TRABAJO FINAL DE GRADO

Título: Diseño de un juego de Escape room y mejoras en herramienta para gamificación

Titulación: Grado en Ingeniería Telemática

Autor: Mario Sánchez Boto

Director: Miguel Valero García

Fecha: 21 de julio del 2021

Título: Diseño de un juego de Escape room y mejoras en herramienta para gamificación

Autor: Mario Sánchez Boto

Director: Miguel Valero García

Fecha: 21 de julio del 2021

Resumen

Este proyecto se ha centrado en la mejora y ampliación de Classpip, una herramienta de gamificación orientada a la docencia. Classpip es el resultado del trabajo conjunto de varios alumnos de la EETAC, que, como nosotros, decidieron ampliar y mejorar esta herramienta.

La gamificación es una técnica de aprendizaje que transfiere la mecánica de los juegos al mundo de la educación con el objetivo de lograr mejores resultados, absorber mejor algunos conocimientos, mejorar algunas habilidades o recompensar acciones específicas mediante diferentes actividades y contenidos que cada usuario personaliza en función de las necesidades de cada estudiante, con la finalidad de atraer la atención de estos para lograr esos objetivos.

En este proyecto hemos participado 4 personas: Jordi Salvador, Ivan Requena, Carlos Vázquez y yo, Mario Sánchez. El objetivo de este trabajo no es otro que el de contribuir en la mejora de Classpip, y para ello hemos introducido nuevas funcionalidades y contribuido con pequeñas mejoras, que hacen que el uso de Classpip sea mucho más cómodo y fácil de entender. Como grupo, hemos trabajado siguiendo la metodología scrum, que explicaremos con detalle en capítulos posteriores.

Mi contribución a Classpip ha sido la siguiente: desarrollar un juego de escape room, para ofrecer otro tipo de herramienta que, actualmente, es un estilo de juego que está teniendo un gran crecimiento, y como he mencionado anteriormente, introducir diferentes mejoras tanto en el código del proyecto como en la parte visual de las herramientas.

En este documento veremos detalladamente cuál ha sido el proceso para llegar a los objetivos que nos propusimos, cuales han sido nuestras dificultades y sobre todo la explicación y planificación para que estos objetivos pudieran seguir adelante de una manera correcta.

Title: Escape room game design and gamification tool enhancements

Author: Mario Sánchez Boto

Director: Miguel Valero García

Date: 21 of July 2021

Overview

This project has focused on the improvement and expansion of Classpip, a gamification tool oriented to teaching. Classpip is the result of the joint work of several EETAC students, who, like us, decided to expand and improve this tool.

Gamification is a learning technique that transfers the mechanics of games to the world of education in order to achieve better results, better absorb some knowledge, improve some skills or reward specific actions through different activities and content that each user customizes according to the needs of each student, in order to attract the attention of these to achieve these objectives.

In this project we have participated 4 people: Jordi Salvador, Ivan Requena, Carlos Vázquez and me, Mario Sánchez. The objective of this work is none other than to contribute to the improvement of Classpip, and for this we have introduced new features and contributed with small improvements, which make the use of Classpip much more comfortable and easier to understand. As a group, we have worked following the scrum methodology, which we will explain in detail in later chapters.

My contribution to Classpip has been the following one: the development of a digital escape room game, to offer another type of tool that, at the moment, is a style of game that is having a great growth, and as I mentioned before, to introduce different improvements in the code of the project as well as in the visual part of the tools.

In this document we will see in detail what has been the process to reach the objectives we set ourselves, which have been our difficulties and also, all the explanation and planning that we have had to do to go ahead in the correct way to finish the project.

ÍNDICE

Capítulo 1: Introducción

Es una evidencia que en pleno 2021 el uso de la tecnología se ha adueñado de nuestro día a día. No hay prácticamente ningún aspecto de nuestras vidas que no se vea afectado por las nuevas tecnologías, y si había gente que aún no había aceptado este hecho, la aparición de una pandemia mundial en marzo de 2020, que ha obligado a todo el mundo a quedarse, como mínimo, unos meses sin salir de casa, lo ha hecho ya irrefutable. A contrarreloj y sin mucho margen de maniobra, todos tuvimos que adaptarnos a esta nueva situación, la cual no habíamos vivido antes, obligando a empresas, escuelas, tiendas y todo tipo de proveedores de bienes y servicios a reinventarse para poder continuar ejercitando sus funciones, de la forma más adecuada y más parecida a la normalidad que hasta ese momento conocíamos.

Uno de los sectores más afectados y que más ha tenido que cambiar su forma de hacer es el mundo de la educación y la enseñanza, el cual, si ya se encontraba en un proceso de cambio para evolucionar y adaptarse a estos nuevos tiempos, la situación que vivimos desde el último año y medio ha forzado la máquina para que este cambio se lleve a cabo urgentemente. El uso de herramientas tecnológicas en las aulas es cada vez más frecuente, lo que favorece la creación de nuevos entornos educativos en los cuales se puede apreciar una mejora en la interacción profesor-alumno y una simplificación de las tareas diarias de los docentes, pero las tecnologías también producen un incremento en las dificultades de los alumnos a la hora de prestar atención y retener conocimientos. Es por eso que se ha iniciado la introducción de la gamificación[1] en las aulas mediante el uso de juegos que ayudan a despertar el interés y la concentración de los alumnos, con el fin de incrementar su rendimiento académico. Por eso, consideramos que la implementación de herramientas de gamificación cómo Classpip acabará siendo clave para contribuir a esta evolución en la forma de enseñar y aprender.

Con el objetivo de contribuir en esta plataforma para aportar nuestro granito de arena en una herramienta que consideramos indispensable para mejorar la enseñanza, mis compañeros Iván Requena, Jordi Salvador, Carlos Vázquez y yo, Mario Sánchez, decidimos realizar este trabajo en el que hemos contribuido en la implementación de mejoras de versiones, estilos, formatos y mecanismos de autentificación en los módulos y aplicaciones ya existentes, además de introducir traducciones, desarrollar un nuevo juego de tipo “Escape Room”, una aplicación “lite” para jugar a juegos rápidos llamada “Classpip Express” y una web con un servidor independiente que sirva como herramienta de expansión y marketing del proyecto. Hay que tener en cuenta que el proyecto lleva ya 6 años de desarrollo, por lo que se encuentra en una fase muy avanzada. Por eso, además de todas las implementaciones, también nos marcamos el objetivo de definir una metodología de trabajo para futuros equipos, y la creación de un plan de pruebas para comprobar el funcionamiento de todas las aplicaciones.

Classpip es una herramienta digital de apoyo para el profesor que le permite utilizar diferentes metodologías basadas en la gamificación con el objetivo de

atraer de forma sencilla la atención de los alumnos, luchando así contra el problema de motivación y de falta de atención mencionados anteriormente. Gracias a Classpip, se ofrecen diferentes herramientas personalizables al profesor que permiten organizar, evaluar, y recompensar a sus alumnos.

El concepto de gamificación se basa en la aplicación de las mecánicas, dinámicas y estéticas de los videojuegos en el ámbito académico-profesional, con el objetivo de mejorar los resultados de las diferentes prácticas llevadas a cabo, además de buscar captar la atención del alumno de una manera mucho más entretenida, ayudando así a comprender de forma mucho más eficiente todos los conocimientos impartidos y generando una motivación extra para seguir aprendiendo.

Como ya he mencionado, este proyecto se ha realizado de forma conjunta, trabajando con un equipo de 4 personas, de manera que todo el trabajo realizado por el equipo se verá repartido en 4 memorias distintas, cada una explicando una parte de todo el trabajo realizado, aunque hay capítulos de estas que inevitablemente van a ser comunes, y, por lo tanto, iguales en las 4 memorias.

En esta memoria, en concreto, se explicará todo el proceso para el desarrollo del escape room, la implementación de un estilo de código definido y la introducción de las temáticas en el dashboard. El documento está dividido en los siguientes capítulos:

En el *Capítulo 2: Classpip* se hace una explicación detallada de que es Classpip, desde su origen y su desarrollo hasta su arquitectura y las diferentes funcionalidades que ofrece al usuario. Además, en el apartado final, se explica qué cambios ha sufrido la herramienta una vez finalizado este trabajo. **Este capítulo es común en las cuatro memorias.**

Una vez explicado qué es Classpip, en el *Capítulo 3: Objetivos y Plan de Trabajo* se explica cuáles han sido los grandes objetivos de este trabajo, desglosándolos en “mini-objetivos” que nos han servido para ir completando de forma ordenada y completa cada uno de los objetivos finales y se define el plan de trabajo que seguirá el grupo durante la elaboración de este para que sea lo más eficaz y sencilla posible, así como la asignación de tareas o la comunicación entre los miembros, entre otros. (**La figura 2 es común en todas las memorias**)

En el *Capítulo 4: Metodología Scrum* se describe la metodología usada para llevar a cabo el proyecto, y cómo la hemos utilizado para obtener nuestro máximo rendimiento. **Este capítulo es común en las cuatro memorias.**

En el *Capítulo 5: Tareas Previas* se explica cuáles fueron nuestros primeros pasos en el proyecto, en qué consistió la primera toma de contacto con el código y cómo realizamos una serie de tareas de aprendizaje con el objetivo de familiarizarnos con todo el entorno.

Con los conocimientos básicos obtenidos y asimilados, en *Capítulo 6: Diseño del dashboard por temática,* se describe el proceso que se ha seguido para

implementar el cambio en el estilo de la herramienta, con el objetivo de ofrecer diferentes posibilidades y adaptar el dashboard a las preferencias del profesor.

En el *Capítulo 7: Diseño del estándar,* se explica el proceso en el desarrollo del estándar. Se explica desde los motivos por los que se ha diseñado, como se ha implementado dicho estándar en el código y qué directrices se han de cumplir con el objetivo de hacer más ágil la programación de las diferentes herramientas.

Como se ha comentado previamente, uno de los grandes objetivos que me planteé fue el desarrollo de un juego de escape room. En el *Capítulo 8: Escape room,* se describe el proceso seguido, el mecanismo de juego tanto por parte del profesor como del alumno, los modelos de datos utilizados y las posibles implementaciones futuras.

Con el objetivo de comprobar todos los cambios realizados en el proyecto y dejar una metodología de trabajo que puedan utilizar futuros participantes, en el *Capítulo 9: Plan de Pruebas,* se explican todos los detalles del plan de pruebas que hemos diseñado. **El apartado 9.1 es común en las cuatro memorias.**

Finalmente, en el *Capítulo 10: Conclusiones* se explican las conclusiones del proyecto y se plantean unas posibles propuestas de mejora de los desarrollos llevados a cabo.

Capítulo 2: Classpip

Como ya se ha mencionado anteriormente, Classpip es una herramienta de gamificación orientada a las aulas, desarrollada por profesores y estudiantes de la *Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels (EETAC)* de la *Universitat Politècnica de Catalunya*. Classpip pretende ayudar a los distintos docentes, tanto de universidades como de escuelas, a realizar de una manera más didáctica su trabajo con el objetivo de captar la atención y mejorar las cualificaciones de los alumnos aplicando las metodologías propuestas por la gamificación. A diferencia de otras herramientas, esta es mucho más versátil ya que el profesor puede configurar las funciones usadas según el objetivo de las distintas clases, pudiendo escoger entre una gran variedad de recursos.

La idea de Classpip se originó en 2016, en el Departamento de Arquitectura de Computadoras, y ha ido evolucionando gracias a las contribuciones de los más de treinta alumnos que, igual que nosotros, han basado su trabajo de final de grado (TFG) en contribuir en el desarrollo y la mejora de esta herramienta.

Gracias a eso, actualmente Classpip se encuentra en una situación muy avanzada y cuenta con diferentes aplicaciones, su propio servidor y su propia API, además de contar con varias funcionalidades que se han ido implementando y una gran variedad de juegos que ofrecen al docente muchas más posibilidades para personalizar sus lecciones.

A continuación, se describirán con más detalle la arquitectura de Classpip, la tecnología usada, las diferentes funcionalidades que ofrece y finalmente los cambios que han sido aplicados a raíz de este proyecto.

2.1 Arquitectura del proyecto

Classpip se dividía en cinco aplicaciones distintas, las cuales juntas conforman el ecosistema de la herramienta. Cada uno tiene sus características que se describen a continuación:

Server: Necesario para gestionar notificaciones a tiempo real entre las distintas aplicaciones. Esto es posible hacerlo gracias al uso de Sockets, método de comunicación entre cliente-servidor. El servidor forma parte del backend y está desarrollado con ExpressJS.

Servicios API-REST: Aplicación que también forma parte del backend, desarrollada con el framework de LoopBack (basado en ExpressJS) la cual aloja la base de datos del sistema y todas las rutas o *endpoints* mediante las cuales se realizan operaciones CRUD (Create, Read, Update, Delete) en los registros. Estos registros se almacenan en un fichero JSON, aunque existe la previsión a corto plazo de hacer la migración de estos datos a MongoDB.

Dashboard: La aplicación Dashboard es la herramienta de gestión del profesor, la cual le permite realizar todo tipo de acciones para preparar sus sesiones en las que decida usar Classpip. A través de la aplicación web es donde el profesor prepara todo tipo de juegos, inscribe a sus alumnos y gestiona sus grupos de trabajo. Esta aplicación forma parte del frontend de la plataforma y ha sido desarrollada con Angular y TypeScript.

Aplicación para móvil orientada al docente: Además de la aplicación web para el navegador, el profesor tiene acceso a una aplicación móvil la cual le permite realizar acciones rápidas sin tener que acceder al Dashboard. El uso de la aplicación móvil facilita acciones como pasar lista, la asignación de puntos o la gestión de juegos que no requieren de mucha complejidad. Esta aplicación forma parte del frontend de la plataforma y ha sido desarrollada con Ionic y TypeScript.

Aplicación para móvil orientada al alumnado: Es la aplicación donde el alumno juega a los juegos en los que el profesor le ha inscrito. Además, permite a los alumnos editar los datos de su perfil y tener un registro de todos los juegos en los que ha participado, estén o no activos. Esta aplicación forma parte del frontend de la plataforma y ha sido desarrollada con Ionic y TypeScript.

Para el desarrollo de una aplicación como Classpip es muy importante escoger las tecnologías adecuadas que permitan llevar a cabo, de una manera correcta, los distintos módulos y mostrar e interconectar la información de manera adecuada.

Para llevar a cabo las distintas funcionalidades con las que ya cogimos el proyecto y para realizar las nuevas hemos utilizados estas diferentes tecnologías:

Angular: Framework *opensource* desarrollado por Google. Hemos escogido esta tecnología porque ya estábamos familiarizados con ella y permite crear aplicaciones web de una manera sencilla y organizada.

Ionic: Framework *opensource* basado en Angular y Apache Cordova mediante la cual se ha desarrollado la interfaz gráfica tanto del móvil del estudiante como el del profesor

Strongloop LoopBack 3: Framework de NodeJS basado en ExpressJS que aloja la base de datos y los endpoints de la API REST.

Bootstrap: Librería usada para el diseño de las aplicaciones que forman parte del frontend. Esta tecnología permite implementar diseños responsive en las aplicaciones para que se adapten a los tamaños de los diferentes dispositivos.

GitHub [2] **:** Herramienta de software de control de versiones en la que se alojan los códigos fuente de todos los proyectos en repositorios. Esta herramienta permite trabajar de forma colaborativa, manteniendo un histórico de los cambios y permite añadir peticiones de modificación del código de otra persona.

Visual Studio Code: Editor de código fuente desarrollado por Microsoft. Hemos utilizado esta aplicación para la edición del código ya que dispone de muchas herramientas y extensiones que mejoran la productividad.

Una vez mencionadas todas las aplicaciones y tecnologías, podemos ver en la *Figura 2.1* cómo están relacionadas y mediante qué protocolos se comunican.

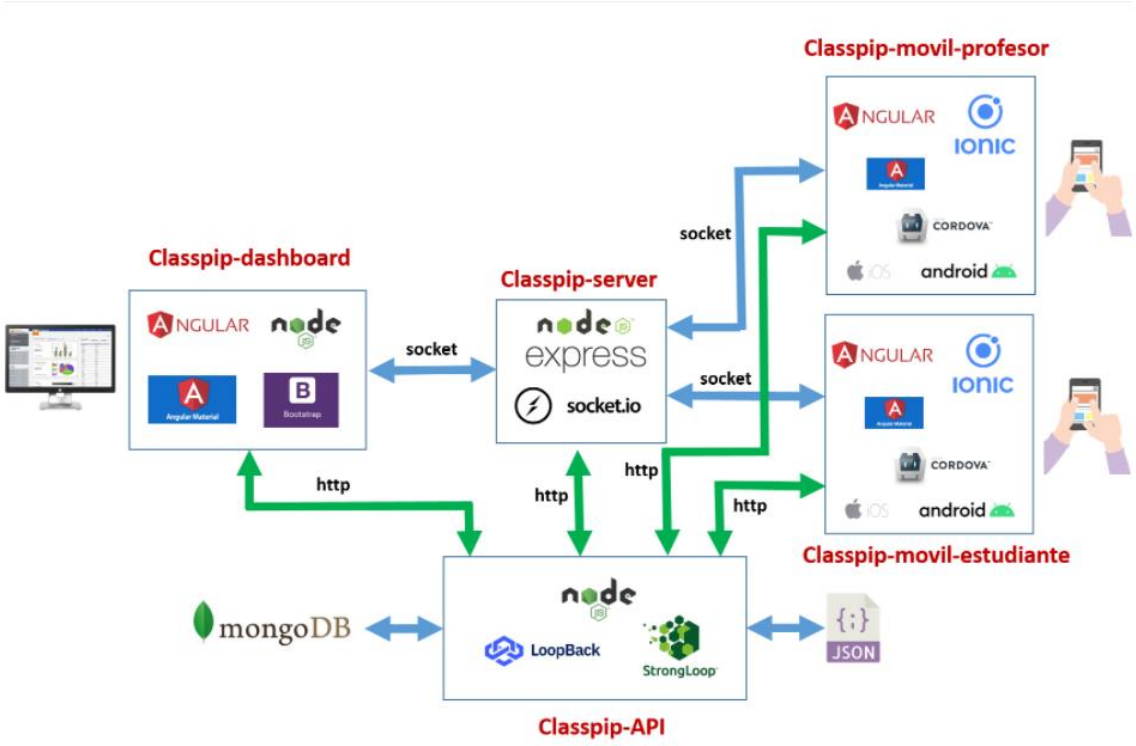


Figura 2.1 ***Arquitectura ecosistema Classpip 1***

2.2 Funcionalidades

A continuación, se describen las funcionalidades que Classpip ofrece a sus usuarios. Tal y como hemos comentado anteriormente, se han desarrollado distintos juegos para poder captar la atención y así, obtener un mejor rendimiento de los alumnos. Estos juegos son los siguientes:

Juego de puntos: El profesor podrá asignar puntos a los alumnos en función de los méritos obtenidos a través de pruebas que se hayan propuesto en clase. Estos puntos se verán reflejados en un ranking y cada vez que se vayan obteniendo más puntos los alumnos podrán acceder a distintos privilegios.

Juego de Cuestionario: En el juego de cuestionario el profesor prepara una batería de preguntas, las cuales pueden ser cualquier temática y de cualquiera de los 4 tipos que hay: Verdadero o Falso, Cuatro Opciones, Respuesta Abierta y/o Emparejamiento. Una vez preparado el cuestionario con todas las preguntas que el profesor haya querido meter, define un tiempo límite para poder responder cada pregunta. Una vez iniciado el juego, el profesor será el encargado de ir lanzando las preguntas, que los alumnos irán recibiendo y deberán contestar, no solo de forma correcta, sino también en el menor tiempo posible, ya que los más rápidos respondiendo correctamente son los que obtendrán más puntos.

Juego de Cuestionario de Satisfacción: El juego de cuestionario tiene una variante, en la cual el profesor, en vez de preparar el cuestionario mediante las preguntas disponibles en la batería de preguntas, prepara un cuestionario de satisfacción/valoración, con preguntas en las cuáles se tiene que puntuar una cualidad o un concepto del 1 al 5 (1 siendo la nota más baja y 5 la más alta), o preguntas de respuesta abierta donde lo que se pide es la opinión del alumno con respecto a algún concepto en concreto.

Juego de Avatar: Los alumnos podrán personalizar su avatar a través de privilegios que el profesor le vaya otorgando. No es un juego de puntos o objetivos si no es una manera de premiar a los alumnos por el esfuerzo y que estos puedan personalizar su avatar de manera única.

