecta: number;
  PuntuacionCorrectaBonus: number;
  PuntuacionIncorrectaBonus: number;
  PreguntasBasicas: number[];
  PreguntasBonus: number[];
  JuegoActivo: boolean;
  JuegoTerminado: boolean;
  id: number;
  profesorId: number;
  groupId: number;
  idescenario: number;

  // tslint:disable-next-line:max-line-length
  constructor(NombreJuego?: string, PuntuacionCorrecta?: number, PuntuacionIncorrecta?: number, PuntuacionCorrectaBonus?: nu
    this.NombreJuego = NombreJuego;
    this.PuntuacionCorrecta = PuntuacionCorrecta;
    this.PuntuacionIncorrecta = PuntuacionIncorrecta;
    this.PuntuacionCorrectaBonus = PuntuacionCorrectaBonus;
    this.PuntuacionIncorrectaBonus = PuntuacionIncorrectaBonus;
    this.PreguntasBasicas = PreguntasBasicas;
    this.PreguntasBonus = PreguntasBonus;
    this.JuegoActivo = JuegoActivo;
    this.JuegoTerminado = JuegoTerminado;
    this.profesorId = profesorId;
    this.groupId = groupId;
    this.idescenario = idescenario;
  }
}

```

Fig I.1. Clase JuegoDeGeocaching (Fuente: Visual Studio Code)

I.II. Componentes

Cada componente consta de 3 archivos: un CSS para definir los estilos gráficos (ejemplo, letra en negrita), un HTML encargado de dar forma a la interfaz gráfica y un TypeScript, el controlador, encargado de los datos y la lógica del componente. A continuación se muestran ejemplos de cada uno de los archivos (Fig I.2, Fig I.3, Fig I.4).

```

<!-- //////////////////////////////////// SI ESCOGEMOS JUEGO DE GEOCACHING //////////////////////////////////// -->
<div *ngIf="tipoDeJuegoSeleccionado === 'Juego De Geocaching' && tengoModo">
  <!--Escogemos el escenario-->
  <mat-step>
    <ng-template matStepLabel><div>Escenario</div></ng-template>
    <div>
      <button type="button" class="btn Agregar" (click)="AbrirDialogoAgregarEscenario()" style="width: auto;">Añadir es
        post_add </i></button>
    </div>
    <div class="btnCambiarStep">
      <button type="button" class="btn Back" matStepperPrevious>Atrás</button>
      <button [disabled]="!tengoEscenario" type="button" class="btn Next" matStepperNext>Siguiente</button>
    </div>
  </mat-step>

```

Fig I.2. Estilos gráficos del componente CSS de Juego (Fuente: Visual Studio Code)

```
.radio-group {
  display: flex;
  flex-direction: column;
  margin: 15px 0;
}

.radio-button {
  margin: 5px;
  color:  black;
}
```

Fig I.3. Componente HTML de Juego (Fuente: Visual Studio Code)

```
// @ts-nocheck
AbrirDialogoAgregarEscenario(): void {
  const dialogRef = this.dialog.open(AsignaEscenarioComponent, {
    width: '70%',
    height: '80%',
    position: {
      top: '0%'
    },
  },
  // Pasamos los parametros necesarios
  data: {
    profesorId: this.profesorId
  }
  );
  dialogRef.afterClosed().subscribe(() => {
    this.escenario = this.sesion.DameEscenario();

    console.log('ESCENARIO SELECCIONADO --->' + this.escenario.Mapa);
    this.DamePuntosGeolocalizablesDelEscenario(this.escenario);
    console.log(this.numeroDePuntosGeolocalizables);
    console.log(this.puntosgeolocalizablesEscenario);
  });
}

DamePuntosGeolocalizablesDelEscenario(escenario: Escenario) {
  console.log('voy a mostrar los puntosgeolocalizables del escenario ' + escenario.id);
  this.peticionesAPI.DamePuntosGeolocalizablesEscenario(escenario.id)
  .subscribe(res => {
    if (res[0] !== undefined) {
      this.puntosgeolocalizablesEscenario = res;
    }
  });
}
```

Fig I.4. Componente Typescript de Juego (Fuente: Visual Studio Code)

I.III. Servicios

Con los servicios, obtenemos acceso a los datos del servidor sin tener que hacerlo en los componentes, con tal de evitar la sobrecarga de información y tener el código mejor estructurado.

I.III.I. Peticiones a la API

Este servicio incluye cualquier solicitud a la base de datos con peticiones http, para obtener, modificar y/o eliminar datos. Este proceso se lleva a cabo siguiendo los siguientes pasos: en primer lugar se declaran las variables que conducirán a las URL que se necesitaran y posteriormente se utilizara una función para solicitar la información que se necesite (Fig I.5, Fig.I.6). Ejemplos de dichas funciones son:

- GET: devuelve la información solicitada.
- POST: crea nueva información.
- PUT: modifica la información.
- DELETE: elimina la información.

```
private APIUrlJuegoDeGeocaching = this.base + '3000/api/JuegosDeGeocaching';  
private APIUrlAlumnoJuegoDeGeocaching = this.base + '3000/api/AlumnosJuegoDeGeocaching';  
private APIUrlEscenarios = this.base + '3000/api/Escenarios';  
private APIUrlPreguntas = this.base + '3000/api/Preguntas';
```

Fig I.5. URL del servicio Peticiones a la API (Fuente: Visual Studio Code)

GET	/JuegosDeGeocaching	Find all instances of the model matched by filter from the data source.