Juego de Colección: En este tipo de juego, el alumno colecciona diferentes cromos de una colección concreta, la cual puede ir variando según la asignatura o el tema que se esté tratando en ese momento (filósofos, músicos, comandos informáticos...) y una vez completada la colección, esta te muestra algún tipo de información relevante de cara a la asignatura (la obra de algún filósofo, una partitura, el código de un programa...). El profesor podrá ir asignando paquetes de cromos a sus alumnos según sus méritos en clase.

Juego de Competición: Hay 3 tipos de variante para este juego:

> **Liga:** Se realizarán enfrentamientos entre dos alumnos y el ganador será escogido por el profesor según los criterios que se haya impuesto ese día**,** el ganador de ese día obtendrá puntos para la clasificación y si el profesor lo desea, recompensas para otros juegos.

Torneo: A lo largo de la clase se irán realizando enfrentamientos entre los alumnos y donde los ganadores de estos pasarán a la siguiente ronda. El profesor podrá marcar los criterios que hagan vencedor a un alumno o a otro así como la recompensa para el ganador.

Fórmula 1: Todos los alumnos se enfrentarán entre ellos cada jornada, como en el de la Liga. El ganador podrá obtener recompensas para los otros juegos.

Juego de Geocaching: En este juego el alumno debe ir respondiendo preguntas mientras se mueve siguiendo un recorrido que el profesor ha determinado con distintas ubicaciones. En cada ubicación el alumno recibe una pregunta que debe contestar correctamente para sumar puntos. La aplicación proporciona una pista difícil al alumno de dónde se encuentra la ubicación que debe encontrar, además de mandar una alerta cuando la distancia es de 25 metros y otra cuando esta es de 5 metros. Al final del juego, el alumno con más puntos acumulados se convierte en el ganador.

2.3 Nuevas aportaciones

El objetivo principal cuando escogimos el proyecto era implementar mejoras en las aplicaciones existentes y el desarrollo de algunos módulos nuevos. Por lo tanto, podemos diferenciar las implementaciones desarrolladas en nuevas funcionalidades, nuevas configuraciones y nuevas aplicaciones.

2.3.1 Nuevas funcionalidades

Juego Escape Room: Se ha diseñado un juego didáctico, en el que se permite al profesor, comprobar si se han absorbido los conocimientos impartidos, pero de una manera mucho más interactiva de lo habitual.
El juego sigue las reglas estándar de los escape rooms reales, es decir, tienen el objetivo de escapar de los escenarios superando diferentes pruebas.

Traducción Dashboard: El objetivo de esta tarea era poder obtener la página del Dashboard en diferentes idiomas, se ha utilizado una extensión de Angular llamada i18n con la cual nos da la opción de tener nuestra página en el idioma que nosotros queramos, el cual escogimos el inglés.

2.3.2 Nuevas configuraciones

Mejoras en mecanismos de autenticación: Cuando empezamos a desarrollar las funcionalidades y a investigar acerca de los modelos de datos, nos dimos cuenta de que las contraseñas de los usuarios se enviaban en claro en todas las comunicaciones HTTP, no se guardaban cifradas en la base de datos, no se comprobaba que el usuario tuviese una sesión activa y no se daba la opción a mantener la sesión iniciada. Al detectar esto, saltaron todas nuestras alarmas, ya que considerábamos que esta configuración era una gran vulnerabilidad en la

seguridad de Classpip, por lo que decidimos llevar a cabo esta tarea implementando un mecanismo de creación de *tokens* para el control de acceso.

Actualización versión Dashboard: Para poder desarrollar con éxito la internacionalización del Dashboard nos vimos en la obligación de actualizar la versión de Angular, ya que no era compatible la versión actual con el i18n de angular.

Reglas y estándar de formato de código: A la hora de realizar las diferentes tareas, que se explicarán a lo largo del proyecto, nos dimos cuenta que el código no estaba unificado, lo que supuso un problema estructural para realizar los diferentes objetivos establecidos en el inicio del proyecto.

2.3.3 Nuevas aplicaciones

Classpip Express: Cuando iniciamos el proyecto, para acceder a los juegos rápidos, actividad dónde el alumno, sin necesidad de estar registrado en Classpip, puede participar a las preguntas que le profesor ha creado poniendo únicamente un numero pin generado aleatoriamente, se requería usar la aplicación del móvil del estudiante. Con el fin de disponer de una aplicación más ligera sobre la cual añadir reglas de estilo de código, se separaron del móvil del alumno las funcionalidades de los juegos rápidos, creando así Classpip Express. Esta aplicación utiliza las mismas tecnologías que el móvil del alumno.

Classpip Web: A pesar de que todavía no hay un plan de negocio definido para comercializar Classpip, la idea es que para cada escuela/universidad que quiera utilizar las herramientas de la plataforma, se cree una organización independiente con su propia API-REST. Esto quiere decir que, entre los diferentes clientes que tenga Classpip, los registros no serán compartidos, por lo que no se podrán compartir los diferentes recursos y juegos creados en las distintas organizaciones. Con el objetivo de que todos los docentes que usen la herramienta puedan compartir recursos y experiencias, se ha desarrollado una web para Classpip. Pensada para dar a conocer el proyecto y captar nuevos clientes y desarrolladores, esta aplicación contendrá información acerca de Classpip, documentación de la aplicación, datos de contacto de los responsables y diferentes componentes sobre los que se puede crear una comunidad de usuarios. Para desarrollar la página web, se ha utilizado una plantilla para Angular.

Servicios API-REST para la web: Como hemos comentado en el apartado anterior, con tal de que todos los docentes puedan compartir recursos, se ha creado un servidor REST con el framework de LoopBack basado en el del Dashboard, mucho más ligero y con menos modelos. Este servidor es único y sirve solo para la aplicación web.

Podemos observar en la *Figura 2.2* cómo sería la nueva estructura de Classpip con la incorporación de estas nuevas aplicaciones.

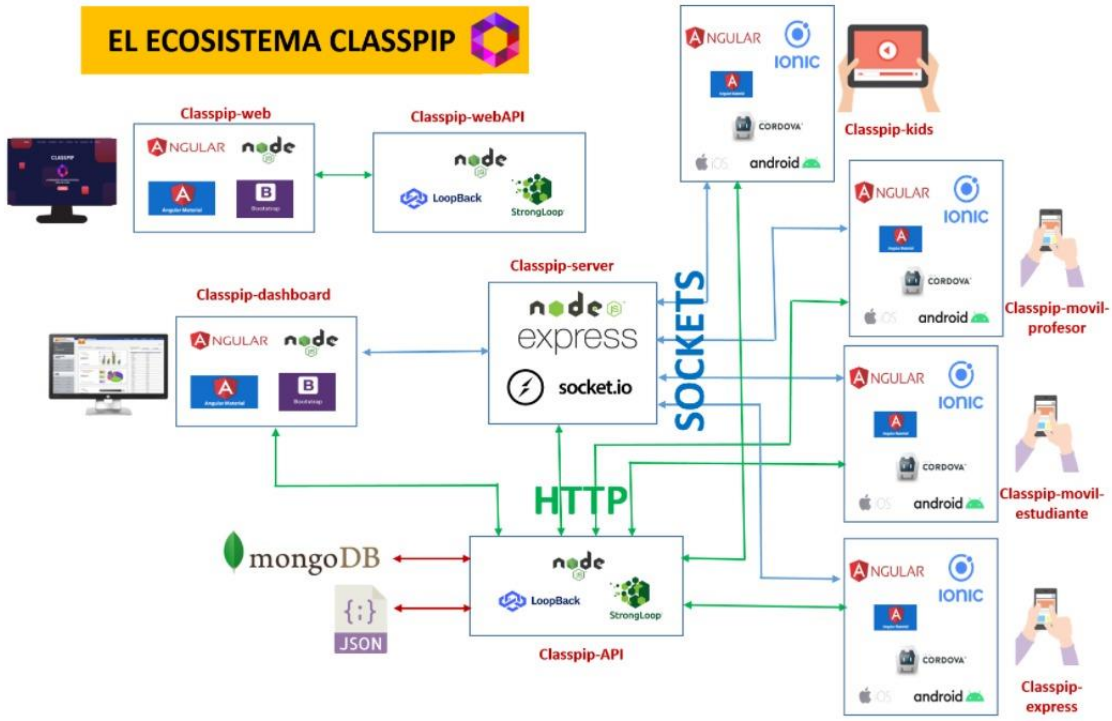


Figura 2.2 Arquitectura ecosistema Classpip 2

Capítulo 3: Objetivos y plan de trabajo

En este capítulo se presentan los objetivos que el equipo, siguiendo las propuestas de los tutores, se planteó al iniciar el proyecto y cuál ha sido el plan de trabajo seguido.

Se distinguen cuatro grandes objetivos:

- El primero es definir una metodología para trabajar en equipo en un proyecto de este tipo.
- En segundo lugar, uno de los objetivos principales del proyecto ha sido el desarrollo de un juego de escape room con el fin de acercar un poco más la gamificación a la docencia.
- En tercer lugar, se pueden agrupar las diferentes tareas que se han ido realizado paralelamente al desarrollo del escape, que son tales como la incorporación de la temática estaciones para el diseño del dashboard y la regulación de un estándar para el código.
- Finalmente, la definición de un plan de pruebas que ayude a comprobar que todas las implementaciones y los cambios en el estilo del código que se han aplicado y se aplicarán en futuros desarrollos, funcionan correctamente.

Estos objetivos se desglosaron en “subobjetivos” que han ayudado al equipo a tener una idea mucho más clara de los pasos a seguir y los requisitos a cumplir, estos se definen a continuación.

Respecto a la definición de una metodología de trabajo:

- Generar una metodología específica para nuestro equipo, teniendo en cuenta las condiciones de cada miembro y sus posibilidades de aportación al proyecto.
- Aplicar y valorar dicha metodología con el fin de estandarizarla para que futuros estudiantes que se encuentren en una situación similar a la de nuestro equipo puedan aplicarla y agilizar su trabajo.

Respecto al desarrollo del juego de escape room:

- Desarrollar una herramienta que ofrezca al profesor la capacidad de examinar y comprobar los conocimientos impartidos en la docencia de una manera mucho más interactiva y atractiva de cara al estudiante.
- El profesor ha de tener la posibilidad de diseñar gran parte del juego. Para ello, se deberá crear un formato que permita al profesor adaptar el juego a sus materias.
- El alumno tendrá una experiencia lo más similar posible a los juegos reales de escape room dentro de las posibilidades que ofrecen las herramientas.

Respecto a la implementación de una temática en el dashboard, el objetivo principal es ofrecer la posibilidad al profesor de cambiar de estilo según la época

del año. Uno de los objetivos que vienen intrínsecos con esta tarea, es desarrollar un formato que guíe en proyectos futuros a estudiantes que desarrollen diferentes temáticas a la de estaciones.

Respecto a la creación de un estándar en el código del proyecto:

- Uno de los objetivos principales es que ha de facilitar la programación de las diferentes herramientas, dado que es un proyecto en el que participan una gran cantidad de personas.
- El estándar ha de uniformizar el diseño de los modelos de Classpip.

Por último, sabíamos que, una vez terminados los desarrollos, deberíamos probar todas las funcionalidades que hayamos implementado y comprobar que las que ya estaban hechas funcionaban correctamente después de todas las modificaciones. Teniendo en cuenta que otro gran objetivo era definir una metodología de trabajo que sirviese para futuros equipos, decidimos elaborar un documento de excel donde registrar todas las pruebas a realizar, explicando paso por paso qué procedimiento debe seguirse para la realización de la prueba y cuál es el resultado esperado. De esta manera, los próximos desarrolladores que se incorporen al proyecto, tienen en un documento centralizadas todas las funcionalidades y cuál debe ser su comportamiento.

Una vez definidos de forma clara los objetivos, decidimos, junto al asesoramiento de los tutores, plantear un plan de trabajo estructurado para llevar a cabo todo el desarrollo de este trabajo.

Previamente a realizar ningún tipo de desarrollo en una aplicación la cual, hasta el momento, era totalmente desconocida para nosotros, decidimos realizar una serie de tareas de aprendizaje, con el objetivo de empezar a familiarizarnos con la estructura y el código de la plataforma. Para ello realizamos un ejercicio de absorción de conocimientos, leyendo la documentación relativa a Classpip que aparece en Github y visualizando los videotutoriales que nuestro tutor, Miguel Valero, tiene colgados en Youtube. Una vez adquiridos los conocimientos básicos, el siguiente paso consistió en entrar en contacto con el código, realizando pequeños cambios de manera que se vieran reflejados en forma de mejoras en diferentes aspectos de la plataforma.

Con los conocimientos básicos obtenidos, durante los primeros sprints, las dos tareas principales fueron buscar información del mundo *escape*, tanto en la vertiente de los juegos reales, como en la de los juegos virtuales y implementar la temática en el dashboard, ya que era una tarea que no es de una complejidad extrema y tenía el objetivo de familiarizarme con el código.

El desarrollo del *escape room* ha estado presente desde el inicio del proyecto hasta el final del mismo. Primero, se empezó diseñando los modelos e incluyendo los primeros pasos del juego tanto en el Dashboard, que es la herramienta que utilizará el profesor para configurar el juego, como en el móvil del estudiante, que es donde el alumno tendrá acceso al juego.

El siguiente paso una vez finalizada la implementación del estilo en el dashboard y desarrolladas las bases del juego en las diferentes herramientas, ha sido crear el diseño del estándar para uniformizar el código del proyecto. Esta tarea se ha desarrollado en este momento del proyecto por los siguientes motivos: no es una tarea de baja complejidad para hacerla inicialmente, se ha de tener conocimientos sobre el proyecto y para facilitar la programación relacionada con el juego de escape room, era interesante haber aplicado previamente los cambios en el código. Mencionar que no ha sido tarea fácil y que se ha implementado en las herramientas de Dashboard, Móvil de Estudiante, Classpip express y Web.

En la etapa final del proyecto, se ha continuado con el desarrollo del escape room y por otro lado, una vez dado por terminados todos los desarrollos, el siguiente paso era comprobar que todos estos que se habían realizado, funcionaban correctamente. Por lo que, el último paso fue realizar un plan de pruebas y aplicarlo a las diferentes aplicaciones, confirmando así que dichas aplicaciones son funcionales y están listas para entrar en producción.

Con tal de cumplir con el plan de trabajo propuesto y los objetivos planteados, el equipo decidió trabajar mediante la metodología Scrum, la cual está explicada en el *Capítulo 3: Metodología Scrum*, y por lo tanto se trabajó por sprints. A continuación, en la *Tabla 1*, se muestra el proceso que se ha seguido y el tiempo dedicado en cada tarea, de forma individual.

Sprint	Tarea	Descripción	Horas Realizadas			
			M	J	I	C
1 10/02 - 26/02	Diseñar web Classpip	Idea inicial con mockups de la plantilla de la web.		15	15	
	Separar juegos rápidos	Separar los juegos rápidos del móvil del alumno en una nueva aplicación llamada ClasspipExpress.		15	20	
	Crear juego Escape Room	Búsqueda de información de Escape Rooms y pensar el diseño inicial del juego.	15			15
	Introducir selección idioma	Buscar información sobre i18n.				
	Rediseñar estilo dashboard	Pensar temática de personalización y primera versión.	10	5		10
2 26/02 - 19/03	Desarrollo web	Búsqueda de la plantilla y adaptación a nuestra web.		36	42	
	Separación juego rápido	Eliminar juego rápido de móvil estudiante y eliminar dependencias innecesarias de ClasspipExpress.		20	10	
	Estilo dashboard	Cambio de variables estáticas a variables y temática por estaciones.	20			5

	Traducción dashboard	Empezar a realizar la instalación del componente.				20
	Creación juego Escape Room	Crear páginas para la creación de personajes, portada y nuevo proyecto Escape Room .	15			
3 19/03 - 08/04	Tecnologías Escape Room	Probar con angular material cómo sería el movimiento de un objeto.	2			
	Página recursos	Desarrollo página de recursos de la web.		20	22	
	Autentificación y perfil web	Login y registro, modificación del navbar y página perfil.		19	20	
	Avances Escape Room	Escenario principal Escape Room, audio y creación de los modelos en la API.	40			
	Exportación aplicaciones	Buscar información de obtener una aplicación para un dispositivo móvil.				20
	Traducción del dashboard	Seguir con la traducción del dashboard				30
4 08/04 - 22/04	Mostrar recursos	Avatares, colección, imagen perfil, pregunta.		37	40	
	Memoria	Identificar apartados comunes de la memorias y redactar.	5	5	5	5
	Escape Room a móvil-estudiante	Adaptar todo el proyecto al móvil del estudiante, es decir, adaptar clases, pantallas, servicios, peticiones.	40			

	Clases Escape Room	Definir las clases finales que se van a utilizar.	25			
	Actualización de angular	Buscar información de como actualizar angular				40
5 22/04 - 13/05	Capacitor	Buscar información del capacitor y ver si es viable la migración.				
	Información y Recursos	Añadir información y recursos a las diferentes páginas de la web de Classpip.		27	12	
	Adaptar tamaños	Adaptar los diferentes componentes al dispositivo.	5	5		
	API	Configuración de la API para incorporar la autentificación.		22	35	
	Escape Room	Se han añadido nuevas funcionalidades.	35			
	Actualización angular	Solucionar errores con la actualización de angular				30
6 13/05 - 03/06	API web	Crear una nueva API independiente para la web		33	40	
	Experiencias	Crear modelos y servicios en la API, diseño del html y integración de Web Services		72	95	
	Escape Room	Adaptación del dashboard y móvil estudiante a los modelos, crear recurso escenario y recurso objeto en el dashboard	90			
	Instalación de capacitor	Se ha migrado cordova a capacitor				35

	Aplicación	Buscar cómo extraer aplicaciones para Android e iOs				30
	Error actualización	Seguir con el error de la actualización del dashboard				15
7 03/06 - 17/06	Escape Room	Mejorar funciones en los recursos escenario y objeto Escape Room, crear escenarios y añadir objeto pista	50			
	Recursos	Integrar nuevas funcionalidades en la web.		68	75	
	Página Perfil	Cambiar datos profesor, foto de perfil y contraseña y poder ver el perfil de otros usuario		18	10	
	Aplicación en Android	Se ha extraído correctamente la aplicación en Android pero en iOs no disponía del material necesario.				30
8 17/06 - 28/6	Ampliación experiencias	Ampliación de la página de experiencias		33	39	
	Autentificación	Avances en la autentificación		23	32	
	Escape Room	Rediseñar los modelos del juego Escape Room y creación de los nuevos recursos, crear juego en el dashboard y rediseñar la mecánica del juego	115		5	
	Estilo del código	Búsqueda de información sobre la instalación, configuración y reglas que con *EsLin*t y *Prettier*				40
	Plan de Pruebas	Creación de un plan de pruebas en Classpip	10	10	30	10

	Memoria	Empezar con el redactado a fondo de la memoria	15	20	8	60
9 28/06 - 11/07	Autentificación	Arreglos autentificación Classpip		25	15	
	Web	*Landing,* diseñar la web más sencilla. Subida a producción de la web y arreglo en Classpip Express		16	20	
	Escape Room	Implementaciones mejoradas y función de guardado.	90			
	App Iphone	Exportar la aplicación para dispositivos iOS				20
	Traducción dashboard	Seguir con la traducción del dashboard				40
	Memoria	Seguir con la redacción de la memória	60	45	50	80

Tabla 3.1: Proceso del proyecto

Capítulo 4: Metodología Scrum

Cómo Classpip se encuentra en una fase de desarrollo avanzada y la idea es que, a partir de ahora, nuevos equipos de trabajo se sumen a mejorar las aplicaciones de la plataforma. Para facilitar el trabajo a estos futuros desarrolladores, hemos implantado unas bases y criterios de organización a seguir a la hora de realizar nuevos proyectos con el fin de definir una metodología de trabajo. Una de las metodologías de trabajo más comunes en equipos de desarrolladores es la metodología Scrum.

En este capítulo se explica en que se basa esta metodología y cómo la hemos aplicado en nuestro equipo para llevar a cabo este trabajo, detallando las ventajas y desventajas con las que nos hemos encontrado durante todo el proceso que hemos seguido, desde la primera reunión hasta la última. Finalmente se presentan los resultados obtenidos al aplicar esta metodología, y se define una propuesta de metodología para futuros equipos de trabajo que se encuentren en la misma situación que nosotros.

4.1 Definición de la metodología Scrum y la herramienta Trello

En este subapartado se describen las características de la metodología Scrum y la herramienta Trello, usadas en este trabajo para la gestión de las tareas y del equipo.

La metodología Scrum[3] es un proceso de trabajo en equipo dónde, para lograr la máxima y óptima colaboración entre los distintos miembros que forman parte del equipo, se aplican un conjunto de buenas prácticas, con el objetivo de lograr el mejor resultado posible para el proyecto en el cual está implicado el equipo. Este tipo de prácticas tiene su origen en un estudio de cuál era la forma de organizarse y trabajar en equipos altamente productivos.

El funcionamiento de Scrum se basa en la realización de entregas parciales y regulares del producto final. Esto la hace una metodología especialmente idónea para proyectos que se realizan en entornos complejos, donde no suele haber unos objetivos definidos claros desde el primer día. Normalmente, en proyectos de programación donde se aplica Scrum, también se utiliza el proceso de trabajo Agile, un proceso de revaluación constante que permite maximizar el valor ofrecido al cliente, creando así métodos que permite obtener resultados en un corto plazo y en el cual predominan la innovación, la competitividad, la flexibilidad y la productividad.

Los proyectos donde se aplica Scrum se ejecutan en ciclos temporales cortos y de duración normalmente fija. Estos suelen tener una duración de dos semanas, aunque se puede adaptar según los requisitos del proyecto. Al final de cada iteración (***sprint***, de ahora en adelante) tiene que haber un resultado completo que contribuya en incrementar el producto final, de manera que cada vez que el cliente solicite su entrega, este pueda ser entregado de la forma más rápida y eficaz.

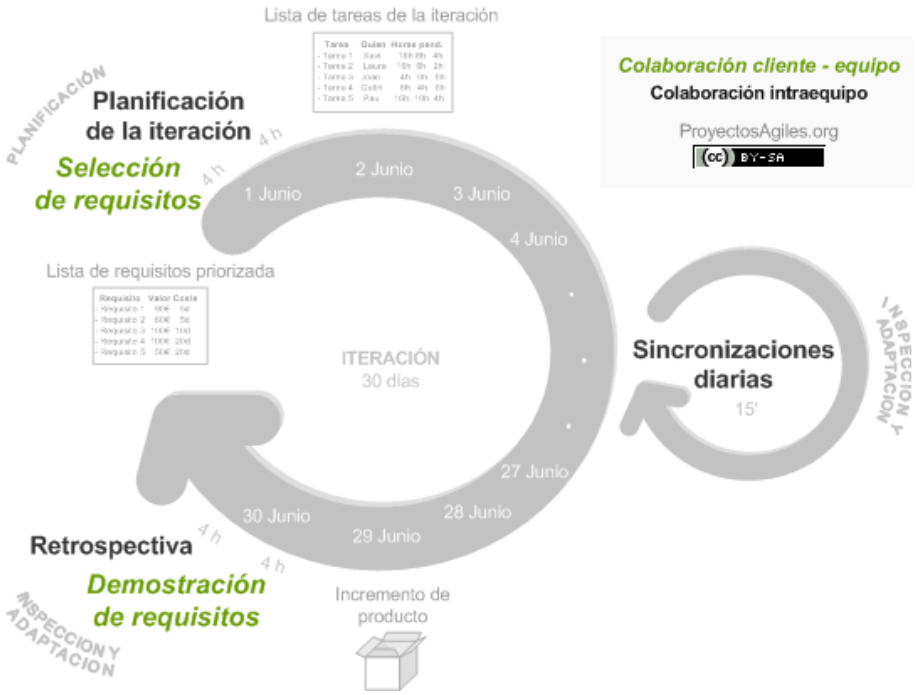


Figura 4.1 ***Ejemplo de ciclo de Sprint***

El proceso se inicia con la elaboración de una lista de objetivos, la cual actúa como plan de proyecto. En esta, el cliente prioriza los distintos objetivos haciendo un balance entre el coste de estos y el valor que aportaran, y estos se reparten en los distintos sprints y entregas. En la *Figura 4.1* se puede apreciar un ejemplo de ciclo de un sprint completo.

Al hablar de Scrum, debemos tener en cuenta una figura imprescindible en proyectos donde se aplica esta metodología: el **Scrum Master**[4]. El "Scrum Master" es un miembro del equipo que se encarga de que la metodología Scrum se aplique de forma correcta a lo largo de cada sprint, de que los distintos miembros del equipo tengan tareas asignadas y resolver los posibles conflictos que se puedan encontrar los miembros del equipo. Ejercen la función de líder del equipo y son la persona de contacto directo entre equipo de trabajo y cliente.

Los pasos que se llevan a cabo a lo largo de cada sprint son los siguientes:

1. Reunión inicial: El primer día de cada sprint se realiza una reunión donde se planifica qué tareas se llevarán a cabo. En la reunión, liderada por el Scrum Master, participan todos los miembros del equipo junto al cliente. En esta reunión, se tratan los siguientes temas:

    1.1. **Selección de requisitos**. En el primer sprint de todos, el que da inicio al proyecto, el cliente presenta la lista de requisitos priorizada que se esperan para el producto final. Una vez definidos los requisitos del cliente, se dividen en pequeñas tareas y subtareas. Entonces el equipo selecciona las tareas más prioritarias que cree que podrá realizar en el

tiempo que dura el sprint, con la finalidad de poder ser entregados si el cliente los pide.

En el resto de sprints, se sigue presentando la lista de requisitos que quedan pendientes y el equipo vuelve a realizar el proceso de creación y selección de tareas mencionado anteriormente.

1.2. **Planificación del sprint**. Una vez terminada la reunión con el cliente, el equipo se reúne de nuevo para elaborar una planificación adecuada de cómo se debe llevar a cabo el sprint para poder cumplir con los requisitos, donde se especifican qué tareas se van a realizar y que miembro la llevará a cabo. Se hace una estimación de esfuerzo de manera conjunta y cada miembro se asigna las tareas que realizará e incluso se forman subgrupos dentro del equipo para que las tareas en las que se trabaje en grupo se realicen de forma más rápida y eficaz.

2. Ejecución del sprint: Un concepto clave de los sprints de Scrum son las **"*daily meetings*"** (o reuniones diarias). Cada día el equipo tiene una reunión de seguimiento donde se analiza el trabajo realizado hasta el momento por todos los miembros, de manera que si hay que realizar algún cambio en la organización del sprint o en las prioridades de los requisitos (debido a algún contratiempo o problema que haya podido surgir durante la realización de alguna de las tareas), se pueda hacer a tiempo de manera que no acabe perjudicando la entrega final del sprint. A lo largo de estas reuniones cortas de seguimiento, cada miembro responde a tres preguntas:

   - ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
   - ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
   - ¿Qué impedimentos tengo o voy a tener que nos impiden conseguir nuestro objetivo?

3. Análisis de la iteración y planificación de la siguiente: El último día de cada sprint se realiza una reunión final donde se tratan los siguientes temas:

   3.1. **Revisión (demostración)**. El primer paso en esta reunión final es presentar al cliente cuál ha sido el trabajo realizado a lo largo del sprint, mostrando el estado del producto que está listo para ser entregado. Dependiendo de los resultados obtenidos y los cambios realizados en el proyecto, el cliente readapta, o no, los requisitos y objetivos del proyecto, volviéndolo a planificar si fuese necesario.

3.2. **Retrospectiva**. Una vez mostrados los resultados y reestructurado el proyecto y sus objetivos (si ha sido necesario), el equipo hace un análisis de cómo ha sido su manera de trabajar, que beneficios han sacado de trabajar de la forma en la que lo han hecho, y también los problemas o obstáculos con los que se han encontrado a lo largo del sprint, de manera que estos se puedan solucionar para los sprints que están por venir.

3.3. **Planificación del siguiente sprint**. Finalmente, una vez presentados los resultados del proyecto y analizado la forma de trabajar del equipo, se aprovecha la parte final de la reunión para planificar, de la forma que se ha explicado anteriormente, el siguiente sprint.

Como se puede ver en la definición de los pasos a seguir en cada sprint, el elemento clave para aplicar la metodología Scrum con éxito son las tareas. Con tal de tener todas las tareas organizadas en un mismo sitio y que sean accesibles para todos los miembros del equipo, se utilizan herramientas que proporcionan tableros en los que se recogen todas las tareas y subtareas. Para nuestro proyecto, la herramienta utilizada para este fin ha sido Trello.

Trello[5] es una aplicación diseñada para la organización de las tareas en equipos de trabajo. Esta herramienta nos proporciona un tablero en el que se pueden añadir listas y tarjetas, permitiéndonos organizar nuestras tareas del proyecto de una manera fácil, rápida y cómoda.

4.2 Nuestra experiencia usando Scrum

Tal y como hemos comentado, la metodología de Scrum facilita las tareas de trabajo y permite solucionar rápido todos los retos e inconvenientes que puedan surgir durante los desarrollos, siendo una de las metodologías más usadas. Viendo con perspectiva el trabajo realizado, coincidimos todo el equipo en que esta metodología, definitivamente, ha sido la adecuada para realizar el proyecto. Teniendo en cuenta que la plataforma de Classpip ya se encuentra en fase avanzada y que debemos adaptarnos a los criterios de los tutores del proyecto para que este siga el *roadmap* establecido, el hecho de trabajar con entregas periódicas nos ha permitido rectificar e incorporar los desarrollos que íbamos realizando.

Al inicio del proyecto, hicimos la reunión inicial. En esta reunión, se definieron los criterios de Scrum que íbamos a seguir, ajustándose a las necesidades tanto de los profesores como nuestras. Se estableció que las entregas de los sprints serían cada dos semanas (permitiendo cierta flexibilidad) y que cada dos sprints, rotaríamos el papel del Scrum Master. Además, se nos proporcionó un tablero de Trello, el cual únicamente tenía una lista llamada “Backlog”, donde estaban recopiladas las posibles tareas a realizar de una manera general en tarjetas (lo que serían los requisitos de cliente). Entre todos escogimos qué tareas íbamos a abordar para nuestros proyectos y, a partir de aquí, fuimos cogiendo estas tarjetas y las íbamos organizando en pequeñas tareas y subtareas para cada Sprint.

Por último, se acordó la realización de un documento para cada sprint (anexo 1) en los que se recogería la siguiente información:

- Tareas realizadas durante el sprint
- ¿Qué cosas han salido bien respecto al grupo?
- ¿Qué cosas han salido mal? En caso de haber alguna, ¿qué vamos a hacer para mejorarlo en el siguiente sprint?

De esta manera, hemos conseguido tener un registro de todas las tareas realizadas durante el sprint, de los conflictos que han salido y se podía mostrar en la reunión del sprint a modo de resumen de los puntos a tratar.

En lo que respecta al Trello, esta herramienta nos ha ayudado muchísimo, ya que nos permitía definir que es lo que íbamos a entregar para cada sprint, hacer un seguimiento de en qué estado se encontraban las diferentes tareas y ver las tareas pendientes. Ahora bien, si esta herramienta no se utiliza de la manera adecuada, puede no tener ningún efecto en la mejora de la productividad del equipo. Por suerte, nosotros ya teníamos experiencia con herramientas y metodologías similares, por lo que definimos un flujo para las tarjetas y creamos una serie de protocolos y herramientas que nos ayudaron a trabajar con Trello de una manera efectiva. Este flujo queda definido en el siguiente apartado como propuesta de metodología a aplicar por los próximos equipos de trabajo.

4.3 Propuesta de metodología

Tras haber trabajado en este proyecto usando la metodología explicada anteriormente, hemos decidido adaptar la metodología definida por Scrum, realizando una propuesta que se adapta a las necesidades de los estudiantes que participen en el proyecto de Classpip en un futuro.

4.3.1 Scrum

En lo que a Scrum respecta, nuestra propuesta se estructura de la siguiente forma:

1. Se realiza una reunión inicial, donde los tutores presentan el proyecto, ayudan en la instalación y conocimiento de la herramienta, proponen tareas iniciales y marcan el plan de trabajo a seguir. En esta reunión inicial se define el primer Scrum Master, el cual lo será durante los próximos dos sprints, y se decide la fecha de entrega del primer sprint (de dos semanas normalmente).

2. Una vez definido el Scrum Master y las tareas a realizar por el equipo, estos organizan una reunión para distribuir las tareas y empiezan a trabajar para finalizarlas en el período marcado. A mitad del período, se reúnen para comprobar cómo está avanzando el sprint y solucionar posibles dificultades que puedan aparecer.
3. El equipo elabora el documento del sprint siguiendo la plantilla que se puede encontrar en los anexos para mantener el registro y presentarla en la reunión.

4. Una vez finalizado el período del primer sprint, el equipo se reúne con los tutores para presentar los avances conseguidos, se valora el trabajo realizado y se planifica

el siguiente sprint, asignando nuevas tareas estimando la carga de trabajo y acordando de nuevo la fecha del siguiente sprint. Este formato de reunión será el que se deberá seguir en todos los sprints restantes, aplicando el cambio de Scrum Master cuándo sea el momento.

Ya que las reuniones con los tutores son cada dos semanas (proporcionando flexibilidad según la disponibilidad del equipo y de los mismos tutores), es muy posible que aparezcan dudas o problemas que sea importante resolver antes de la entrega del sprint. En este caso, el equipo contacta vía correo electrónico con los tutores y se les plantea la duda, si se considera necesario, se hará una reunión extraordinaria para resolver estas dudas o problemas. A nivel de equipo, siempre que sea necesario realizar una consulta entre los miembros se podrá convocar una reunión de trabajo para plantear dudas en común, encontrar soluciones, proponer mejoras o ayudar en alguna tarea.

4.3.2 Trello

En lo que respecta al Trello, se ha definido una metodología propia con el objetivo de que la herramienta proporcione valor real al equipo. Para explicarla, vamos a explicar detalladamente los diferentes elementos que forman parte de Trello, cómo configurarlo y un ejemplo de flujo de una tarea. Esta herramienta será proporcionada en la reunión inicial.

Listas

Como se puede ver en la *Figura 4.2*, Trello está formado por listas donde se almacenan las tarjetas. El tablero tendrá 3 listas principales:

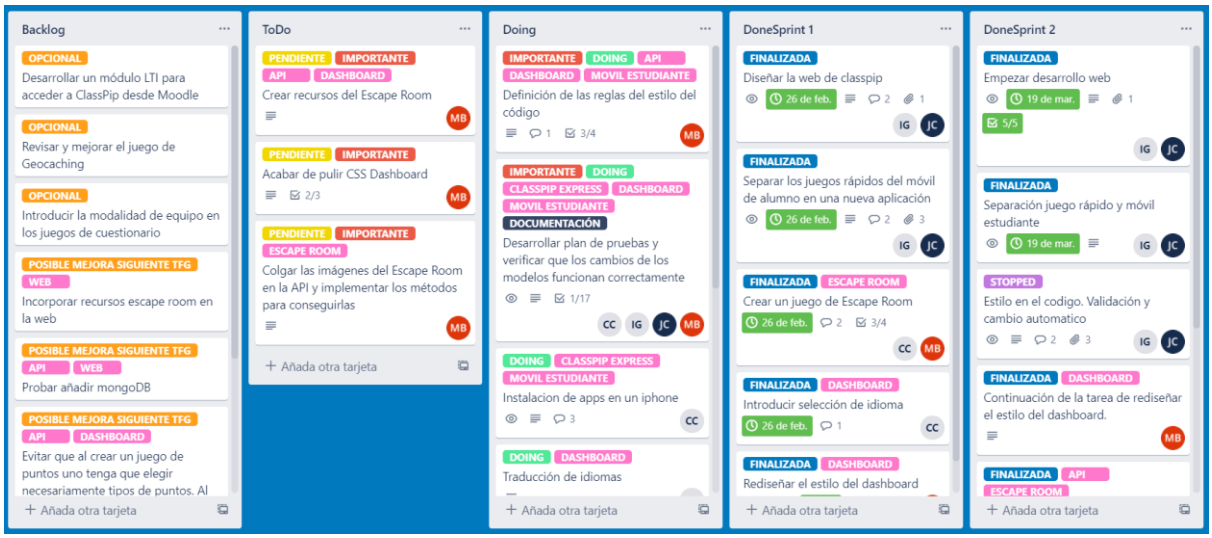


Figura 4.2 *Captura del tablero de Trello*

- ***Backlog:*** la rellenan los tutores con las posibles grandes tareas a realizar para los proyectos y llegan a un acuerdo con el equipo sobre qué tareas realizar en el primer sprint, dejando las demás en la lista para tener opción de abordarlas más adelante.

- **ToDo:** se agrupan todas las tarjetas con las subtareas derivadas de las tarjetas del Backlog (o de otras subtareas) que se deben realizar durante el sprint actual.

- **Doing:** se agrupan las tarjetas de las tareas que se están realizando actualmente.

Adicionalmente, para cada sprint se creará una lista DoneSprint + *número del sprint,* en las que se irán agrupando las tareas que estén finalizadas, permitiendo complementar el documento realizado para los sprints.

Tarjetas

Las tarjetas son el elemento clave de Trello. A continuación, describiremos las partes que tienen las tarjetas y todas las posibles opciones que tenemos.

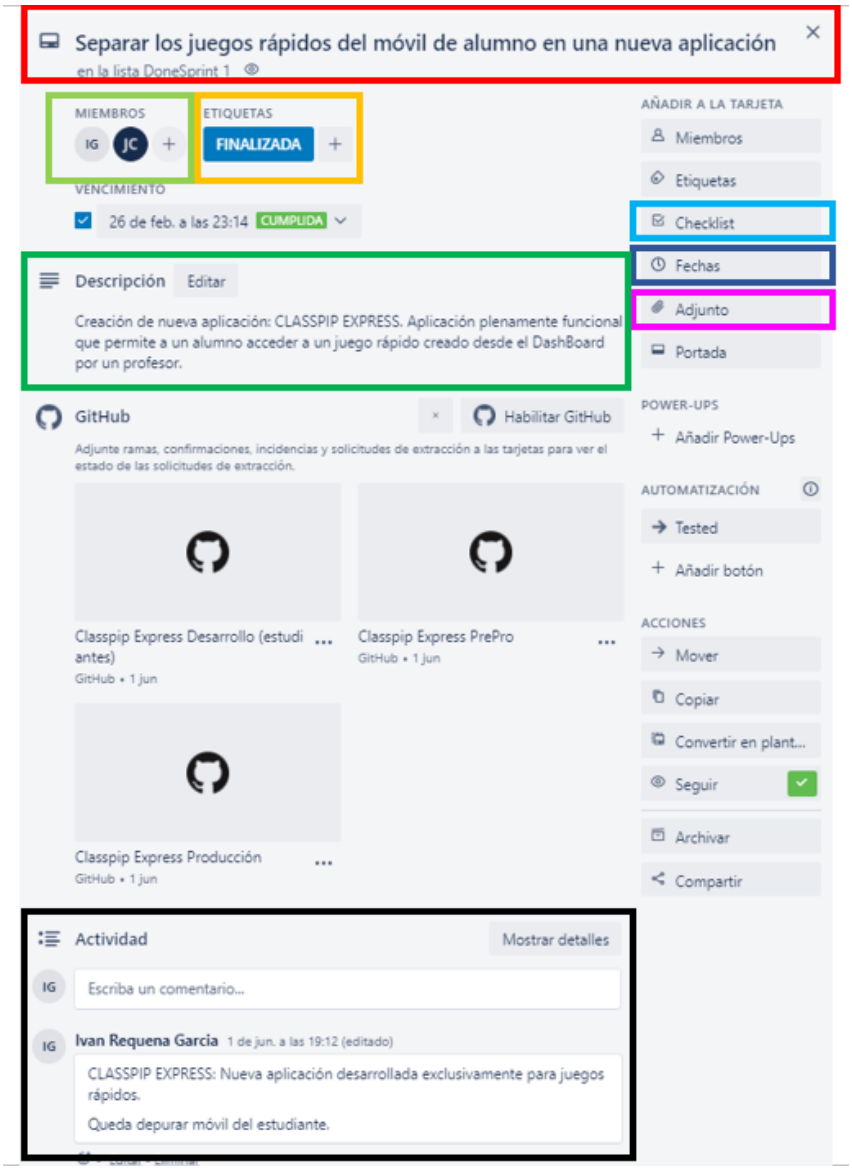


Figura 4.3 *Captura ejemplo tarjeta*

Como podemos ver en la Figura 4.3, el título de la tarjeta es el que define la tarea a realizar. Cada tarjeta se asignará a él/los miembros que la van a realizar y se le puede añadir una pequeña descripción opcional para dar más detalles. Abajo de la tarjeta se encuentra la sección de actividad, donde los miembros pueden ir realizando comentarios y actualizaciones del estado de las tareas.

Además, tenemos la opción de añadir *checklists* para dividir la tarea en pequeñas subtareas, aparte de adjuntar enlaces y documentos, o incluir las fechas del período de realización de la tarjeta.