PUT	/JuegosDeGeocaching	Replace an existing model instance or insert a new one into the data source.
POST	/JuegosDeGeocaching	Create a new instance of the model and persist it into the data source.

Fig I.6. Ejemplos de funciones para JuegosDeGeocaching (Fuente: Visual Studio Code)

I.III.II. Cálculos

Este servicio tiene como objetivo hacer modificaciones sobre datos obtenidos desde el servicio de Peticiones a la API (Fig I.7).

```
private EliminarAlumnosJuegoDeGeocaching() {
    // Pido los alumnos correspondientes al juego que voy a borrar
    this.peticionesAPI.DameAlumnosDelJuegoDeGeocaching(this.sesion.DameJuego().id)
    .subscribe( AlumnosDelJuego => {
        if (AlumnosDelJuego[0] !== undefined) {

            // Una vez recibo las inscripciones, las voy borrando una a una
            // tslint:disable-next-line:prefer-for-of
            for (let i = 0; i < AlumnosDelJuego.length; i++) {
                this.peticionesAPI.BorraAlumnoDelJuegoDeGeocaching(AlumnosDelJuego[i].id)
                .subscribe(() => {
                    console.log('Inscripcion al juego borrada correctamente');
                });
            }
        } else {
            console.log('No hay alumnos en el juego de geocaching');
        }
    });
}
```

Fig I.7. Servicio de Cálculos (Fuente: Visual Studio Code)

I.III.III. Sesión

Este servicio hace posible reutilizar las constantes obtenidas en el pasado y que no han vuelto a ser usadas, de tal manera que cuando vuelvan a ser requeridas no hará falta volverlas a buscar pues se encontrarán en la Sesión (Fig I.8).

```
public TomaPuntoGeolocalizable(puntogeolocalizable: PuntoGeolocalizable) {
    this.puntogeolocalizable = puntogeolocalizable;
}

public TomaPuntosGeolocalizables(puntosgeolocalizablesEscenario: PuntoGeolocalizable[]) {
    this.puntosgeolocalizables = puntosgeolocalizablesEscenario;
}

public DamePuntosGeolocalizables(): PuntoGeolocalizable[] {
    return this.puntosgeolocalizables;
}
```

Fig I.8. Funciones dentro del servicio de Sesión (Fuente: Visual Studio Code)

I.IV. Datos cuantitativos

En este apartado se va a recoger todas las líneas que se han incorporado al código y también se especificará las clases y componentes que se han añadido.

Dashboard:

- Se han implementado las siguientes clases:
 - AlumnoJuegoDeGeocaching.ts (15 líneas)
 - Escenario.ts (12 líneas)
 - JuegoDeGeocaching.ts (34 líneas)
 - JuegoDeGeocachingPregunta.ts (11 líneas)
 - PuntoGeolocalizable.ts (18 líneas)
 - TablaAlumnoJuegoDeGeocaching.ts (20 líneas)

Total líneas: 110 líneas.

- Se han implementado los siguientes componentes:
 - crear-escenario (457 líneas)
 - mis-escenarios (189 líneas)
 - editar-escenario (282 líneas)
 - agregar-puntogeolocalizable-dialog (411 líneas)
 - editar-puntogeolocalizable-dialog (182 líneas)
 - asigna-escenario (140 líneas)
 - asigna preguntas (140 líneas)
 - juego-de-geocaching-seleccionado-activo (237 líneas)
 - informacion-juego-de-geocaching-dialog (347 líneas)
 - juego-de-geocaching-seleccionado-inactivo (235 líneas)
 - juego-de-geocaching-seleccionado-preparado (84 líneas)

Total líneas: 2704 líneas.

Móvil profesor: se han implementado las siguientes clases:

- Se han implementado las siguientes clases:
 - AlumnoJuegoDeGeocaching.ts (15 líneas)
 - Escenario.ts (14 líneas)
 - JuegoDeGeocaching.ts (32 líneas)
 - PuntoGeolocalizable.ts (21 líneas)
 - TablaAlumnoJuegoDeGeocaching.ts (20 líneas)

Total líneas: 102 líneas.

- Se han implementado los siguientes componentes:
 - editar-mis-puntosgeolocalizables (132 líneas)
 - juego-seleccionado-inactivo (157 líneas)
 - juegoseleccionadopreparado (162 líneas)
 - misescenarios (127 líneas)
 - mipuntosgeolocalizables (148 líneas)

Total líneas: 726 líneas.

I.V. Resultados

I.V.I. Sesión

Hemos creado en el Dashboard el apartado Escenarios, dentro de Recursos. En este, se podrán distinguir dos opciones, Ver y Crear.

En primer lugar, si seleccionamos la opción de Ver, visualizaremos una lista con los escenarios disponibles y si escogemos uno tendremos la posibilidad tanto de editarlo como de eliminarlo (Fig I.9, I.10).