Etiquetas

Por último, pero no menos importante, las tarjetas pueden ser etiquetadas con unas etiquetas personalizables. En nuestro caso, creamos una serie de etiquetas, como las que aparecen en la *Figura 4.4,* para poder identificar el estado de las tarjetas en la lista de Backlog, el estado de las tarjetas para las listas de ToDo y Doing y para identificar a qué aplicación pertenece la tarea.

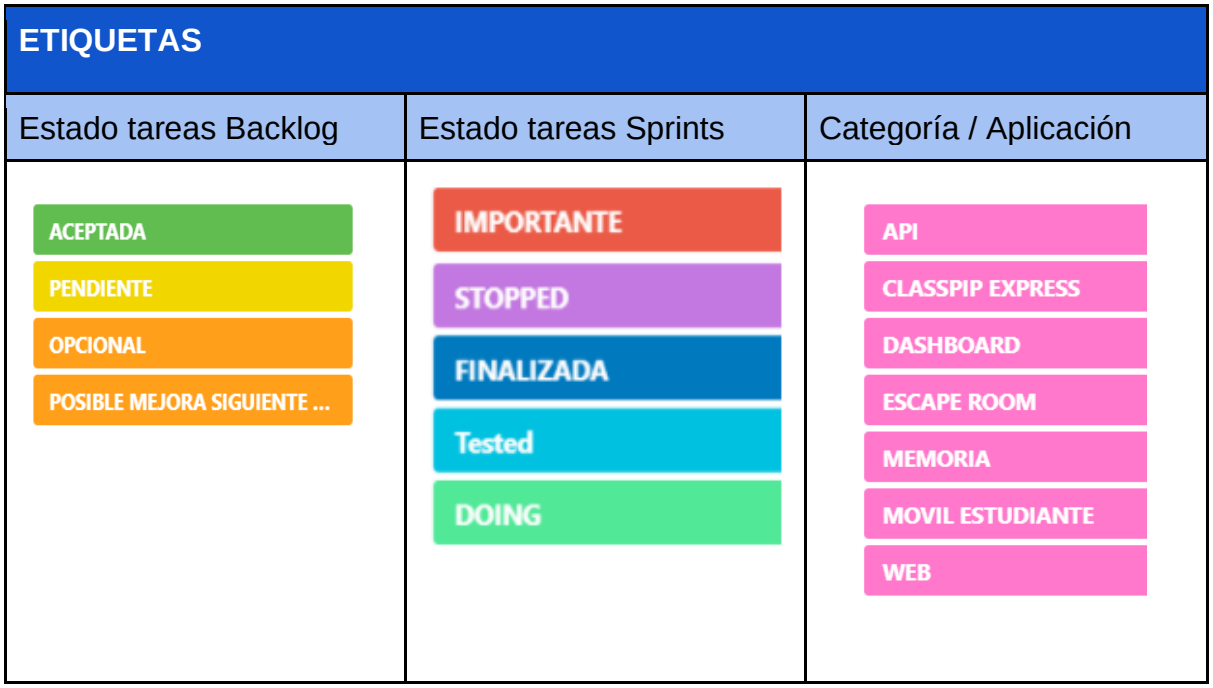


Figura 4.4 *Tabla de etiquetas creadas*

Una vez identificados todos los elementos que vamos a utilizar y etiquetadas que tarjetas del Backlog son aceptadas u opcionales, llega la hora de definir el flujo de vida de una tarjeta en el tablero. Este proceso se debe llevar a cabo en la reunión

- Una tarjeta aceptada en la lista de ***Backlog*** pasa a la lista de ***ToDo*** y se etiqueta como pendiente, tal y como observamos en la *Figura 4.5*. Se asignan los miembros que la van a realizar y se etiquetan con la categoría

correspondiente. Si es una tarea genérica, se crea una *checklist* (o varias) con las diferentes subtareas a realizar como se puede ver en el siguiente ejemplo:

Figura 4.5 *Ejemplo de tarea pendiente*

Se puede dar el caso de que la tarea tenga demasiada carga de trabajo como para dividirla en subtareas en una sola tarjeta, por lo que se pueden crear también varias tarjetas para identificar estas subtareas e ir realizándose en diferentes sprints.

- En el momento en que se empieza a desarrollar la tarea, esta pasa a la lista de ***Doing***, se le asigna la fecha máxima de realización y se le pone la etiqueta verde (Doing). La tarjeta permanecerá en esta lista hasta que la misma, o sus subtareas, estén realizadas y se irán notificando las actualizaciones importantes en la sección de actividad. En caso de algún tipo de incidencia que impida que la tarea sea realizada, se etiqueta como **STOPPED** y se indicará en la sección de actividad cuál ha sido el motivo, tal y como vemos en la Figura 4.6

Figura 4.6 *Ejemplo de tarea con incidencia*

- Una vez finalizada la tarea, se le asigna la etiqueta azul de Finalizada y se traslada la tarjeta a la lista ***Done*** del sprint en el que ha sido realizada.

Las tareas que no se vayan a realizar, se dejan en el Backlog como posible mejora.

Capítulo 5: Tareas previas

Antes de decidirnos por qué proyecto de fin de grado realizar, mantuvimos una breve reunión con los tutores de Classpip para que nos explicaran en qué consistía el proyecto, cuáles serían los objetivos y las tareas a realizar. Gracias a la documentación, los tutoriales e incluso, el código fuente, pudimos conocer en detalle de qué trataría el proyecto.

Una vez aceptamos el reto que suponía Classpip, realizamos la instalación y configuración de las herramientas: creamos una bifurcación de los repositorios para que cada uno de nosotros tuviéramos acceso al código de manera individual, y pudiéramos subir nuestras contribuciones a nuestro Github personal. Es decir, esta bifurcación se realizó con el objetivo de crear un entorno de desarrollo individual donde poder programar, y una vez comprobado el correcto funcionamiento de las nuevas implementaciones, poder incorporarlo directamente al entorno de producción. El método para subir el código desarrollado al entorno de producción es enviar una petición *pull request* para realizar un *merge* en el proyecto principal, donde uno de nuestros tutores debería comprobar y aceptar los cambios realizados.

Una vez incorporadas las herramientas, se nos propuso una tarea inicial individual y sencilla que nos sirvió para familiarizarnos con el código y su estructura.

En mi caso, la tarea consistía en implementar la funcionalidad de ver los recursos públicos de los cuestionarios y los cuestionarios de satisfacción, des del mismo componente. Es decir, que el profesor que quisiera utilizarlos tuviera la posibilidad de verlo sin necesidad de tener que copiar el recurso del profesor que lo había creado, y descargarlo en su propio repositorio.

Esta tarea me ayudó a entender mejor cómo funcionaba Classpip. Comprendí cómo estaban distribuidos los componentes del proyecto, los diferentes servicios de los que disponía y cómo debía moverme entre ellos con facilidad. Además, este primer acercamiento, también me ayudó a comprender con mayor claridad la parte técnica y aprender, de ese modo, cómo estaban distribuidos los recursos públicos, como afectan a los juegos, cuál era el mecanismo que se utilizaba para diferenciar los recursos públicos de los privados o incluso, como crearlos. Estos aspectos, han resultado muy útiles a la hora de desarrollar posteriormente todos los recursos necesarios del escape room.

Por otro lado, creo conveniente remarcar la importancia de los tutoriales de Youtube que se nos facilitaron desde el inicio sobre las diferentes herramientas y procesos del proyecto. Si bien es cierto que no fue un requisito indispensable para iniciar el proyecto, en mi caso ha resultado de gran ayuda, ya que, para realizar correctamente el trabajo, eran necesarias herramientas que no he tenido la ocasión de conocer durante el grado, como podría ser la base de datos en loopback.

En conclusión, esta primera fase de nuestro trabajo ha tenido el claro objetivo de conocer y profundizar en el proyecto para poder seguir adelante de una manera organizada y eficaz.

Capítulo 6: Diseño del dashboard por temática

Cómo ya se ha explicado anteriormente, el dashboard es la herramienta que utiliza el profesor para configurar las diferentes funcionalidades que proporciona Classpip.

Concretamente, en este capítulo, profundizaremos en cómo se ha llevado a cabo el desarrollo del diseño del dashboard para que este pueda variar en función de una temática.

El objetivo era pasar de un diseño estático a un diseño dinámico, en el que el profesor tuviera la opción de ajustar a sus preferencias el estilo del dashboard. Para poder explicarlo de una manera eficiente, mostraremos a continuación un ejemplo de cómo ha variado gracias a esta implementación dinámica.

Cómo vemos en la *Figura 6.1*, no daba la posibilidad al profesor de poder adaptarlo en función de ninguna variable:

Figura 6.1 ***Versión antigua del componente***

Sin embargo, el diseño actual que hemos realizado se distingue por las cuatro estaciones, es decir, en función de la estación que el profesor escoja, este cambiará de aspecto.

- **Invierno**

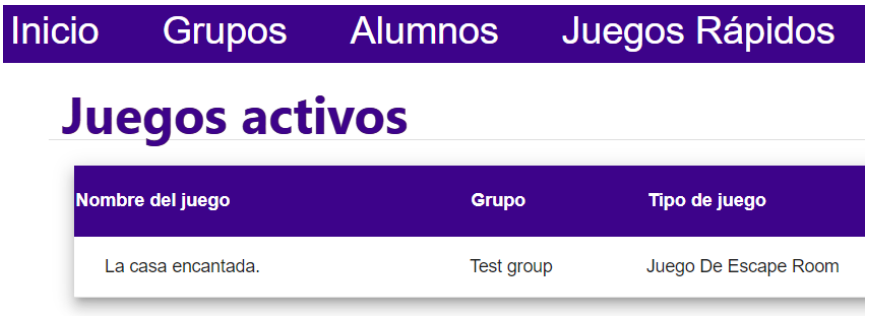


Figura 6.2 ***Versión invernal del componente***

- **Primavera**

Figura 6.3 *Versión primaveral del componente*

- **Verano**

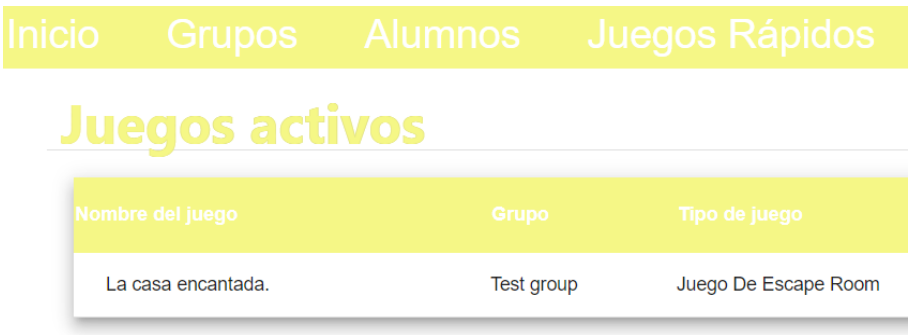


Figura 6.4 *Versión veraniega del componente*

- **Otoño**

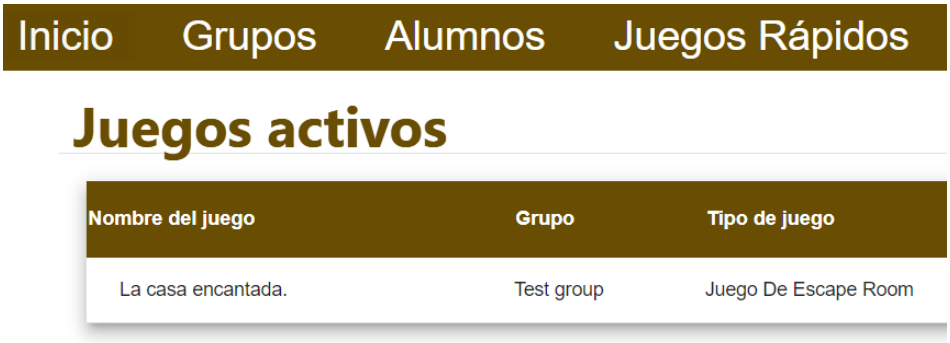


Figura 6.5. *Versión otoñal del componente*

Para poder llevar a cabo este proceso, se deberá realizar desde la pantalla de configuración que mostramos a continuación:

Figura 6.6 *Componente configuración de estaciones*

Este es el resultado del cambio de estilo en el dashboard. A continuación, se va a explicar el proceso que se ha seguido para conseguir el dinamismo en el estilo.

6.1. Proceso de creación

1. Creación de un atributo estación en la clase Profesor.
2. Modificación de cada componente en el documento HTML y typescript.
3. Añadir en el documento CSS las variables de colores del fondo según las estaciones.

A continuación, vamos a ver un ejemplo dónde se pueda ver como se ha configurado:

- **CSS** (*styles.scss*)

```
:root{
  --main-color-summer: rgb(245, 247, 134);
  --main-color-spring: rgb(128, 245, 128);
  --main-color-autumn: rgb(104, 77, 3);
  --main-color-winter: rgb(61, 3, 138);
}
```

Figura 6.7 *Variables de colores en el style.scss*

```css
.titulosummer{
  width: 70%;
  margin-left: 15%;
  height: 60px;
  padding: 10px;
  color:var(--main-color-summer);
  font-size: 2.5rem !important;
  font-weight: bold;
}
.titulospring{
  width: 70%;
  margin-left: 15%;
  height: 60px;
  padding: 10px;
  color:var(--main-color-spring);
  font-size: 2.5rem !important;
  font-weight: bold;
}
```

```css
.tituloautumn{
  width: 70%;
  margin-left: 15%;
  height: 60px;
  padding: 10px;
  color:var(--main-color-autumn);
  font-size: 2.5rem !important;
  font-weight: bold;
}
.titulowinter{
  width: 70%;
  margin-left: 15%;
  height: 60px;
  padding: 10px;
  color:var(--main-color-winter);
  font-size: 2.5rem !important;
  font-weight: bold;
}
```

Figuras 6.8 y 6.9. ***Ejemplo de la configuración del título por estaciones***

Como se puede observar en las imágenes, se declara una variable global para cada estación (Figura 15), y para cada elemento del dashboard (títulos, tablas, etc) en el que queramos introducir el dinamismo, se hace un duplicado del formato para cada estación en el que únicamente cambiará el color (Figura 16 y Figura 17). No hemos podido simplificar más el código, ya que el CSS no permite introducir parámetros, son variables estáticas.

- **TYPESCRIPT** (*inicio.ts*)

```typescript
export class InicioComponent implements OnInit {

  displayedColumns: string[] = ['nombre', 'grupo', 'tipo', 'modo', ' '];
  dataSource;
  juegosActivos: Juego[] = [];
  listaGrupos: Grupo[];
  profesor: Profesor;
  tabla: any [] = [];

  varTituloColumnaTabla: string;
  varTitulo: string;

  date: Date = new Date();
  settings = {···
  }
  imagenSilueta = 'http://147.83.118.92:3000/api/imagenes/imagenColeccion/download/compositores.png';

  constructor(  private peticionesAPI: PeticionesAPIService,···
                private router: Router) { }


  @ViewChild(MatSort) sort: MatSort;

  ngOnInit() {
    this.profesor = this.sesion.DameProfesor();
    this.varTitulo = "titulo" + this.profesor.estacion;
    this.varTituloColumnaTabla = "tituloColumnaTabla" + this.profesor.estacion;
    this.ObtenJuegosActivosDelProfesor();
  }
```

Figura 6.10 ***Inicio.ts***

En la figura 18, podemos ver que por cada componente typescript tenemos que crear el número de variables correspondientes a los elementos dinámicos que haya en esa pantalla, por ejemplo, en el inicio.ts se tiene que modificar según la temática el título de la página y el formato de la tabla, por lo que creamos las variables *varTitulo* y *varTituloColumnaTabla*.

A continuación, vamos a mostrar cómo se debe modificar el HTML.

- **HTML** (*inicio.html*)

```
<body class = "backgroundInicio">
<div class={{varTitulo}}>Juegos activos</div>
<mat-divider style="width: 70%; margin-left : 15%"></mat-divider>
```

```
<ng-container matColumnDef="nombre">
  <th mat-header-cell class= {{varTituloColumnaTabla}} *matHeaderCellDef> Nombre del juego </th>
  <td mat-cell style= "text-align: left" *matCellDef="let juego"> {{juego.nombre}} </td>
</ng-container>
```

Figuras 6.11 y 6.12. *Configuración actual del inicio.html*

En la figura 19, vemos que, en vez de poner una variable estática, se le pasa la variable *varTitulo* que hemos visto en el apartado del TYPESCRIPT.
En la segunda vemos el mismo ejemplo, pero para el formato de la tabla, en vez de tener una variable estática, se le pasa la variable *varTituloColumnaTabla*.

Capítulo 7: Diseño del estándar

En este capítulo se va a hacer una breve descripción del motivo por el que surgió la tarea, el objetivo y como se ha llevado a cabo.

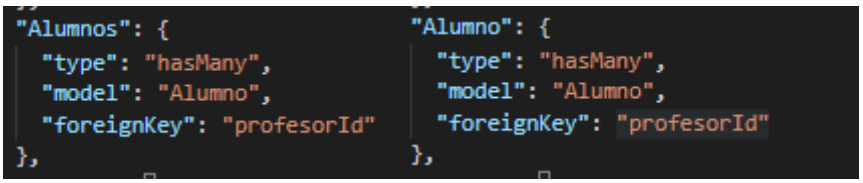
El estándar es una serie de directrices sobre el estilo que ha de tener el código, con el objetivo de que quede un código uniformizado, ya que al programar sobre un proyecto común, se acumulan errores si no hay un esquema bien definido, y como consecuencia, se crea una dificultad añadida, y de no corregirse se podría complicar mucho la programación del proyecto.

Ejemplo sobre un problema típico:

- La mecánica utilizada por la mayoría de desarrolladores, es configurar las relaciones entre modelos en mayúsculas y en plural, pero un programador X, configura la relación entre Profesor y Alumno en singular, y como consecuencia, se crea la siguiente URL:

Figuras 7.1 y 7.2 *URL creada a partir de la relación entre Alumnos y Profesores*

El error que se ha descrito como ejemplo en el párrafo anterior, es el que se muestran en las *Figuras 7.1 y 7.2*, REVISAR LA NUMERACION DE LAS FIGURAS y es que como podemos observar, escribir una letra más o una letra menos, marca la diferencia entre que funcione o que no.



Figuras 7.3 y 7.4 *Relación Alumnos con Profesores*

Como consecuencia del error descrito, si el desarrollador X no se da cuenta de su error, lo más probable es que el siguiente programador que quiera hacer esa petición a la base de datos, no sepa porque no puede ser atendida, provocando que, por no tener unas normas básicas de código, se vayan acumulando errores y se compliquen aspectos que a priori deberían tener una complejidad menor.

De todos los apartados que conforman un proyecto, nos hemos querido centrar en los modelos de datos, ya que son el fundamento principal y muchos de los errores provienen de ahí. A partir de ellos, se crean las clases con su nombre y sus respectivos atributos, se crean las peticiones HTTP y tanto en el html como en el typescript se ha de respetar el formato del Objeto (clase) para su correcto funcionamiento.

Por ese motivo, las reglas que se indican en el estándar han sido enfocadas en dichos modelos.

En el anexo hay un documento, anexo 4, con el proceso que se tiene que seguir cuando se quiere crear un nuevo modelo de datos, desde la creación del modelo en loopback, la creación de la clase y la implementación en el código.

A continuación, vamos a ver un ejemplo de cómo ha de quedar un modelo de datos, para clarificar la explicación:

Modelos de datos

- El modelo de datos / clase en angular, se debe crear con la primera letra en mayúscula. Ejemplo: *"Profesor".*
- Todos los atributos de dicho modelo van a estar escritos con la primera letra minúscula. Ejemplo: *"nombre"*.
- A la hora de crear el modelo en loopback, se nos permite especificar el nombre público que se tendrá que indicar en la URL para hacer las peticiones a la API. Este nombre estará escrito con la primera letra en mayúsculas y en plural. Ejemplo: *"Profesores".* En caso de que sea un modelo con una palabra compuesta, como, por ejemplo, ObjetoJuego, el nombre público sería el siguiente: *"ObjetosJuego"*, es decir, siempre indicando el plural en la característica principal del modelo, que en este caso es que es un tipo de *Objeto.*
- Se debe especificar otro parámetro a la hora de crear un modelo, que es el nombre público de las relaciones. Por ejemplo, en el caso de que existiera una relación 1:N entre Profesor y Juego, es decir, que un profesor puede tener diferentes juegos, pero un juego solo puede tener un profesor, se crearía una relación en el modelo Profesor (se podría poner también en la del Juego) y dicha relación sería indicada con la primera letra en mayúsculas y en plural. Ejemplo: La relación entre Profesor y Juego dentro del modelo Profesor sería *"Juegos"*.

- **Loopback - Modelo *Profesor***

Figura 7.5 *Modelo Profesor en la página localhost:3000/explorer/*

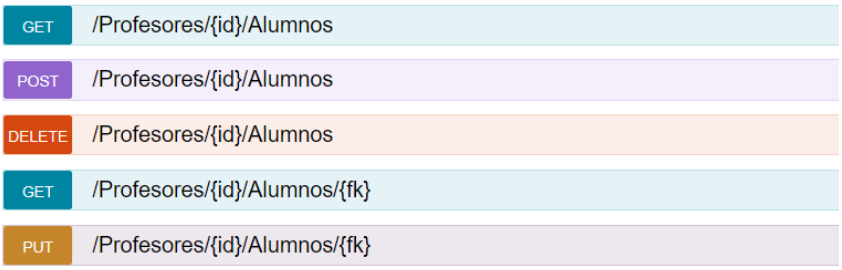


Figura 7.6 *URL creada por la relación entre profesores y alumnos*

- **Angular**

```
export class Profesor {

  nombre: string;
  primerApellido: string;
  segundoApellido: string;
  nombreUsuario: string;
  email: string;
  password: string;
  imagenPerfil: string;
  estacion: string;
  identificador: string;
  id: number;

  constructor(  Nombre?: string, PrimerApellido?: string, SegundoApellido?: string, ···
  }

}
```

Figura 7.7 *Clase Profesor.ts*

Además de dejar definido el estándar, que se ha de llevar a cabo en los nuevos modelos, se ha implementado en las herramientas más desarrolladas de Classpip, que son el Dashboard y el móvil del estudiante. Para ello, se ha tenido que modificar lo siguiente:

93 Modelos en loopback.

- **Dashboard**:
  - Las respectivas clases a los modelos de loopback.
  - 97 componentes, en los cuales se tenía que modificar tanto el typescript como el documento HTML.
  - Los servicios de *Sesión, Cálculos* y *Peticiones*.
- **Móvil de estudiante**:
  - Las respectivas clases a los modelos de loopback.
  - 31 componentes, en los cuales se tenía que modificar tanto el typescript como el documento HTML.
  - Los servicios de *Sesión, Cálculos* y *Peticiones*.

Mis compañeros Iván y Jordi lo han implementado en la Web y el Classpip Express.

Capítulo 8: Escape room

En este capítulo, se va a explicar todos los aspectos relacionados con el escape room, empezando por una descripción general y mencionando porque hemos decidido desarrollar un juego de este estilo en una herramienta de gamificación, y siguiendo por los diferentes apartados que forman el juego, que son la mecánica, los modelos de datos implementados, el desarrollo técnico e implementaciones que se pueden considerar en un futuro.

8.1 Descripción

El nombre Escape room es la descripción literaria del objetivo que tiene el juego, escapar de una habitación. Es un juego mental en el que ya sea de manera individual o de manera colectiva, se ha de resolver una serie de pruebas, normalmente dentro de un límite de tiempo definido, con el objetivo de escapar del lugar donde se realizan dichas pruebas.

Inicialmente, este tipo de juego de escapismo fue diseñado para realizarse presencialmente, en el que destacaba la ambientación y los diferentes obstáculos que te encontrabas dependiendo del escape room que escogieras, ya que cada uno tiene una historia propia con diferentes personajes y diferentes escenarios.
A medida que han ido aumentando el número de empresas dedicadas a la creación de juegos de escape room, y impulsados por la necesidad de la época actual, se han diseñado juegos de escape room virtuales, juegos que tienen el mismo objetivo que los juegos presenciales, con la diferencia de que la jugabilidad de uno es de manera presencial y la jugabilidad del otro es de manera telemática.

Una de las grandes ventajas y que a la vez es uno de los motivos por los cuales la incursión de este tipo de juegos ha sido tan brusca en estos últimos años, es la gran versatilidad que tiene. Es un estilo de juego que depende totalmente de la creatividad de los creadores, no hay nada estipulado, los escenarios son diferentes, el tipo de pruebas a resolver, los personajes… la única característica que comparten es que el cliente tiene que escapar de un lugar. Como consecuencia, al haber infinidad de temáticas, estilos de escape… El público que tiene este tipo de juegos es enorme, cualquier persona tiene a su disposición un estilo de escape que sea acorde con sus gustos.

Este tipo de juegos también tienen un gran inconveniente que a la vez es una gran ventaja, y es que es un juego de un solo uso, una vez has jugado a un determinado escape room, no se puede repetir ya que sabes como resolver todos los enigmas, sería como jugar a un juego con la solución delante.

El gran inconveniente que tiene es fácil de entender, al ser de un solo uso, no se puede volver a disfrutar de la experiencia que conlleva realizar este tipo de juegos ya sea de manera individual o de manera colectiva.
Es más complicado de ver que esto también conlleva una ventaja, y es que no hay competencia real, como cada escape room es totalmente diferente al resto de escapes y solo puedes disfrutarlo una vez, lo que más interesa es que el cliente realice

cuantos más escapes mejor, por lo que la comunidad que se ha creado alrededor del mundo del escapismo es una comunidad muy sana. Un hecho en el que se ve reflejado, es que normalmente al final del juego, en el momento en el que conoces a los actores y creadores del juego, te recomiendan y te proporcionan una serie de descuentos si realizas otros escapes.

A continuación, vamos a ver en detalle todos los apartados referentes al juego de escape room que se ha diseñado para este proyecto, empezando por la mecánica de este.

8.2 Mecánicas del juego

Actualmente el juego está diseñado para que haya dos personas implicadas, el alumno y el profesor.
Con las herramientas proporcionadas, hemos diseñado un juego en el que se refleje lo máximo posible las características que hemos comentado que tiene este modelo de juego, y es que sea entretenido y que se tengan que superar diferentes enigmas para completar la historia y conseguir escapar, pero a la vez se ha diseñado con aspectos que consideramos que se distinguen del estilo estándar del tipo de juegos.

Uno de los objetivos principales que nos hemos marcado a la hora de diseñar el escape, es que el profesor tenga a su disposición el máximo de herramientas para que cada juego que diseñe pueda ser completamente diferente al anterior, aspecto en el que se diferencia de los juegos presenciales, debido a las facilidades que se nos presentan por el simple hecho de ser un juego virtual. El otro aspecto que se diferencia del resto es que al final el proyecto Classpip es un proyecto de docencia, por lo que hemos diseñado el juego para que se pueda convertir en un juego didáctico.

Durante los diferentes apartados de este capítulo vamos a entrar en detalle en todo el entorno que conlleva el juego, empezando por a ver las funcionalidades que tiene en primer lugar el usuario Profesor y en segundo lugar el usuario Alumno:

8.2.1 Profesor

El profesor va a diseñar el juego desde el dashboard, que es la herramienta principal del Profesor donde lleva a cabo toda la configuración de Classpip.

El juego tiene dos componentes principales:

- Escenarios
- Objetos

El juego está dividido por escenas, cada escena tendrá un escenario y diferentes objetos.

Los escenarios y los objetos serán creados por el profesor y serán depositados en la página de recursos, que podemos observar a continuación:

- **Página principal de recursos**

Figura 8.1. ***Recursos escenarios del escape***

Figura 8.2. ***Recursos objetos del escape***

- **Escenario**
    - *Mis escenarios*

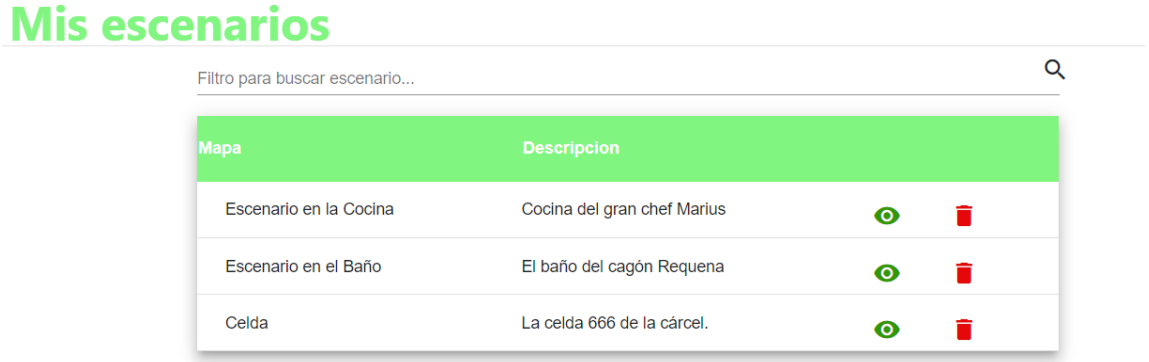


Figura 8.3. ***Componente mis escenarios en la herramienta dashboard***

La página de Mis escenarios tiene las siguientes funcionalidades: Ver los escenarios que ha creado el profesor, con el nombre y la descripción, con las opciones de ver la imagen asociada a dicho escenario y poder borrarlo.

Figura 8.4 ***Ejemplo del escenario Cocina en la herramienta dashboard***

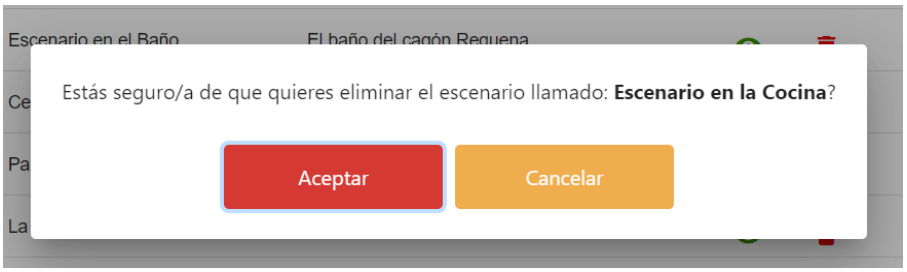


Figura 8.5 ***Función eliminar escenario en la herramienta dashboard***

También tiene un filtro para encontrar el escenario con cualquier palabra que encuentre tanto en el nombre como en la descripción.

- *Crear escenario*

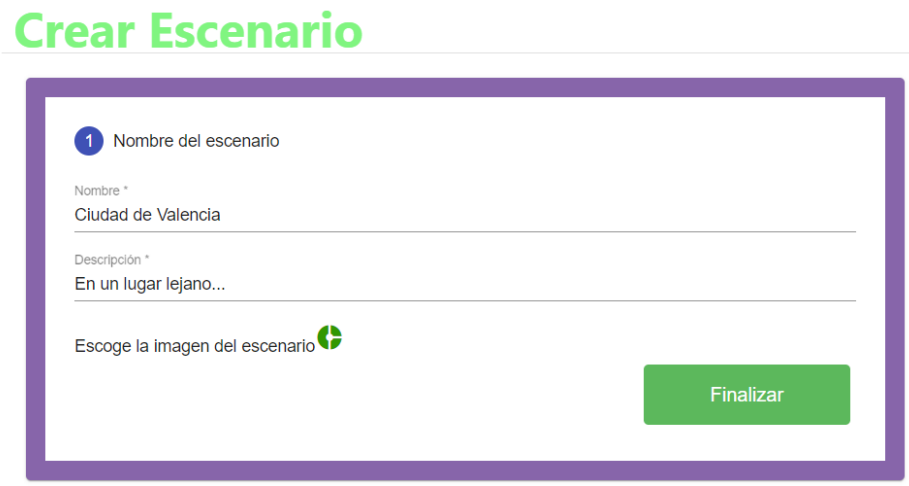


Figura 8.6 ***Componente "crear escenario" en la herramienta dashboard***

Los parámetros necesarios para crear un escenario son el *Nombre*, una pequeña *Descripción* y escoger la imagen asociada a ese escenario.

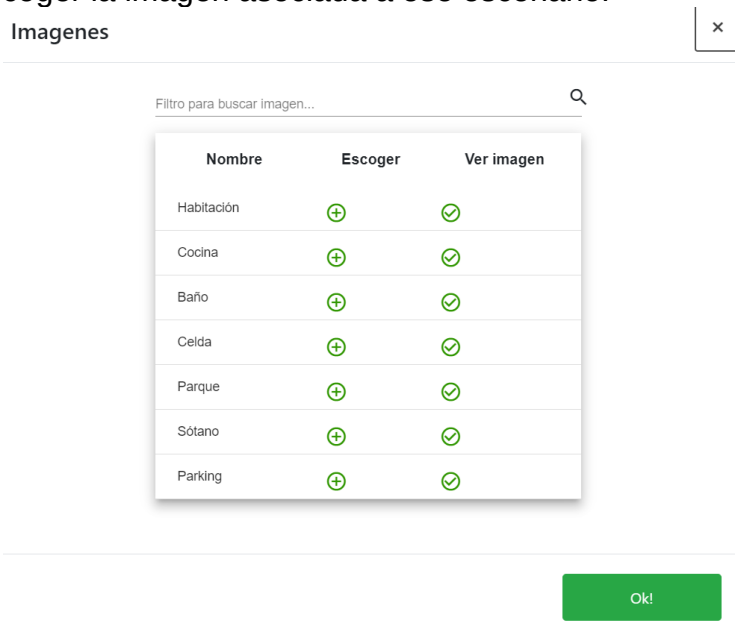


Figura 8.7 ***Ejemplo de diferentes imagenes escenario***

Figura 8.8 ***Ejemplo de la función ver imágen en la herramienta dashboard***

En la imagen podemos observar que se marcan las diferentes posiciones que se tendrán que configurar cuando el profesor cree la escena. A continuación, vamos a ver los objetos.

- **Objetos**
  - *Mis objetos*

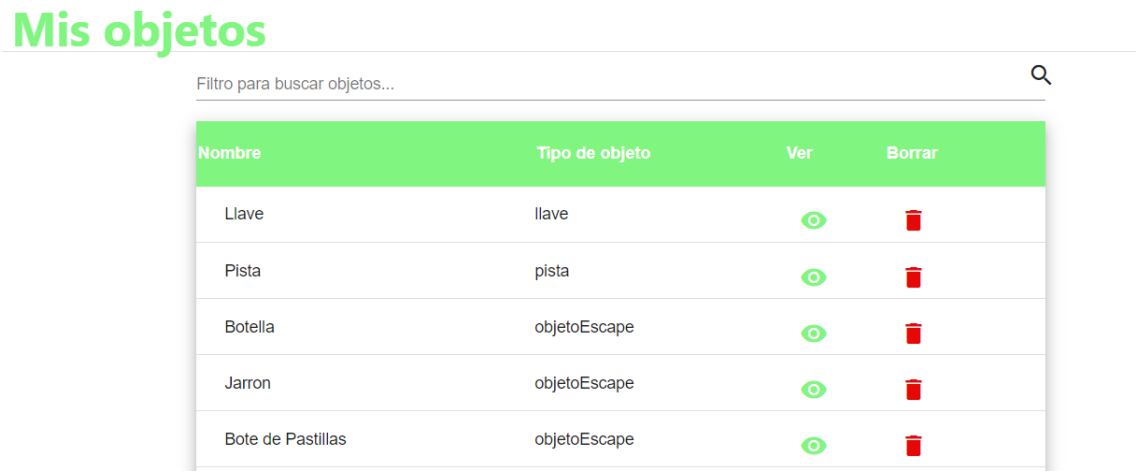


Figura 8.9 ***Componente mis objetos en la herramienta dashboard***

La página de Mis objetos tiene las siguientes funcionalidades: Ver los objetos que ha creado el profesor, con el nombre y el tipo de Objeto, con las opciones de ver la imagen asociada a dicho objeto y poder borrarlo.

Figura 8.10 ***Ejemplo de la imagen del objeto botella***

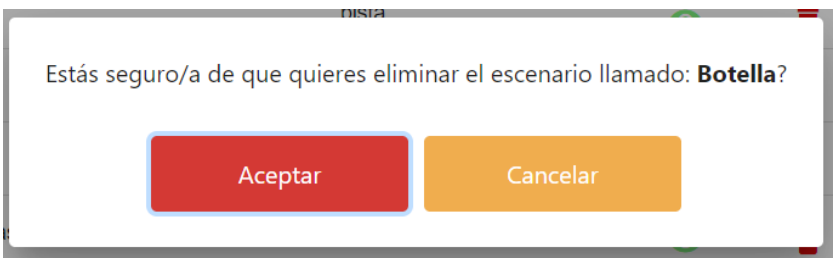


Figura 8.11 ***Ejemplo de la función eliminar objeto***

También tiene un filtro para encontrar el objeto con cualquier palabra que encuentre tanto en el nombre como en el tipo de objeto.

- *Crear objeto*

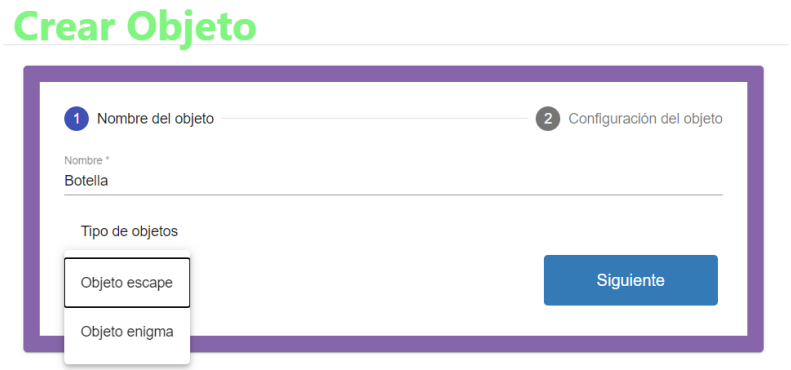


Figura 8.12 *Componente crear objeto en la herramienta dashboard I*

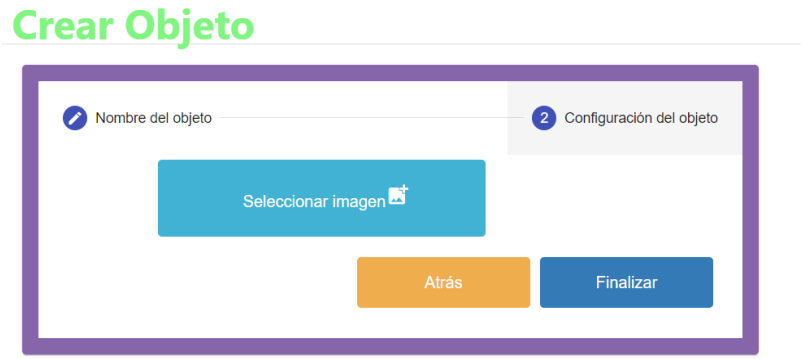


Figura 8.13 *Componente crear objeto en la herramienta dashboard II*

El objetivo que me propuse fue que se pudieran reutilizar tanto los objetos como los escenarios para todos los juegos que el profesor quisiera crear, por ese motivo se encuentran en recursos y se tienen que crear antes de empezar a crear el juego, ya que podemos decir que son independientes. Esto se consigue de manera que no le vincules ningún atributo que haga referencia al juego, tanto en las clases de Objeto como en las de Escenario, y que tampoco tengan relación entre sí.

Esta es una de las dos principales funciones que tiene el profesor a la hora de configurar el juego, la otra función principal que tiene es la creación del escape room, que está la encontramos documentada y guiada en el anexo de la memoria.

Para ir entendiendo mejor el funcionamiento, voy a ir explicando la mecánica con un ejemplo simultáneamente. La mecánica del juego es la siguiente:

El profesor va a decidir cuantas escenas quiere que tenga el juego, y todas se van a configurar de la misma manera.

Por ejemplo, si el profesor desea diseñar un juego de 2 escenas, tendrá que elegir el escenario que quiere vincular a cada escena y los objetos
Como hemos mencionado anteriormente, habrá cuatro tipos de objetos:

- Objetos escape
- Objetos enigma
- Pistas
- Llaves

Para cada escena el profesor deberá escoger 3 objetos escape, 2 objetos enigma, 1 pista y 1 llave.

- **Objetos Escape** → Serán un complemento necesario para poder pasar de escena junto a la llave. El profesor decidirá qué objeto es necesario para cada escena, teniendo la opción de poner ese objeto como requisito de escenas posteriores. Por ejemplo, si el profesor decide poner una botella en la primera escena, podrá escoger que en esa misma escena el alumno pueda salir sin recogerla, pero que sea necesaria para poder salir de la segunda escena.

- **Objetos Enigma** → Habrá dos tipos de objetos enigma, los principales y los secundarios, habiendo uno de cada tipo por escena. Serán principales los objetos enigma que al resolverlos te den el objeto Llave, mientras que serán objetos enigma secundarios los que al resolverlos te den el objeto Pista. El objeto tendrá una pregunta y una respuesta que se configurará al escogerlos.