Recursos



Fig I.9. Pantalla Recursos (Fuente: Dashboard)

Mis escenarios

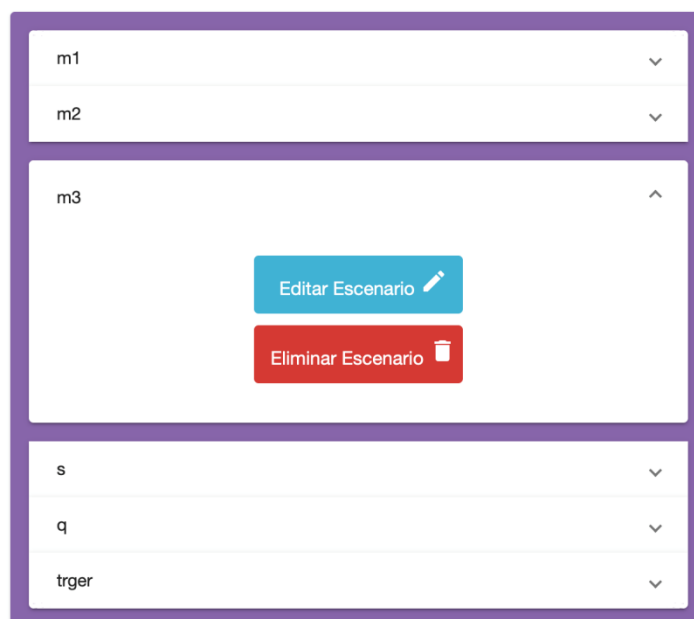


Fig I.10. Pantalla Mis Escenarios (Fuente: Dashboard)

Si elegimos Editar Escenario, se mostrará la siguiente ventana, donde podremos cambiar el nombre y la descripción del escenario (al editar dichos parámetros aparecerá un botón para poder aplicarlos) y además también será posible añadir, editar y eliminar los puntos geolocalizables asociados a dicho escenario (Fig I.11).

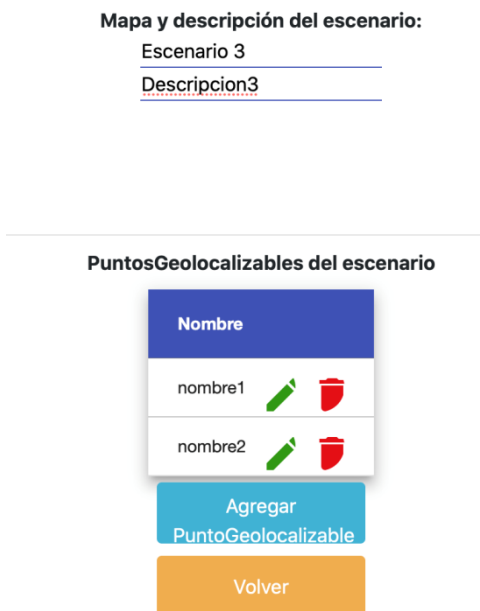


Fig I.11. Pantalla Editar Escenario (Fuente: Dashboard)

Para añadir un punto, debemos presionar el botón Agregar PuntoGeolocalizable, y al hacerlo, se mostrará el siguiente listado de parámetros que caracterizan al nuevo punto a añadir. Si se desea vaciar los campos para comenzar de nuevo, el botón Limpiar Campos dejará los campos por defecto (Fig I.12).

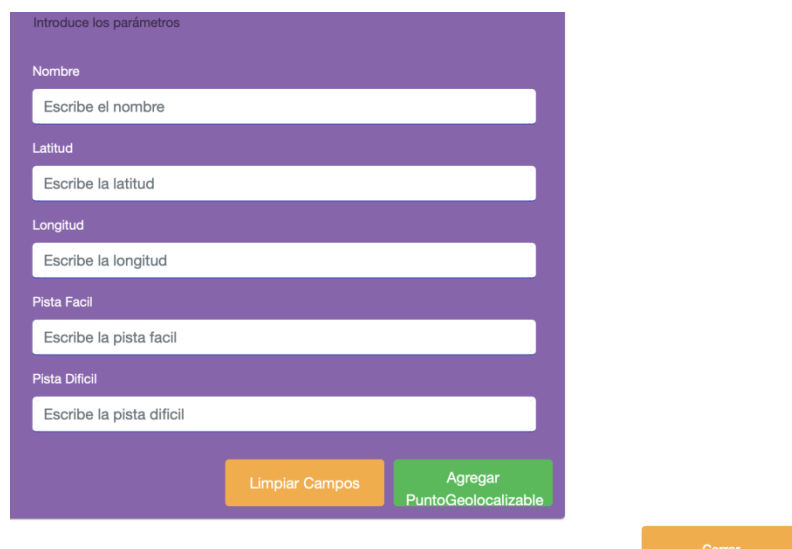


Fig I.12. Pantalla Agregar PuntoGeolocalizable
(Fuente: Dashboard)

También hemos añadido funcionalidades en el apartado Grupos, el cual nos ofrece dos opciones, Mis Grupos y Crear Grupo (Fig I.13).



Fig I.13. Pantalla principal del Dashboard (Fuente: Dashboard)

En Mis Grupos encontraremos una lista de los grupos disponibles, con su nombre y una pequeña descripción asociada (Fig I.14).

Mis Grupos

Nombre	Descripción
Grupo 1	Física 1ºESO
Grupo 2	Catalán 2º ESO
Prueba asignacion puntos	a
Demo avatares	Demo avatares
Prueba crear juegos	prueba
AAA	AAA
Sabado	Sabado

Fig I.14. Pantalla Mis Grupos (Fuente: Dashboard)

Si hacemos clic sobre uno de los grupos, accederemos a la ventana mostrada en la siguiente imagen, donde encontraremos las siguientes opciones: Editar Grupo, Eliminar Grupo, Sesiones de clase, Equipos y Juegos. También encontraremos un filtro para buscar un alumno en concreto, una tabla donde encontramos a los integrantes de dicho grupo y por último un botón para volver a la pantalla anterior (Fig I.15).