- **Pistas** → Tendrán un texto configurable con el objetivo de ayudar a superar esa escena al alumno.

- **Llave** → La única función que tiene es que es necesaria para pasar a la siguiente escena, obligando al alumno a tener que resolver al menos un objeto enigma.

Los objetos tendrán una posición dentro de la escena, otro de los aspectos que tendrá que configurar el profesor y que podemos ver con detalle en la guía del anexo.

Una vez explicado el formato del juego, pasamos a describir el rol del alumno en el desarrollo del escape room.

Antes de entrar en detalle, mencionar que durante la defensa del proyecto se presentará una demostración visual, en la que podremos observar mediante un ejemplo, los pasos a seguir en la creación del juego por parte del profesor y como sería la jugabilidad del alumno.

8.2.2 Alumno

El juego está diseñado para ser utilizado en el móvil del alumno, por lo que el alumno tendrá que acceder a su cuenta para poder jugar.
Cada alumno tendrá tantos juegos como le haya asignado su profesor y podrá entrar y salir de cada juego sin ningún problema.

El procedimiento que seguirá el alumno al empezar un juego será el siguiente:

- Una vez logeado entrará dentro del juego asignado. Si es la primera vez que juega a ese juego, nos indicará que empecemos la historia. En caso contrario, que continuemos con nuestra historia.

- Cuando empezamos una historia, nos permite escoger un personaje de los 4 que se nos ofrece.

Una vez empezado el juego, el alumno tendrá el objetivo de escapar de los escenarios que haya configurado el profesor. Como hemos mencionado anteriormente, la mecánica será ir recogiendo los diferentes objetos escape y resolviendo los diferentes enigmas, que le permitirán ir encontrando las diferentes llaves y pistas escondidas por los mapas.

Dentro de lo que es diseño del juego, el alumno dispondrá de una mochila en la que puede observar los objetos recogidos y que todavía no ha utilizado.

Por último, mencionar que el alumno podrá parar en cualquier momento de jugar, ya que hemos incorporado la función de guardar el juego, para que cuando vuelva a reanudar la historia, aparezca en el lugar en el que lo dejó.

8.3 Descripción de los modelos de datos

A continuación, vamos a describir qué modelos de datos se han utilizado para desarrollar el juego y qué relaciones hay entre ellos.

Podemos clasificar los modelos en principales y secundarios:

- **Principales**. Son los modelos base del juego, y son los siguientes:
    - Alumno
    - Juego de Escape Room
    - Escena de juego
    - Escenario
    - Objeto global

- **Secundarios.** Son los modelos que están creados a partir de las relaciones entre los modelos principales, y son los siguientes:
    - Alumno juego de escape room
    - Partida escape
    - Objeto juego

8.3.1 Relaciones N:M entre modelos

Una relación N:M entre dos modelos se puede describir de la siguiente manera:

- Un modelo A, puede tener N modelos B, mientras que el modelo B puede tener M modelos A.
  Ejemplo: Un juego de escape room puede tener N escenas de juego, y una escena de juego puede tener M juegos de escape room. En la práctica, sería lo siguiente: 1 Juego de escape con 2 escenas, y por ejemplo la primera escena de ese juego se está utilizando en otros 3 juegos diferentes.

Para las relaciones N:M se puede crear un modelo auxiliar, que tiene el objetivo de almacenar información relevante para los dos modelos.

En la *Figura 8.14* NUMERACION, podemos observar las relaciones N:M de los modelos de datos del escape room.

Figura 8.14 ***Relaciones N:M de los modelos de datos del escape room***

- Relación entre juego escape room y alumno:

En la figura 40, no se muestran todos los atributos, únicamente se muestran los parámetros que tienen implicación en las relaciones. En este caso, la relación implica que un juego podrá tener M alumnos, y un alumno podrá tener simultáneamente N juegos.

Dicha relación se podría programar de dos maneras: la primera sería crear una lista de identificadores tanto en la clase juego escape room como en la clase alumno para que se pudieran referenciar, y la segunda, que es la que se ha escogido, es crear una clase auxiliar en la que se albergará tanto el identificador del juego como del alumno, e inclusive, otro tipo de atributos como puede ser el personaje.

La implicación que tiene esta clase auxiliar es la siguiente: Si por ejemplo, en total el profesor ha creado 2 escape rooms diferentes, y ha incluido a dos alumnos distintos

en cada escape room, se crearán 4 modelos alumno escape room en la base de datos, tal que así:

	Identificador juego	Identificador alumno
Alumno escape room 1	1	1
Alumno escape room 2	1	2
Alumno escape room 3	2	1
Alumno escape room 4	2	2

Tabla 8.1. Ejemplo clase "alumno escape room"

Actualmente, el juego de escape room está diseñado para jugar una persona, por lo que realmente ahora se está utilizando como N:1 (1 alumno puede tener N juegos, pero un juego solo puede tener un alumno), pero se ha diseñado de esta manera porque una de las implementaciones futuras es implementar el modo grupo en el juego.

- Relación entre juego escape room y escena:

En la figura 40, la segunda relación N:M que podemos observar es la de juego escape room y escena. Esta es la relación que se ha utilizado de ejemplo en los párrafos anteriores para describir las N:M.
Igual que en el caso anterior, se ha creado una clase auxiliar, llamada partida escape, con la función de guardar los identificadores de la clase juego y de la clase escena e indicar en qué posición, por ejemplo: En 1 juego con 3 escenas, se crearían las siguientes partidas escape:

	Identificador juego	Identificador escena	Posición
Partida escape 1	1	1	1
Partida escape 2	1	2	2
Partida escape 3	1	3	3

Tabla 8.2. Ejemplo clase "partida escape"

En el ejemplo, el identificador de escena coincide con la posición, pero no será siempre así, esto es únicamente uno de los muchos casos que se pueden dar.

8.3.2 Relaciones 1:N entre modelos

Una relación 1:N entre dos modelos se puede describir de la siguiente manera :

- Un modelo A, puede tener N modelos B, mientras que el modelo B únicamente puede tener 1 modelo B.
  Ejemplo: Una escena de juego puede tener N objetos juego, pero un objeto juego únicamente puede tener 1 escena de juego. En la práctica, sería lo siguiente: 1 Escena de escape room diseñada con 5 objetos juego, y esos objetos juego únicamente se utilizan en dicha escena.

En los modelos en los que se establece una relación 1:N, se crea la relación a través de un atributo, llamado *foreign key,* en el modelo B. Siguiendo el ejemplo anterior, dentro del modelo objeto juego, se creará un atributo que hará referencia a la escena en la que se esté usando ese objeto juego, normalmente se define como idEscena cuando se crea la relación.

En la figura 41, podemos observar las relaciones 1:N de los modelos de datos del escape room.

Figura 8.15 ***Relaciones 1:N de los modelos de datos del escape room***

- Relación entre escenario y escena:

Esta relación se ha creado con el propósito de poder reutilizar el modelo escenario. El profesor que, por ejemplo, diseñe el escenario Cocina, podrá utilizarlo en tantas escenas como el desee, pero en dichas escenas, únicamente habrá el escenario Cocina.

- Relación entre objeto global y objeto juego:

El objeto global, tiene los siguientes atributos:

- Nombre del objeto
- Tipo de objeto
- Imagen del objeto
- Profesor id

En el Objeto juego, se almacena la información referida al juego:

- Recogido
- Usado
- Pregunta
- Respuesta
- Principal
- ...

El propósito de dicha relación es separar de los objetos, las características globales de las características que pertenecen al juego en el que se utilizan. De esta manera, el profesor va a poder reutilizar el objeto tantas veces como quiera, por ejemplo: Si el profesor ha creado un objeto que se llama “Pelota”, se creará en la base de datos un objeto global llamado “Pelota” con la imagen y el tipo de objeto escogido. Cuando se pretenda utilizar en un juego, se creará un modelo de objeto juego, en el que se referenciará con un atributo, objetoGlobalId, igual que en el caso anterior.

- Relación entre escena y objeto juego:

Esta es la relación utilizada en el ejemplo del primer párrafo donde se describen las 1:N. En cada objeto juego, se creará una *foreign key,* llamada escenaId, que hará referencia a la escena en la que se utiliza el objeto, ya que el objeto juego se creará únicamente para ser usado en esa escena, ya que como se ha mencionado anteriormente, tiene las propiedades relacionadas con la mecánica del juego (recogido, usado…).

- Relación entre juego escape room y objeto juego:

Por último, la relación entre juego escape room y objeto juego tiene la misma finalidad que la relación anterior entre escena y objeto juego, y es que el objeto únicamente pertenece a un juego escape. A priori se podría pensar que esta relación es redundante, ya que tenemos la escena - objeto juego, pero como hemos explicado, una escena puede ser reutilizada en diferentes juegos, y no siempre se van a configurar los mismos objetos para esa escena en concreto, ya que lo más seguro es que estén configuradas para diferentes juegos, por lo que también se ha de incluir esta relación. Un ejemplo para clarificar: 2 juegos diferentes, con la misma escena y 2 objetos juego diferentes por escena:

	Identificador juego	**Identificador escena**	**Identificador objeto**
Objeto juego 1	1	1	1
Objeto juego 2	1	1	2
Objeto juego 3	2	1	3
Objeto juego 4	2	1	4

Tabla 8.3 Relaciones 1:N de los modelos de datos del escape room

8.4 Programación del escape

En este apartado se va a describir la parte técnica del proyecto, que modelos se han creado y modificado para el escape room, cuántos componentes se han necesitado y dificultades que hemos tenido con la mecánica del juego.

8.4.1 Modelos de datos

- *Alumno juego escape room.*
- *Juego de escape room*
- *Escena de juego*
- *Partida escape*
- *Escenario escape room*
- *Imagen escenario*
- *Objeto global escape*
- *Objeto juego*

También se han modificado los siguientes modelos debido a las relaciones:

- ***Alumno***. Se ha añadido las relaciones con Juego.
- ***Profesor***. Se ha añadido las relaciones con Juego, ObjetoGlobal y EscenarioEscape.

Para cada modelo, se ha creado una clase correspondiente en el dashboard y en el móvil del estudiante. También se han creado clases auxiliares, que no se ha considerado necesario crear los modelos correspondientes en la base de datos, pero que nos han servido de ayuda en la programación del juego. Estas clases son: ObjetoEscape, ObjetoEnigma, Pista, Llave y Mochila.

8.4.2 Componentes

En este apartado hay que diferenciar entre herramientas, por un lugar tenemos el dashboard como herramienta del profesor, y por otro el móvil de estudiantes como herramienta del alumno.

- **Dashboard**

Para la configuración del escape se han tenido que crear los siguientes componentes (o se han tenido que ampliar):

- Para los recursos del profesor:
    - *Crear escenario escape room*
    - *Ver escenario escape room*
    - *Crear objeto escape room*
    - *Ver objeto escape room*

- Para la creación del escape room:

    - Se ha añadido todo el código referente a este apartado en el componente *juego*.

- **<u>Móvil de estudiantes</u>**

Para la configuración del escape se han tenido que crear los siguientes componentes (o se han tenido que ampliar):

- *Se ha modificado el componente inicio.*
- *Juego de escape room*
- *Escoger avatar*
- *Escenario*
- *Mochila*
- *Mochila de pistas*

8.4.3 Dificultades en el proceso

Técnicamente hablando, una de las mayores dificultades que he tenido ha sido el proceso de mapear la información. Es decir, una vez definidos los modelos y con la idea clara de cómo quería que fuese la mecánica del juego, el recoger toda la información de la base de datos, mapearla a las clases locales de la manera más eficiente posible, no fue tarea fácil.
Una de las herramientas que no estaba acostumbrado a utilizar y que me ha ayudado mucho a la hora de almacenar la información, son los mapas. Hasta ahora siempre había tenido preferencia por las listas de objetos, pero en este caso creí que sería más óptimo utilizar los mapas. Y en efecto, no se puede demostrar empíricamente, pero creo que ha sido un acierto utilizar esta herramienta. Para tener una idea de la complicación, para mapear la información han hecho falta 16 mapas, que podemos observar en la figura x:

```
mapEscenasPorJuego: Map<number, Map<number, EscenaDeJuego>> = new Map<number, Map<number, EscenaDeJuego>>();
mapEscenarioPorEscena: Map<number, EscenarioEscapeRoom> = new Map<number, EscenarioEscapeRoom>();
mapObjetosPorEscena: Map<number, Map<number, ObjetoJuego>> = new Map<number, Map<number, ObjetoJuego>>();
mapInformacionGlobalDelObjetoJuego: Map<number, ObjetoGlobalEscape> = new Map<number, ObjetoGlobalEscape>();
mapEscenas: Map<number, EscenaDeJuego> = new Map<number, EscenaDeJuego>();
mapObjetosJuego: Map<number, ObjetoJuego> = new Map<number, ObjetoJuego>();
mapObjetosJuegoAuxiliar: Map<number, ObjetoJuego> = new Map<number, ObjetoJuego>();
mapObjetosEscapeFromObjetosJuego: Map<number, Map<number, ObjetoEscape>> = new Map<number, Map<number, ObjetoEscape>>(); /
mapObjetosEnigmaFromObjetosJuego: Map<number, Map<number, ObjetoEnigma>> = new Map<number, Map<number, ObjetoEnigma>>(); /
mapObjetosEscapeAuxiliar: Map<number, ObjetoEscape> = new Map<number, ObjetoEscape>();
mapObjetosEnigmaAuxiliar: Map<number, ObjetoEnigma> = new Map<number, ObjetoEnigma>();
mapPosicionObjetosDeEscena: Map<number, any> = new Map<number, any>(); //escena actual
mapLlavePorEscena: Map<number, Llave> = new Map<number, Llave>();
mapPistaPorEscena: Map<number, Pista> = new Map<number, Pista>();
mapObjetosRequeridosPorEscena: Map<number, Map<number, ObjetoEscape>> = new Map<number, Map<number, ObjetoEscape>>();
mapPosicionObjetosDeTodasLasEscenas: Map<number, Map<number, any>> = new Map<number, Map<number, any>>(); //escena actual
```

Figura 8.16. *Mapas utilizados*

Las peticiones a la base de datos, se realizan desde el *calculos.service* y se guarda la información en el *session.service,* mientras el alumno está en la pantalla de entrada

(componente: juego escape room). Posteriormente, todo el desarrollo del juego se produce en el componente escenario. En este, para hacerse una idea, el ngOnInit() del componente son aproximadamente unas 400 líneas, únicamente para pasar la información de los mapas a los objetos necesarios para cargar el escenario.

Igual que el proceso de mapear la información ha sido de los más complicados, el proceso de mapear las clases locales a los modelos de datos para poder actualizar el estado del escape room, ha tenido cierta dificultad.

En el dashboard, el proceso complicado ha sido el de ir creando los modelos en su respectivo momento, ya que al haber identificadores de por medio, no ha sido tarea fácil.

Pero sin duda, lo más complicado a la hora de desarrollar cualquier tipo de juego, es el definir los modelos que se van a utilizar. Durante la primera etapa del proyecto, se realizaron los primeros pasos, poder crear un juego desde el dashboard, que se conectará con el móvil del estudiante a través de la base de datos, etc. Esa misma tarea que inicialmente, pudimos tardar varias semanas, ahora mismo tardaríamos días, pero de eso trata el crecimiento. Por lo que, cuando se definieron los modelos iniciales, a medida que íbamos aprendiendo más y conociendo mejor el proyecto, nos íbamos dando cuenta de más errores, llegando al punto que, aproximadamente en el mes de junio, se hizo un cambio entero de los modelos, se definieron las escenas, los objetos juego, etc. Una vez bien definidos los modelos, y llevando las herramientas por la mano, la dificultad es mucho menor, por eso en mi opinión, después de haber realizado este proyecto, es que a la hora de desarrollar un juego, o cualquier otro tipo de desarrollo, se han de tener las bases lo más claras posibles, en este caso, los modelos de datos.

8.5 Implementaciones futuras

En este apartado, se van a describir una serie de opciones que en un futuro se podrían integrar en el juego.

8.5.1 Reloj

Buscando información sobre el mundo del escape room, hemos visto que una de las opciones que se utiliza es la del reloj. Dependiendo del tipo de historia, añaden un factor de tensión introduciendo un tiempo limitado de salida.
Esta idea sería interesante poder incorporarla al escape room, si bien no como factor obligatorio, se podría incluir una opción en el dashboard para que el profesor pudiera configurar que el alumno supiera cuánto tiempo le queda.

8.5.2 Grupo

El juego actualmente está diseñado en modo individual, ya que como había que centrarse en desarrollar la mecánica del juego y los modelos que se iban a utilizar, llegamos a la conclusión de que iba a ser un factor que le añadiría demasiada dificultad inicial, pero en general, tanto la mecánica como el diseño de las clases, están pensados para que en un futuro se incorpore el modo grupo.

8.5.3 Chat

En caso de que en un futuro se decidiese incorporar el modo grupal, sería interesante incorporar la opción de tener un chat con los diferentes miembros del juego, ya que esto podría incrementar exponencialmente la jugabilidad. Incluso, se podrían llegar a introducir mecanismos en los que se necesitará una comunicación entre integrantes para superar los.

8.5.4 Recursos públicos

Actualmente, el profesor únicamente puede crear y ver, tanto sus escenarios de escape como sus objetos, y por falta de tiempo, no se ha podido implementar la funcionalidad de hacer dichos recursos públicos. Pero, esta no debería ser una tarea complicada, ya que los modelos están creados con los atributos del profesor que los ha creado, y se puede seguir la metodología que se ha implementado en otros recursos que sí que tienen esa funcionalidad.

8.5.5 Interactividad entre profesor y alumno

Otra de las opciones que serían interesantes, es ofrecer algún tipo de mecanismo en tiempo real, permitiendo al profesor ver en cada momento la situación del alumno, poder introducir un chat entre el alumno y él, por tema de pistas por ejemplo, y ver si de alguna manera se pudiese crear un juego que fuera cambiante, bidireccional y se fuera adaptando a medida que avanza la historia.

Capítulo 9: Plan de pruebas

En este capítulo se definen las bases del plan de pruebas llevado a cabo para comprobar el correcto funcionamiento de todas las integraciones realizadas, además de ser la propuesta estándar de éste para futuros proyectos.

La herramienta utilizada para el plan de pruebas es Microsoft Excel. La estructura es la siguiente:

- ***Primera pestaña de introducción.*** Explicamos los motivos por los que se ha creado el documento, la metodología que se ha de utilizar a la hora de crear una prueba y la información necesaria sobre los documentos auxiliares.
- ***Herramientas de Classpip.*** Se han creado cuatro pestañas en las que vamos a clasificar las pruebas realizadas para este proyecto: DASHBOARD, MOVIL ALUMNO, EXPRESS y WEB.

El objetivo, ya mencionado anteriormente, es comprobar el correcto funcionamiento de las diferentes herramientas utilizadas que forman el ecosistema de Classpip y el de generar una propuesta de un estándar para un plan de pruebas que no sirva solo para este proyecto, sino para cualquier otro tipo de trabajo que se realice, de manera que todas las modificaciones que se vayan añadiendo a lo largo de Classpip puedan ser probadas, y resueltas en caso de error, de una forma sencilla y eficaz.

Este capítulo se ha realizado conjuntamente con los miembros del proyecto pero diferenciando cada apartado en sí para cada uno de nosotros. Por lo tanto, a pesar de que ha sido una propuesta grupal cada uno ha tenido que realizar varias pruebas en un apartado distinto. En mi caso, me he centrado en las pruebas relacionadas con el Escape room.

9.1 Formato

A continuación, se detallan el formato de las pruebas, pero primero mencionar que en este apartado no se realiza una guía de cómo crear una prueba con un ejemplo incluido, esta guía se encuentra en el anexo 2.

Se ha diseñado un estándar que es reutilizable para cualquier tipo de prueba , tanto para este proyecto como cualquier otro tipo de integraciones que se puedan añadir al entorno de Classpip. Para mostrar dicho estándar a continuación se muestra, a modo de ejemplo, una prueba realizada en el Dashboard, en concreto, el registro de un usuario.

	Cód. Caso de Prueba	Entidad	Título
AUTENTIFICACIÓN	DASH-001	Usuario / Profesor	Registro

Figura 9.1 ***Pestaña dashboard en el excel de pruebas I***

- El elemento que vemos en la primera columna es la ***temática*** de la prueba. En la captura podemos observar que esta prueba está relacionada con la autentificación en el dashboard.
- El siguiente elemento es el ***Código*** que se le asigna a la prueba. Está diseñado de esta manera con un propósito, y es que se pueda identificar fácilmente y se pueda referenciar en otras pruebas.
- El tercer elemento es ***Entidad*** y como el nombre indica, hace referencia a la entidad sobre la que se hace la prueba, que en este caso es el Profesor.
- Además de un código de prueba, hemos añadido un elemento para esclarecer qué prueba es y ese elemento es el ***título***.

Requisitos para la prueba	Descripción de la prueba	E
Ninguno	Accede a la página inicial del DashBoard y pincha en el botón de registro. Rellena el formulario con los datos especificados y dale a "Registrar".	p

Figura 9.2 ***Pestaña dashboard en el excel de pruebas II***

- ***Requisitos para la prueba.*** Dependiendo del tipo de prueba, se van a necesitar unos requisitos u otros. En este elemento, se utiliza de manera muy eficiente el código que hemos mencionado con anterioridad, ya que depende del tipo de prueba, puede que sea necesario realizar alguna comprobación previa.
- ***Descripción de la prueba.*** En este apartado se detalla paso por paso como se ha de proceder para realizar dicha prueba. En las pruebas en las que sea necesaria una explicación más detallada, se permite la opción de adjuntar cualquier tipo de documento explicativo, como puede ser un documento word o un documento excel.

Resultado esperado	Creado por
El usuario y el profesor se crean correctamente y el usuario accede a la página principal del DashBoard, en la que se muestra los "Juegos Activos" (en este caso, como nos acabamos de registrar, no aparecerá ningún juego). Para corrobar que ha ido bien, comprobar que en "Session Storage" se ha guardado el **ACCESS_TOKEN.** A partir de ahora, este Token se incluirá en la cabecera de todas las peticiones para tener acceso y autorización a los Endpoints de la API.	Iván

Figura 9.3 *Pestaña dashboard en el excel de pruebas III*

- Después de describir los requisitos y los pasos a proseguir para realizar la prueba, se detalla previamente a la realización de la prueba, ***el resultado esperado*** si se realiza correctamente.
- Posteriormente se deja documentado quién ha sido el ***creador*** de la prueba.

Fecha de Realización	Realizado por	OK	KO
25/06/2021	Jordi Salvador	X	

Figura 9.4 *Pestaña dashboard en el excel de pruebas IV*

- ***Fecha de Realización.*** Como indica el nombre, es la fecha en la que se realiza la prueba, **NO** es la fecha en la que se crea, que son conceptos diferentes.
- ***Realizado por.*** La persona que crea la prueba no tiene por que ser la misma persona que la ejecute, por lo que hemos querido diferenciar en dos apartados, uno para el creador y otro para el que ejecuta dicha prueba.
- Los dos siguientes apartados son para documentar el resultado de la prueba, en caso positivo se pone una cruz en el OK y en caso negativo se pone una X en el KO.

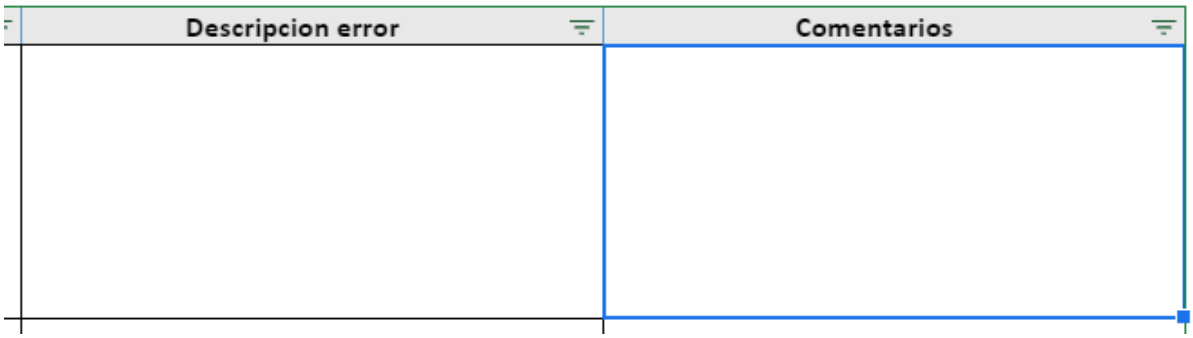


Figura 9.5 ***Pestaña dashboard en el excel de pruebas V***

- Por último, tenemos dos apartados: En el caso de que el resultado de la prueba haya sido fallido, tenemos la ***Descripción del error,*** dónde cada usuario que realice la prueba puede documentar los motivos por los que no ha salido correctamente la prueba y los apartados que se han de solucionar. En caso de que haya sido positiva la prueba, si queremos dejar algún tipo de comentario lo podemos hacer en el apartado de ***Comentarios.***

Este formato no va ligado a ningún tipo de prueba en concreto, permitiendo al usuario que realice el plan de pruebas realizar cualquier tipo de prueba de una forma sencilla y eficaz, no solo dentro del ecosistema Classpip, sino en cualquier otro proyecto en el que sea necesario realizar pruebas de funcionamiento.
Ese era uno de los objetivos planteados al iniciar el diseño del documento, y es que si nos fijamos en el hilo de nuestras memorias, al final uno de los objetivos que siempre hemos tenido presente es estandarizar y uniformizar lo máximo posible el proyecto. Creemos que es el método de trabajo más óptimo en cualquier proyecto, y en este más si cabe, ya que es un trabajo por el que pasan diferentes estudiantes cada año.

Este documento está creado para reportar cualquier tipo de error en las herramientas, por lo que está abierto a todos participantes que quieran realizar cualquier tipo de comprobación, únicamente tendrán que seguir los pasos que hemos detallado en la guía.

9.2 Pruebas en el Dashboard

Como hemos mencionado en la introducción del capítulo, estas pruebas fueron diseñadas de manera conjunta, pero al haber tantos proyectos en las que realizarlas decidimos dividirlas entre los 4. En mi caso realice todas las pruebas referentes al escape room.

Las pruebas realizadas sobre el escape room, son pruebas en las que he tenido que desarrollar un documento aparte para describir el proceso, sobre todo en la creación del juego. En el momento de realizar las pruebas que necesitaba más de un integrante, ya había hecho diferentes pruebas para estar lo más seguro posible de que el código fuera robusto, por lo que cuando hicimos las pruebas, no tuvimos ningún error aparente, más que cambiar detalles visuales.

Tanto mis compañeros como yo, realizamos muchas pruebas para dejar el proyecto lo más funcional posible, pero es cierto que no realizamos pruebas todo el código, así que, para futuros integrantes de Classpip les recomendaría que realizarán las máxima pruebas posibles para comprobar el funcionamiento antes de hacer nuevas funcionalidades, y una vez realizadas sus aportaciones al proyecto deberían realizar todas las pruebas que crean oportunas para seguir con el criterio que hemos establecido.

Capítulo 10. Conclusiones

En este último apartado, se realizará una introspección y valoración del proyecto, teniendo en cuenta los objetivos iniciales que nos propusimos, el camino que hemos recorrido y el resultado final alcanzado.

10.1 Objetivos

Como bien hemos visto durante la memoria, desarrollamos tanto como grupo como individualmente una serie de objetivos a cumplir. En mi caso, se me otorgaron dos de ellos: desarrollar un juego de escape room para Classpip y los diferentes arreglos en las herramientas del proyecto.

En mi opinión, es que es difícil valorar si uno ha cumplido sus objetivos, ya que, en tareas de este estilo, siempre existe una posible mejora, siempre se podría llegar a desarrollar de manera más eficiente o simplemente haber implementado más funciones al proyecto. Pero, el resultado final ha sido bastante satisfactorio, hemos dejado el proyecto con una web funcional, que se han encargado mis compañeros Jordi e Ivan, la opción de poder traducir la herramienta del dashboard, que se ha encargado mi compañero Carlos, un juego de *Escape Room* operativo, del cual me he encargado yo, y un plan de pruebas implantado para futuros estudiantes.

En cuanto a la calidad del contenido, solo puedo hablar sobre la implicación y dedicación que hemos tenido todos los miembros de este proyecto desde su inicio en febrero, ya que esta ha sido elevada. Creemos que hemos cumplido con lo que se nos propuso.

Por otro lado, desde el primer momento, decidimos que queríamos desarrollar un método de trabajo que fuera tanto eficiente, como cómodo para trabajar en el proyecto más largo e importante de nuestra carrera, y desde mi punto de vista, lo hemos conseguido con éxito. Hemos conseguido desarrollar una metodología de trabajo acorde a las necesidades de todos los miembros del grupo y, gracias a eso, ha resultado mucho más sencillo de lo que en sus inicios podía parecer un trabajo grupal. Nos hemos apoyado unos a otros y hecho equipo.
Además, hemos dejado documentado todo el proceso que llevábamos a cabo para organizarnos y creemos que puede llegar a ser muy útil para futuros alumnos que quieran dejar su huella en este proyecto.

10.2 Valoración personal del proyecto

Considero importante hablar sobre mi experiencia y objetivos personales a la hora de realizar este trabajo de fin de grado. Actualmente, estoy cursando una doble titulación de Telecomunicaciones y Telemática, y como consecuencia, podría haber escogido un proyecto que englobase ambas carreras, pero este proyecto llamó mucho mi atención.

Uno de los motivos principales de mi elección, fueron los dos profesores encargados del proyecto. Ambos son profesores de aquella parte a la que me gustaría dedicar, la programación y me han marcado para bien en el ámbito estudiantil. Tanto Miguel Valero en Sistemas Operativos como Roc en Proyecto de programación habían conseguido que me interesara por el mundo de la programación y a nivel personal, los consideraba unas personas dedicadas y dispuestas a ayudarnos en aquello que estuviera en sus manos.

Evidentemente, este no fue el único factor por el que decidí hacer este trabajo de final de grado, como he mencionado, la programación ha sido uno de los aspectos más importantes de la carrera. Entré al grado sin saber absolutamente nada de programar, y se ha convertido en uno de los aspectos en los que quiero enfocar mi carrera profesional. Por ese motivo, este trabajo era una nueva oportunidad para aprender, utilizar nuevas herramientas y tener un primer acercamiento a lo que podría ser un proyecto más profesional, en el que hemos tenido que trabajar en grupo, realizando control de versiones, reuniones con los profesores, etc. Personalmente, considero que tanto mis compañeros como yo, hemos salido de este proceso siendo más profesionales de como entramos y más habituados, si cabe, al trabajo en grupo. El trabajar con compañeros cercanos a mí, tanto en la carrera como en mi vida personal, también fue un gran impulso a la hora de decidirme.

Por último, uno de los motivos que me hacía especial ilusión, era que iba a trabajar en un proyecto en el que iba a poder aportar mi granito de arena, es decir, durante la carrera, la gran mayoría de los trabajos, únicamente tienen el objetivo de aplicar conocimientos aprendidos, pero esos proyectos, a posteriori, se pierden en el olvido. Sin embargo, este está siendo utilizado en universidades, y el poder participar en él y conseguir darle más visibilidad, me llena de satisfacción.

Con estos comentarios, parece que no han surgido dificultades durante el proceso, pero también hemos tenido complicaciones. La primera etapa, fue bastante dura, ya que el proyecto estaba muy desarrollado, y el llegar a hacerlo nuestro, no fue tarea fácil. Para futuros participantes del proyecto, recomiendo encarecidamente utilizar todas las herramientas que nos proporcionan tanto Miguel como Roc para familiarizarnos con el código, tanto los tutoriales como las tareas iniciales, ya que a nosotros es lo que nos ha hecho aprender y tener una buena base para cumplir nuestros objetivos.

Como conclusión en la valoración, no me arrepiento en absoluto de haber escogido este proyecto, el trato con mis compañeros y con los profesores ha sido ideal y he tenido un gran aprendizaje tanto a nivel personal como a nivel profesional.

10.3 Mejoras a considerar

En este último apartado, hablaré de los aspectos que me hubiera gustado realizar en las tareas que he desarrollado, en caso de que se hubiera prolongado la duración del proyecto.

En el diseño del dashboard, se podría configurar para que el profesor pudiese escoger entre distintas temáticas y no únicamente la de estaciones.

En cuanto a las directrices que hemos marcado en el diseño del estándar, se han aplicado en la herramienta dashboard, móvil de estudiante, web y classpip express, nos ha quedado pendiente el móvil del profesor.

Por último, en el apartado *8.5 Implementaciones futuras,* se describen una serie de implementaciones que nos hubiera gustado poder desarrollar, tales como el escape room para grupos, un chat, interactividad entre el profesor y el alumno, etc. Este tipo de juegos la ventaja que tienen es que son muy creativos, por lo que con más tiempo, se nos hubieran ocurrido muchas más funcionalidades que podríamos desarrollar.

Como he comentado con anterioridad, todo y que nos hubiera gustado haber podido implementar las funciones que nos han faltado por desarrollar, estamos más que satisfechos con el trabajo realizado.

Bibliografía

[1] Educación 3.0 - *¿Qué es la gamificación y cuáles son sus objetivos?*: https://www.educaciontrespuntocero.com/noticias/gamificacion-que-es-objetivos/

[2] Miguel Valero García, Roc Meseguer Pallarés - *Repositorios de Classpip en Github*: https://github.com/classpip

[3] Proyectos ágiles - *Metodología Scrum*: https://proyectosagiles.org/que-es-scrum/

[4] Scrum Org - *What is a Scrum Master?*: https://www.scrum.org/resources/what-is-a-scrum-master

[5] Trello: https://trello.com/es

[6] Miguel Valero García - *Tutorial para herramientas de Classpip*: https://www.youtube.com/playlist?list=PL64O0POFYjHqXWZgAtku2zgG-wEFxJ3xM

[7] LoopBack - *Using built in models*: https://loopback.io/doc/en/lb3/Using-built-in-models.html

[8] Escape room - *¿Qué es y en qué consiste este juego en equipo?:* https://escapecollegemadrid.com/escape-room-que-es/

[9] Estilos de código - *¡Se ACABARON las discusiones de ESTILO en el CÓDIGO!* https://www.youtube.com/watch?v=PG4q3zEkzw0

[10] Diseño de los mapas - *Google Slides Bitmoji Escape Room Tutorial* https://www.youtube.com/watch?v=jjKkmRdQ8ac

[11] Escape room - *Buscador y Guía de Escape Rooms* https://www.escaperoomlover.com/es

Anexos

ANEXO 1 - Documentos de los sprints

Número del Sprint: 1

Fecha: 10/02 - 26/02

Descripción: Para la realización de este sprint se nos propuso un par de retos y dos proyectos distintos:

1. *Retos:* Primero de todo nos pusimos a trabajar en los retos planteados, que en este caso eran dos, separar los juegos rápidos del móvil del alumno en una nueva aplicación y la organización de estilos, tanto diseño como poder traducir la página web al inglés.
2. *Proyectos:* Se nos planteó la idea de hacer dos proyectos distintos, unos consistían hacer una página web y el otro en la realización de un Scape Room. Hemos desarrollado una presentación donde explicaremos las ideas que se nos han ocurrido para poder empezar a programar y cómo se nos han ocurrido.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto durante estas dos semanas ya sea vía Whatsapp, Google Meet o presencialmente, lo cual nos ha facilitado mucho la realización de este sprint. Siempre hemos podido ver los resultados del trabajo realizado por nuestros compañeros y hemos tenido una muy buena comunicación entre nosotros.

¿Qué cosas han salido mal respecto al grupo? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

Nada ha salido mal.

Número del Sprint: 2

Fecha: 26/02 - 19/03

Descripción: En este sprint, hemos empezado con el desarrollo de los retos y proyectos del anterior sprint:

1. *Retos:* Hemos avanzado en la separación de los juegos rápidos. En el móvil del estudiante hemos eliminado los componentes de este tipo de juegos, y en la aplicación del juego rápido hemos eliminado las dependencias innecesarias. Por otro lado, hemos investigado acerca de las herramientas de organización de estilos, hemos seguido retocando los estilos globales y hemos continuado ampliando el desarrollo para traducir el dashboard al inglés.
2. *Proyectos:* Hemos iniciado el desarrollo de la página web utilizando una plantilla. El navbar y el footer ya se han diseñado (junto a todas las páginas que tendrá la web) y hemos hecho el login y el register. Respecto al escape room, hemos estado planificando qué elementos se podrían incluir en el juego.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto durante estas tres semanas vía Whatsapp y Google Meet. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando ideas y consejos para los diferentes proyectos que tenemos que realizar.

¿Qué cosas han salido mal respecto al grupo? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

Nada ha salido mal.

Número del Sprint: 3

Fecha: 19/03 - 08/04

Descripción: En este sprint, hemos continuado con el desarrollo de los retos y proyectos del anterior sprint:

1. *Retos:* Hemos empezado a investigar acerca de las herramientas para uniformizar estilos en el código, aprovechando que el proyecto de juego rápido está finalizado y no es muy extenso. Respecto a los estilos gráficos, están hechos los estilos de las estaciones, pero falta acabar de perfeccionarlos.

2. *Proyectos:* Hemos añadido nuevas funcionalidades en la página web. Actualmente, el login y el registro ya funcionan, quedan a la espera de añadir el token una vez hayamos configurado la API. Además, hemos empezado a incorporar los recursos públicos y hemos añadido la página de perfil del usuario. Por otro lado, se ha empezado con el desarrollo del escape room, creando un nuevo proyecto, instalando las dependencias necesarias y empezando a aprender a utilizar elementos drag&drop.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto en todo momento, hablando y planificando las tareas. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando soporte a los compañeros cuando lo han necesitado.

¿Qué cosas han salido mal respecto al grupo? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

Durante este sprint no hemos podido dedicarle las horas que nos hubiese gustado, ya que cada uno ha tenido sus cosas y solo hemos podido ponernos en semana santa cuando no teníamos prácticas.

Número del Sprint: 4

Fecha: 8/04 - 22/04

Descripción: En este sprint, hemos continuado con el desarrollo de los retos y proyectos del anterior sprint:

1. *Retos:* Hemos iniciado el proceso de uniformizar estilos en el código, aprovechando que el proyecto de juego rápido está finalizado y no es muy extenso. Respecto a los estilos gráficos, hemos ido retocando fallos que íbamos viendo tanto en el dashboard como en el móvil del estudiante.

2. *Proyectos:* Hemos añadido una primera interfaz gráfica donde aparecen los recursos en la página web. Actualmente, el login y el registro ya funcionan, quedan a la espera de añadir el token una vez hayamos configurado la API. En cuanto al escape room, hemos avanzado con la idea de que el profesor pueda modificar el escape room desde el dashboard.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto en todo momento, hablando y planificando las tareas. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando soporte a los compañeros cuando lo han necesitado.

¿Qué cosas han salido mal? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

Hemos tenido problemas de instalación con las diferentes herramientas, sobre todo con el *EsLint* y el *Prettier* pero hemos conseguido solucionarlo. Otro de los problemas que han surgido es con la API.

Número del Sprint: 5

Fecha: 22/04 - 13/05

Descripción: En este sprint, hemos implementado la autentificación con el modelo User de loopback y protegido las primeras rutas con los access tokens. También hemos avanzado en la implementación de los recursos públicos en la web. En cuanto al escape room, hemos acabado de desarrollar una versión inicial de configuración en el dashboard y hemos unificado el proyecto externo que teníamos de escape room con el móvil del estudiante para tener una versión funcional del escape room.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto en todo momento, hablando y planificando las tareas. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando soporte a los compañeros cuando lo han necesitado.

¿Qué cosas han salido mal? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

No ha salido nada mal.

Número del Sprint: 6

Fecha: 13/05 - 03/06

Descripción:

En este sprint hemos continuado con el desarrollo de varias funciones de la web:

- Hemos desarrollado la API propia de la web, totalmente funcional y sin fallos. Además, hemos mantenido los modelos originales para garantizar compatibilidad y uniformidad con los otros proyectos aparte de la web.

- Hemos avanzado en la creación de la página de experiencias

- Hemos avanzado en el desarrollo de la página de recursos

También hemos continuado con el desarrollo del juego de Escape Room:

- Se han diseñado dos modelos de objeto nuevos

    - **ObjetoEscape:**
        - Nombre
        - Usable
        - Recogido
        - Posición

    - **ObjetoEnigma**:
        - Nombre
        - Pregunta
        - Respuesta
        - Resuelto

- Se ha creado la mochila con la respectiva función para poder guardar objetos en ella.
- Se ha creado una primera versión de los recursos del escape room en el dashboard, es decir, los escenarios y los objetos.

Además, hemos continuado con la mejora de la calidad del código, cambiando el estándar de los modelos de toda la API, el dashboard y el móvil del estudiante. En el próximo sprint, se cambiará también ClasspipExpress.