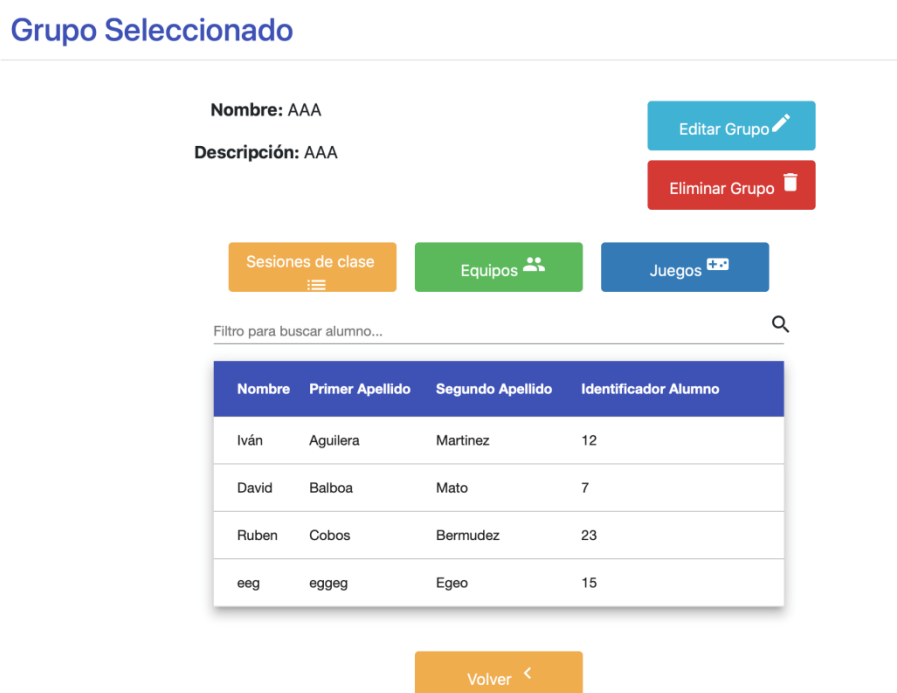


Fig I.15. Pantalla Grupo Seleccionado (Fuente: Dashboard)

A continuación explicaremos detalladamente cada una de las siguientes opciones dentro de nuestro grupo seleccionado:

-Editar grupo: en esta opción podremos editar el nombre o la descripción del grupo, seleccionar a uno o varios alumnos y eliminarlos (o todos a la vez, haciendo clic en el primer cuadrado seleccionable) o añadir un nuevo alumno (Fig I.16).

Nombre: AAA

Descripción: AAA

Aceptar Cambios ✓

Añadir Alumno +

Eliminar Alumno

Filtro para buscar alumno...

<input type="checkbox"/>	Nombre	Primer Apellido	Segundo Apellido	Identificador Alumno
<input type="checkbox"/>	Iván	Aguilera	Martinez	12
<input type="checkbox"/>	David	Balboa	Mato	7
<input type="checkbox"/>	Ruben	Cobos	Bermudez	23
<input type="checkbox"/>	eeg	eggeg	Egeo	15

Volver

Fig I.16. Pantalla Editar Grupo (Fuente: Dashboard)

Si hemos seleccionado la opción de Añadir Alumno, se nos mostrará una nueva ventana donde podremos ver los alumnos que ya están en el grupo y eliminarlos, y también los alumnos restantes y añadirlos (Fig I.17).

Agregar alumnos al grupo

Alumnos que ya estan en el grupo

Nombre	Primer Apellido	Segundo Apellido	
Iván	Aguilera	Martinez	
David	Balboa	Mato	
Ruben	Cobos	Bermudez	
eeg	eggeg	Egeo	

Mis alumnos

Filtro para buscar alumno...

Nombre	Primer Apellido	Segundo Apellido	
Iván	Aguilera	Martinez	
Iván	Aguilera	Martinez	

Fig I.17. Pantalla Añadir Alumno (Fuente: Dashboard)

-Eliminar Grupo: una ventana emergente aparecerá preguntándonos si estamos seguros de elegir esa opción (Fig I.18).

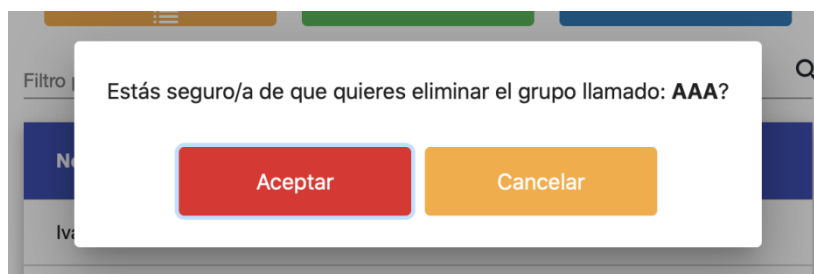


Fig I.18. Mensaje alerta Eliminar Grupo (Fuente: Dashboard)

- Juegos: aquí encontraremos dos apartados, lista de juegos y crear juego. En la lista de juegos estarán los diferentes juegos organizados en tres grupos: activos, inactivos y preparados.

Los juegos activos serán aquellos que estén disponibles en ese momentos y los inactivos lo contrario, en cambio cuando un profesor diseñe un juego y lo cree, pasará al apartado de preparados donde esperará a ser activado cuando lo desee su creador (Fig I.19, I.20).

Juegos

Lista de juegos Crear juego

Juegos Activos

- Puntos
Juego De Puntos: Individual
- Compositores
Juego De Colección: Individual
- Sherlock
Juego De Colección: Individual
- Masestros del ajedrez
Juego De Colección: Individual
- Mafalda
Juego De Colección: Individual
- Avatares
Juego De Avatar: Individual

Juegos Inactivos

- Compositores
Juego De Colección: Individual

Fig I.19. Pantalla Juegos Activos (Fuente: Dashboard)

Para crear un juego, lo primero que habrá que elegir será el tipo de juego, que en nuestro caso será Juego De Geocaching (Fig I.21).

Juegos

Lista de juegos Crear juego

1 Nombre 2 Tipo 3 Modo

Selecciona un tipo de juego:

- Juego De Puntos
- Juego De Colección
- Juego De Competición
- Juego De Avatar
- Juego De Cuestionario
- Juego De Geocaching

Has elegido ...

AtrásSiguiente

Fig I.20. Pantalla Crear Juego (Tipo) (Fuente: Dashboard)

Juegos Inactivos

- ↻ *Compositores*
Juego De Colección: Individual

- ↻ *Sherlock*
Juego De Colección: Individual

- ↻ *sherlock*
Juego De Colección: Individual

Juegos Preparados

No hay juegos preparados

Aviso Legal | PrivacidadClasspip © 2016 - 2019

Fig I.21. Pantalla Juegos Inactivos/Preparados (Fuente: Dashboard)

A continuación elegiremos el tipo (individual o por equipos). Cabe destacar, que para un juego de geocaching, esta decisión no tiene efecto pues el juego está diseñado para ser jugado de forma individual. Existe dicha opción, porque la plantilla de creación es la misma para todos los juegos (Fig I.22).

Juegos

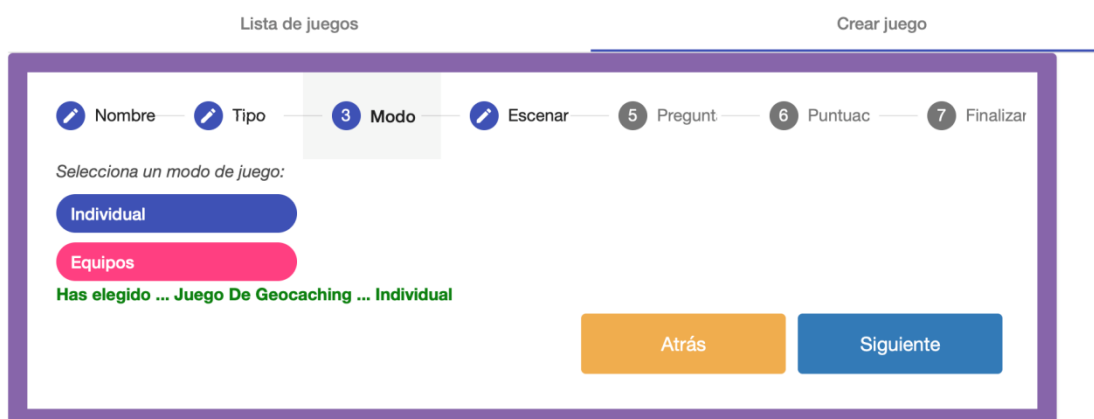


Fig I.22. Pantalla Crear Juego (Modo) (Fuente: Dashboard)

El siguiente paso es elegir el escenario para nuestro juego, para ello pulsaremos sobre Añadir Escenario y veremos una ventana donde aparecerán los escenarios disponibles con una breve descripción, y además un filtro para buscar uno en concreto. Como siempre veremos un botón de cerrar para volver a la pantalla anterior (Fig I.23, I.24).

Juegos

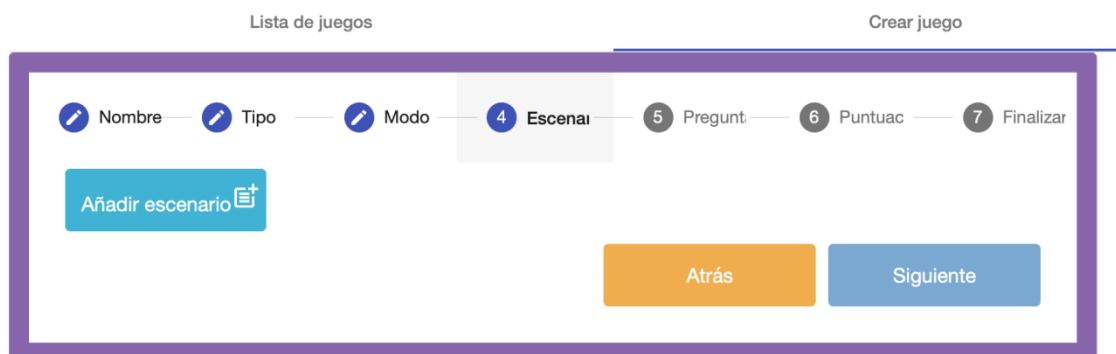


Fig I.13. Pantalla Crear Juego (Añadir Escenario) (Fuente: Dashboard)

Escenarios

Filtro para buscar escenarios...



Mapa	Descripcion	
m1	d1	+
m2	d2	+
Escenario 3	Descripcion3	+
s	s	+
q	q	+
trger	fgs	+

Cerrar

Fig I.24. Ventana emergente Añadir escenario (Fuente: Dashboard)

Añadir las preguntas pertinentes será el siguiente paso, al pulsar Añadir Preguntas abriremos una ventana emergente donde veremos el listado de preguntas disponibles con su título, la pregunta, la temática, y la opción de añadirla como pregunta básica o de bonus. También encontraremos un breve resumen en la parte superior sobre las selecciones que hemos realizado y el número total de estas. Considerar que, en nuestro caso (juego de geocaching), el número de preguntas básicas debe ser igual al número de preguntas con bonus y de no ser así no aparecerá el botón de aceptar (Fig I.25, I.26).

Juegos

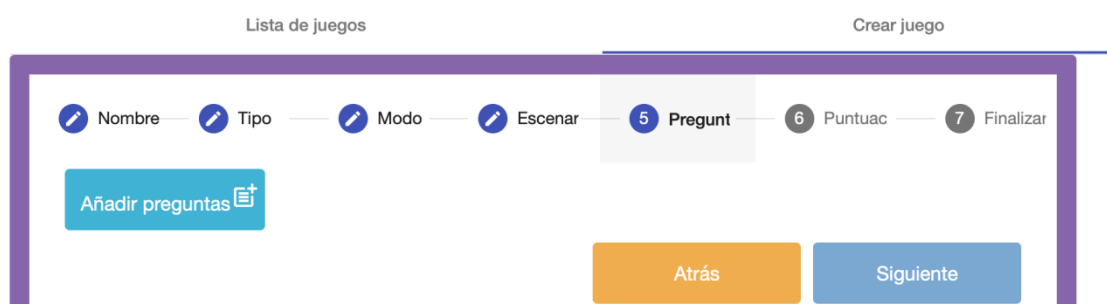


Fig I.25. Pantalla Preguntas (Fuente: Dashboard)

Agregar preguntas al Juego

Selecciona un total de 1 Preguntas Basicas y 1 Preguntas Bonus

Preguntas Basicas seleccionadas

Preguntas Bonus seleccionadas

Mis preguntas

Filtro para buscar pregunta...



Titulo	Pregunta	Tematica	Basicas	Bonus
2	1	1	+	+
6	6	6	+	+
aa	aaa	aaa	+	+
Color de algo	¿Cual es el color del mar?	Biologia	+	+
Levantamiento de peso	¿Que animal puede levantar 1000 veces su peso?	Biologia	+	+
Ciencia de primero	¿Cual es el color que se forma a partir de todos los colores?	Ciencia	+	+

Fig I.26. Ventana emergente Añadir Preguntas (Fuente: Dashboard)

Continuaremos introduciendo las puntuaciones y las penalizaciones que sufrirá el alumno debido a sus acciones (Fig I.27).

Juegos

Lista de juegos Crear juego

Nombre Tipo Modo Escenar Pregunt **6 Puntuac** 7 Finalizar

Introduzca la puntuacion por respuesta correcta

Introduzca la puntuacion por respuesta incorrecta (si no quiere restar ponga 0)

Introduzca el porcentaje de bonificación por respuesta bonus correcta

Introduzca el porcentaje de penalización por respuesta bonus incorrecta

Atrás Siguiente

Fig I.27. Pantalla Puntuaciones (Fuente: Dashboard)

Para finalizar el proceso de creación de un juego, una opción de confirmación aparecerá y de no desear crear dicho juego tendremos la opción de retroceder (Fig I.28).

Juegos

Lista de juegos Crear juego

Nombre Tipo Modo Escenar Pregunt Puntuac **7 Finaliza**

Atrás Crear juego de geocaching

Fig I.28. Pantalla Finalización (Fuente: Dashboard)

I.V.II. Funcionalidades del móvil profesor

En este apartado, se demuestran las funcionalidades realizadas en la aplicación móvil del profesor, que es donde los usuarios que accedan como profesor podrán modificar y visualizar todas las opciones del juego de geocaching.

En primer lugar, aparecerá la pantalla de acceso a la aplicación y se deberá ingresar con la identificación de un profesor, en caso contrario el acceso se denegará con el mensaje correspondiente (Fig I.29).

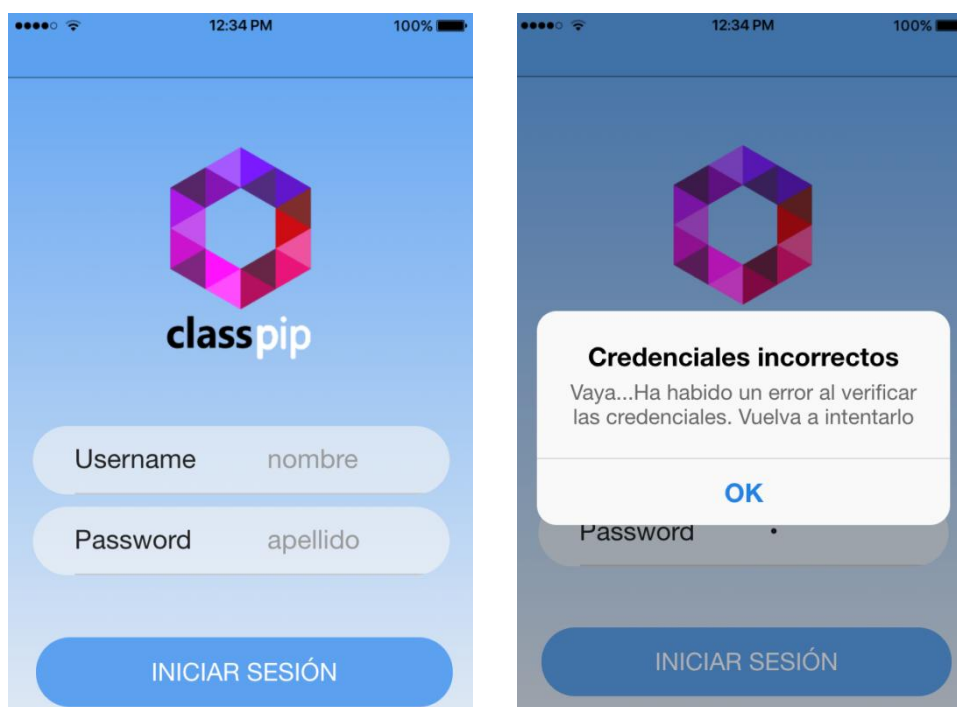


Fig I.29. Pantalla Inicio/Credenciales erróneas (Fuente: Aplicación del profesor)

Una vez que se acceda a la aplicación, lo primero que se encontrará será el apartado de Mis Grupos. Podemos ver una lista con los grupos disponibles y al hacer clic sobre uno de ellos, se abrirá un desplegable donde podremos acceder con facilidad a los siguientes campos: Mis Alumnos, Mis Equipos y Mis Juegos (Fig I.30).

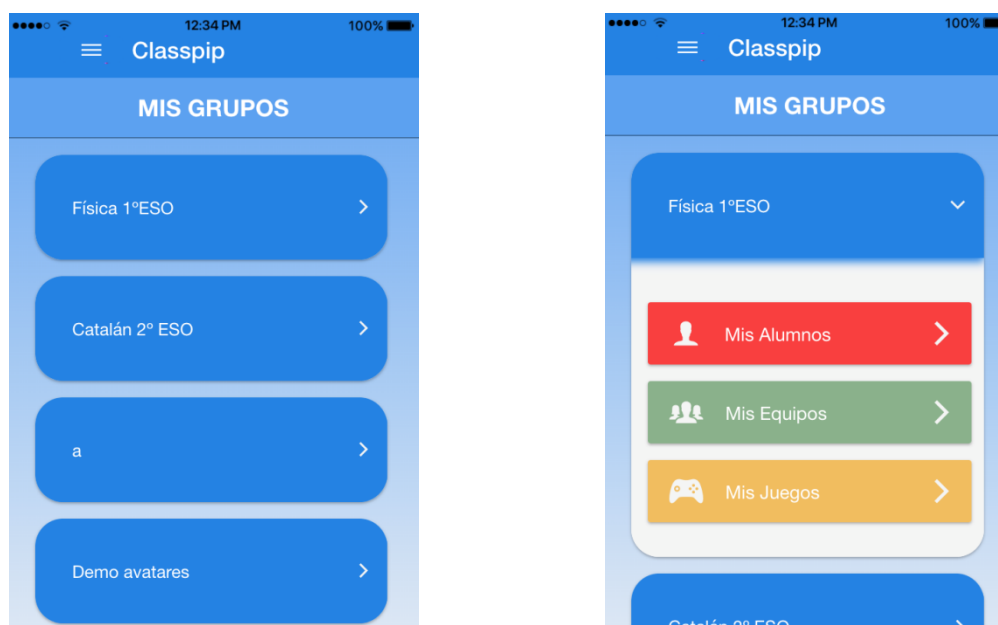


Fig I.30. Pantalla Mis Grupos y pantalla con un grupo seleccionado
(Fuente: Aplicación del profesor)

Si entramos en Mis Juegos, aparecerán tres grupos distinguiendo los juegos por su estado, que pueden ser activos, inactivos y preparados (Fig I.31).



Fig I.31. Pantalla Mis Juegos Activos e Inactivos/Preparados
(Fuente: Aplicación del profesor)

Si se hace clic sobre un juego de geocaching activo el profesor accederá a una ventana donde podrá ver la clasificación de los alumnos, un filtro para buscar un alumno en concreto y un botón rojo para finalizar el juego cuando lo desee. Si el profesor por otro lado, quisiera entrar en un juego inactivo, vería únicamente la clasificación de los alumnos y si en el último caso quisiera ver un juego preparado, se mostrará la clasificación con un botón en verde para poder activar el juego (Fig I.32).

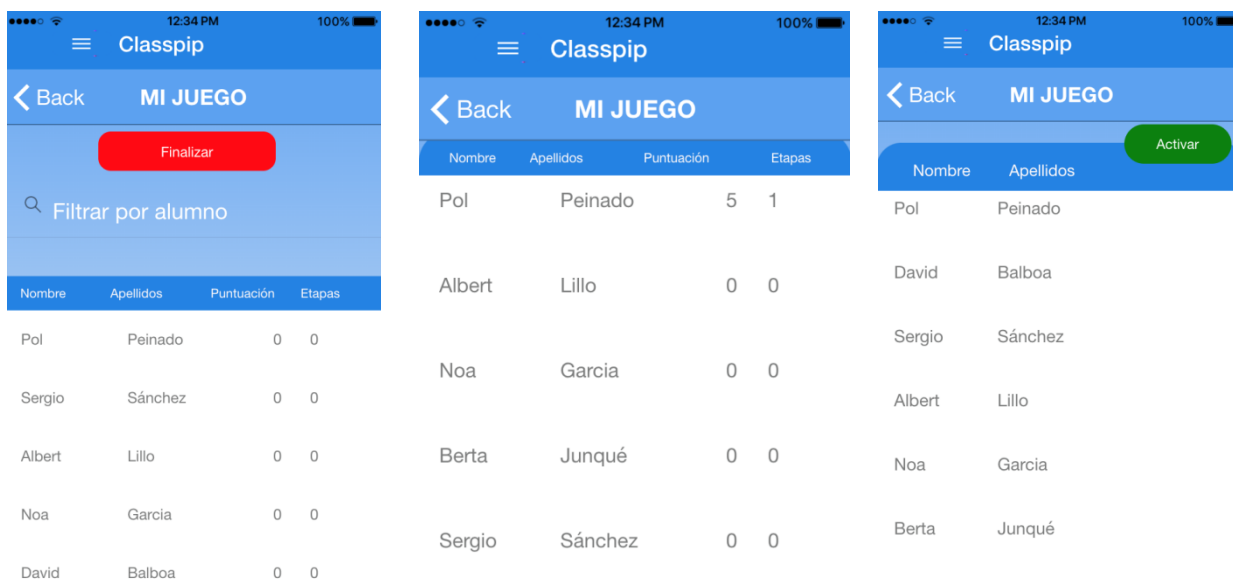


Fig I.32. Pantalla un juego Activo, Inactivo y Preparado respectivamente (Fuente: Aplicación del profesor)

Otra de las funcionalidades que se han incorporado es Mis Escenarios. Para llegar hasta ella, en la pantalla inicial se deberá pulsar sobre las tres barras horizontales en el margen superior izquierdo, entonces una lista de opciones aparecerá y entre ellas encontraremos Mis Escenarios. Incorporo unas imágenes a continuación para simplificar la explicación (Fig I.33).



Fig I.33. Pantalla de Inicio y Opciones respectivamente (Fuente: Aplicación del profesor)

Una vez en Mis Escenarios podremos ver todos los escenarios disponibles y cuando el profesor entre en cualquiera de ellos, se mostrarán los puntos geolocalizables asociados a dicho escenario, mostrándose su nombre y sus coordenadas de latitud y longitud. Además, si quiere cambiar las coordenadas de un punto en concreto, podrá hacerlo haciendo clic en el punto que quiere modificar y seguidamente en el botón Capturar Coordenadas (Fig I.34).

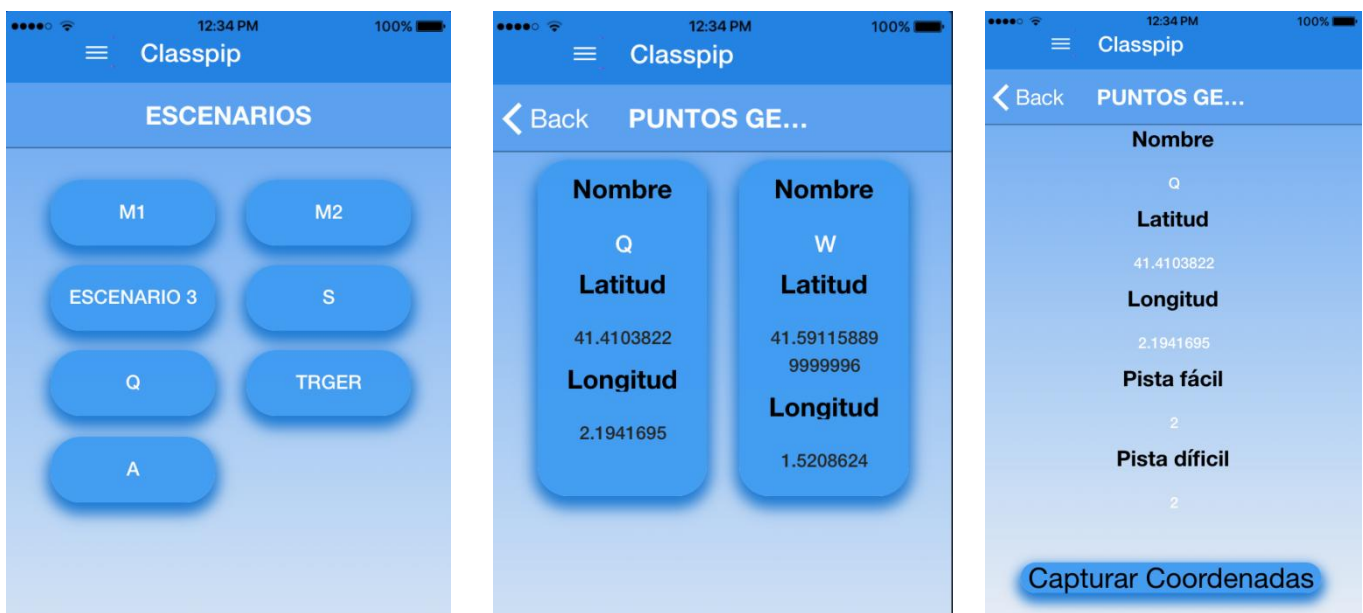


Fig I.34. Pantalla de Escenarios, un Escenario concreto y un punto concreto respectivamente (Fuente: Aplicación del profesor)