También hemos iniciado el cambio de cordova por capacitor y hemos generado el ejecutable para poder ver los console.log en el móvil, pero nos ha dado problemas.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto en todo momento, hablando y planificando las tareas. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando soporte a los compañeros cuando lo han necesitado.

¿Qué cosas han salido mal? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

No ha salido nada mal.

Número del Sprint: 7

Fecha: 03/06 - 17/06

Descripción:

En este sprint hemos continuado con el desarrollo de varias funciones de la web:

- Hemos finalizado la implementación para poder subir recursos, tanto rellenando el formulario como subiendo un json. También hemos añadido recomendaciones en los recursos.
- También hemos añadido la opción de descargar varias preguntas a la vez, generando un solo json con todos los datos, o un json por pregunta.
- Hemos añadido también las recomendaciones en todos los tipos de recurso.
- Hemos implementado la función para subir ficheros e imágenes junto con las experiencias, y que se puedan descargar los ficheros.
- Hemos añadido la opción de editar los datos del profesor logueado
- Hemos añadido la opción de cambiar la contraseña.
- Además, hemos arreglado varios errores que nos hemos ido encontrando:
    - Cuando se borraba un recurso, no se borraban las imágenes, ahora si.
    - Cuando se descargaba un recurso, se descargaba el fichero json con los campos auxiliares que añadimos, ahora ya no.

También hemos continuado con el desarrollo del juego de Escape Room:

- Se han casi finalizado los recursos del Escape Room en el dashboard
- Se ha empezado el diseño de los nuevos escenarios con la lógica del juego

También hemos avanzado con las traducciones del Dashboard y con la memoria.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto en todo momento, hablando y planificando las tareas. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando soporte a los compañeros cuando lo han necesitado.

¿Qué cosas han salido mal? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

No ha salido nada mal.

Número del Sprint: 8

Fecha: 17/06 - 28/06

Descripción:

En este sprint hemos mejorado e implementado los mecanismos de autentificación en la página web, en el dashboard y en el móvil del estudiante:

- Hasta ahora, sólo habíamos implementado autentificación en la web, debido a que quedaba pendiente el cambio de variables en las distintas aplicaciones para uniformizar estilos. Para ello, investigamos acerca de los mecanismos que ofrecía Loopback e hicimos un diseño inicial. Lo implementamos para la web y la tarea quedó parada.

- A la hora de retomar la tarea, hemos visto un mecanismo mejor para implementarlo que nos facilitaba mucho la faena para implementarlos luego en las demás aplicaciones. Por lo tanto, se han cambiado los modelos y reprogramado las funciones de la web.

- Una vez comprobado en la web que iba, lo pasamos a las demás aplicaciones. Actualmente web y dashboard están funcionales al 100%, móvil del estudiante falta revisar modificación perfil.

- Investigamos sobre cómo implementar mecanismos para recuperar la contraseña enviando un token temporal a través de correo electrónico.

También hemos replanteado la lógica del juego de Escape Room:

- Se han rediseñado los modelos de las clases del juego, implementando nuevas relaciones y una nueva clase auxiliar para el juego llamada Escena, dónde se va recopilando la información de los diferentes escenarios

- Al replantear los modelos, se ha tenido que retocar la lógica. A partir de ahora, un juego de Escape Room puede tener tantas escenas como se quieran (las cuáles pueden ser públicas y reutilizarse en otros juegos) y el profesor, a la hora de configurar el juego, configura las diferentes escenas con los objetos que aparecerán en ella.

- Se han implementado las funciones para poder subir imágenes de escenarios y objetos a la API y tener esas clases como recursos.

Respecto a la web, hemos terminado las integraciones, hemos redactado y ejecutado el plan de pruebas y corregido pequeños errores que había.

Por otro lado, hemos continuado con:

- Migración de Córdova a capacitor
- Traducción de la página web (comentar con roc y miguel)

Por último, hemos empezado a preparar el documento del plan de pruebas y a estructurar y desarrollar la memoria.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto en todo momento, hablando y planificando las tareas. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando soporte a los compañeros cuando lo han necesitado.

¿Qué cosas han salido mal? En caso de haber alguna explica que vais a mejorar de cara al siguiente sprint.

En este Sprint se ha visto afectado el ritmo de desarrollos que llevábamos debido a que hemos tenido que rediseñar algunos modelos y reprogramar la lógica ya desarrollada para adaptarla a los nuevos cambios, además de que nos hemos encontrado varios errores que nos han impedido seguir desarrollando.

Respecto al punto de recuperar contraseña de autentificación, hemos estado investigando acerca de cómo incluir un certificado en Loopback para enviar correos electrónicos, pero no hemos encontrado ningún ejemplo ni documentación que nos ayudará. Por lo tanto, teniendo en cuenta que la intención es hacer un último sprint, y que esta tarea nos puede acarrear cierta dificultad, cosa que afectaría a la finalización de los demás desarrollos, hemos decidido dejarla de lado para próximas mejoras.

En la traducción no se ha podido realizar correctamente ya que hay una incompatibilidad de versiones y la migración de Córdova a capacitor en la parte del móvil del profesor por el componente bundle no se ha podido realizar.

Número del Sprint: 9

Fecha: 28/06 - 11/07

Descripción:

En este sprint hemos finalizado el desarrollo de la web:

- Se han acabado de diseñar las diferentes páginas de la web que no requerían de integraciones.

- Se han modificado los recursos para garantizar uniformidad con el Dashboard:
    - Se pueden descargar recursos de la web y subir al dashboard y viceversa
    - Se especifican los mismos tamaños de las imágenes que el Dashboard.

- Se ha vuelto a pasar el plan de pruebas para corroborar que todo funciona perfectamente y arreglar los errores que pudiesen salir

- En el móvil del estudiante: Se han implementado funcionalidades nuevas, se ha mejorado el código, se ha creado el juego Escape Room y se han añadido nuevos mapas para una versión operativa.

- Creación de una guía de Loopback y otra de Escape Room. También hemos creado el documento de pruebas relacionadas con el escape room.

- En el dashboard hemos configurado la creación del juego con las diferentes escenas. También hemos modificado el tema de las imágenes, hasta ahora las cogíamos desde la carpeta assets y ahora están en la base de datos.

- Exportar las aplicaciones de Android y IOs

- Configuración de reglas de estilo con Prettier y Eslint.

- Actualización de versión de angular (Comentar con Roc y Miguel)

Por último, hemos continuado con la redacción de la memoria y hemos añadido nuevas pruebas en el plan de las diferentes aplicaciones. Para realizar las pruebas, hemos tenido que ir comprobando diferentes trozos del código de todas las aplicaciones para arreglar errores de cambios de variable que se hicieron a la hora de uniformizar los estilos y definir un estándar.

¿Qué cosas han salido bien respecto al grupo?

Hemos estado en contacto en todo momento, hablando y planificando las tareas. Nos hemos ido notificando de todos los avances y cambios, y hemos colaborado dando soporte a los compañeros cuando lo han necesitado.

¿Qué cosas han salido mal?

Al ser el último Sprint, hemos tenido que hacer varias reuniones para concretar qué es lo que nos iba a dar tiempo de acabar de desarrollar y qué dejábamos como posibles mejoras para siguientes aportaciones. Además, realizar el plan de pruebas nos está costando, ya que hay algunas funcionalidades que no sabemos cómo van porque no hemos llegado a utilizarlas, y tenemos que ir revisando errores de cambio de variables en el código, cosa que ralentiza la elaboración y ejecución de las pruebas.

ANEXO 2 - Guía para crear una prueba

Este documento contiene los pasos a seguir para crear una prueba en el Excel de pruebas siguiendo las directrices impuestas.

- ***Paso 1.*** Descargar el documento Excel de pruebas.

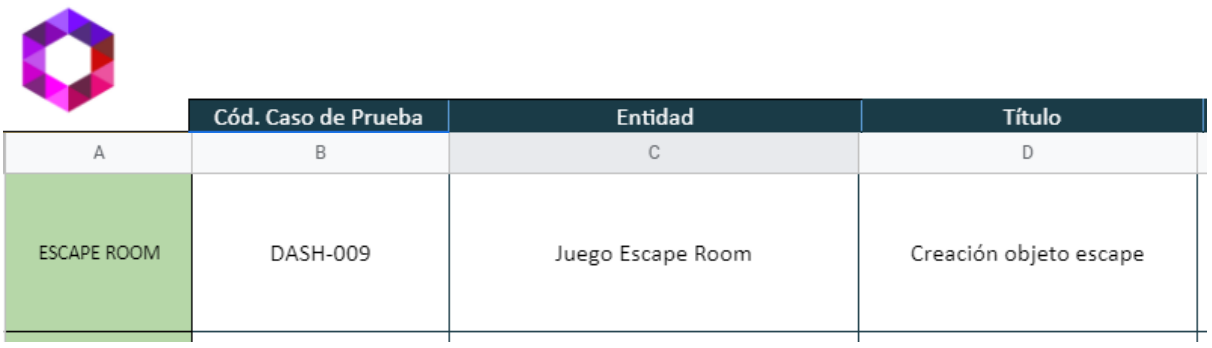
A continuación, vamos a realizar la introducción de una prueba ejemplo para explicar los pasos a seguir. Mencionar que los pasos son idénticos en todas las herramientas.

La prueba que vamos a realizar va a ser la creación de un objeto del juego escape room en el dashboard.

- ***Paso 2.*** Abrimos la pestaña del ***DASHBOARD***, que será la herramienta en la que realizaremos la prueba.

- ***Paso 3. Configuración de la prueba:***

  - ***3.1.*** El primer paso será escribir en la primera columna la temática de la prueba. En nuestro ejemplo la temática será ***"ESCAPE ROOM".***
  - ***3.2.*** En el apartado de ***"Código caso de prueba"***, la metodología a seguir es la siguiente: SIGLAS DE LA HERRAMIENTA UTILIZADA - CÓDIGO. En nuestro caso como estamos utilizando el dashboard y es la novena prueba creada, el código será ***"DASH-009".***
  - ***3.3. Entidad de la prueba:*** En este apartado se hace referencia al objeto que en el que se realiza la prueba, por ejemplo, si la prueba es de autentificación en el dashboard, la entidad sería *Profesor.* En nuestro caso la entidad es ***"Juego de Escape Room".***
  - ***3.4 Título de la prueba: "Creación objeto escape".***



	Cód. Caso de Prueba	Entidad	Título
A	B	C	D
ESCAPE ROOM	DASH-009	Juego Escape Room	Creación objeto escape

Figura 2.1A *Prueba introducida en la pestaña del dashboard I*

  - ***3.5 Requisitos para la prueba:*** Para realizar esta prueba, los requisitos son ***estar logueado como Profesor en el Dashboard y tener descargado el documento de "PRUEBAS ESCAPE ROOM"*** que encontraremos en la siguiente dirección: En este apartado podemos referenciar códigos de otras pruebas en el caso de que hubieran sido necesarias para este apartado.
  - ***3.6 Descripción de la prueba:*** En este apartado describiremos la prueba a realizar, en nuestro caso como la descripción es más

detallada de lo habitual, se ha creado documentado la prueba en el documento que se ha mencionado con anterioridad. El aspecto será el siguiente:

Requisitos para la prueba	Descripción de la prueba
Sesión iniciada en DashBoard y tener descargado el documento "PRUEBAS ESCAPE ROOM"	Abrir el documento "PRUEBAS ESCAPE ROOM" y realizar los pasos especificados en el apartado "CREACIÓN OBJETO ESCAPE DESDE DEL DASHBOARD"

Figura 2.2A *Prueba introducida en la pestaña del dashboard II*

- ***3.7 Resultado esperado:*** En este apartado se describe el resultado esperado si la prueba se realiza correctamente. En nuestro ejemplo, el resultado sería que ***se cree correctamente el objeto y que nos aparezca en el apartado de "Mis objetos".***
- ***3.8 Creado por:*** En este apartado se identificará la persona que ha creado la prueba.
- ***3.9 Fecha de Realización:*** La fecha en la que se ejecuta la prueba.

Resultado esperado	Creado por	Fecha de Realización
El objeto se crea correctamente y aparece en el apartado Mis Objetos.	Mario	15/7/2021

Figura 2.3A *Prueba introducida en la pestaña del dashboard III*

- ***3.10 Realizado por:*** En este apartado se diferencia la persona que ha creado la prueba de la persona que la ha ejecutado. En nuestro ejemplo es la misma persona, pero podría ser perfectamente dos personas diferentes.
- ***3.11 Resultado de la prueba:*** En los siguientes dos apartados se documenta el resultado de la prueba, se pone una X en OK si ha salido correctamente o en el caso contrario, una X en el KO. En nuestro caso vamos a indicar que ha salido correctamente, por lo que ***marcamos una X en el OK.***
- ***3.12 Descripción del error / comentarios.*** En caso de que la prueba haya resultado negativa, se propone un apartado de descripción de error donde el usuario podrá documentar cuál ha sido el error y los motivos en caso de saberlos. Por otro lado, si la prueba ha salido correctamente se da la opción de dejar un comentario en caso de que fuera necesario. Para nuestro ejemplo, como ha salido correctamente y no se ha desviado de lo esperado, ***dejamos en blanco los dos campos.***

Realizado por	OK	KO	Descripcion error	Comentarios

Figura 2.4A *Prueba introducida en la pestaña del dashboard IV*

- Paso 4. En caso de que haya salido KO como resultado de la prueba, después de solucionar los errores se deberá volver a pasar la prueba y así sucesivamente hasta que salga OK.

Como hemos mencionado en el inicio del documento, los pasos detallados y las explicaciones son los mismos en todas las herramientas, ya que hemos uniformizado el formato.

ANEXO 3 - Guía para crear un juego de escape room

El documento contiene los pasos a seguir para crear un juego de escape room en el Dashboard.

Los requisitos previos para poder realizar la creación de un juego de escape room son los siguientes:

- Estar logeado con un usuario de Profesor.
- Tener un alumno que se haya registrado con el código del profesor.
- Haber creado los escenarios en los recursos para poder utilizarlos en el juego.
- Haber creado los objetos en los recursos para poder utilizarlos en el juego.

A continuación, vamos a crear un juego ejemplo y explicar paso por paso.

- **Paso 1: *Asignar un grupo a un alumno***. En nuestro caso será el alumno Ivan en el grupo 1.
- **Paso 2: *Crear el juego escape room.*** El primer paso es elegir el nombre del juego. En este ejemplo el nombre será "*El juego encantado*".

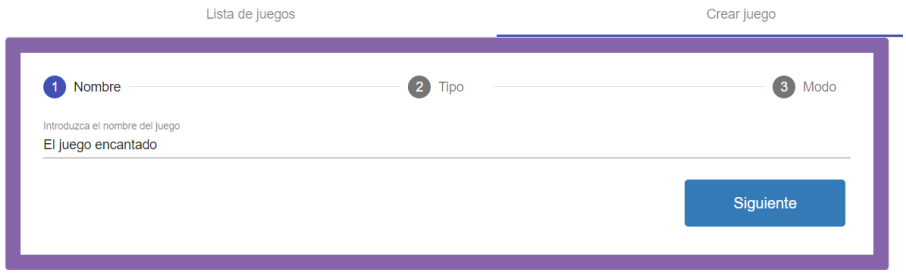


Figura 3.1A *Pantalla de configuración del juego I*

- **Paso 3: *Escoger el tipo de juego***. El tipo de juego será Juego de Escape Room.

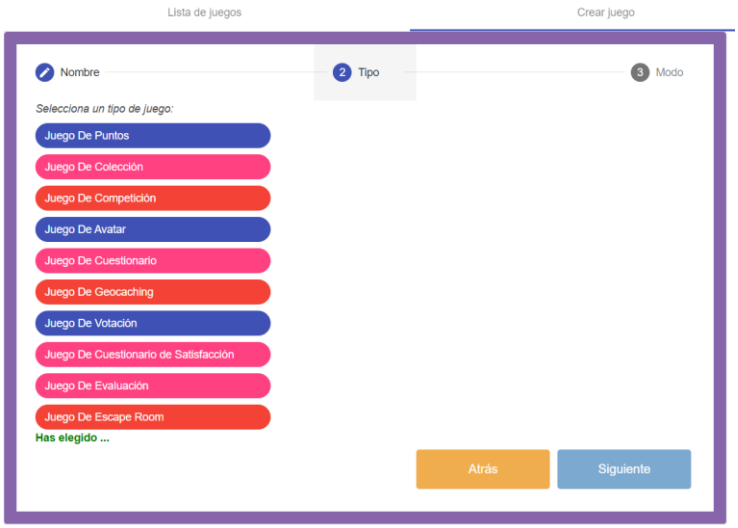


Figura 3.2A *Pantalla de configuración del juego II*

- **Paso 4: *Escoger el modo de juego.*** En nuestro caso individual.

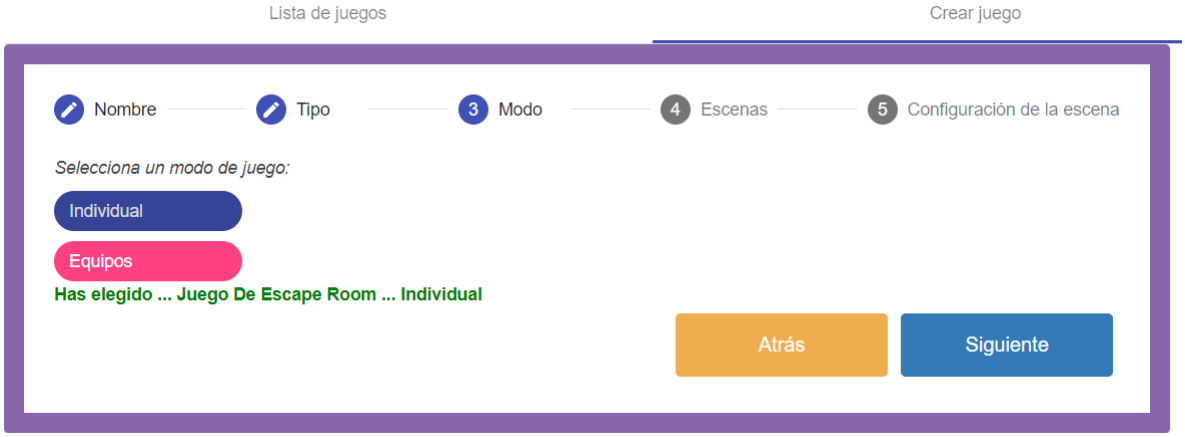


Figura 3.3A. *Pantalla de configuración del juego III*

- **Paso 5: *Escoger el número de escenas.*** Para la guía vamos a configurar un juego de 2 escenas, pero se podrían configurar las escenas que queramos, siempre y cuando tengamos objetos suficientes para configurar diferentes escenas sin repetir objetos.

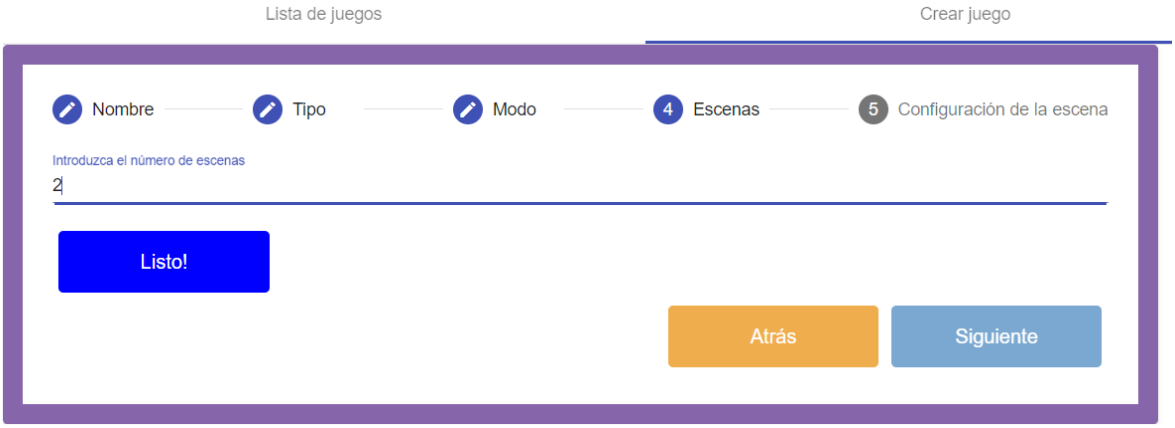


Figura 3.4A *Pantalla de configuración del escape room I*

- **Paso 6: *Menú de configuración.*** A continuación, vamos a mostrar el menú de configuración de una escena.

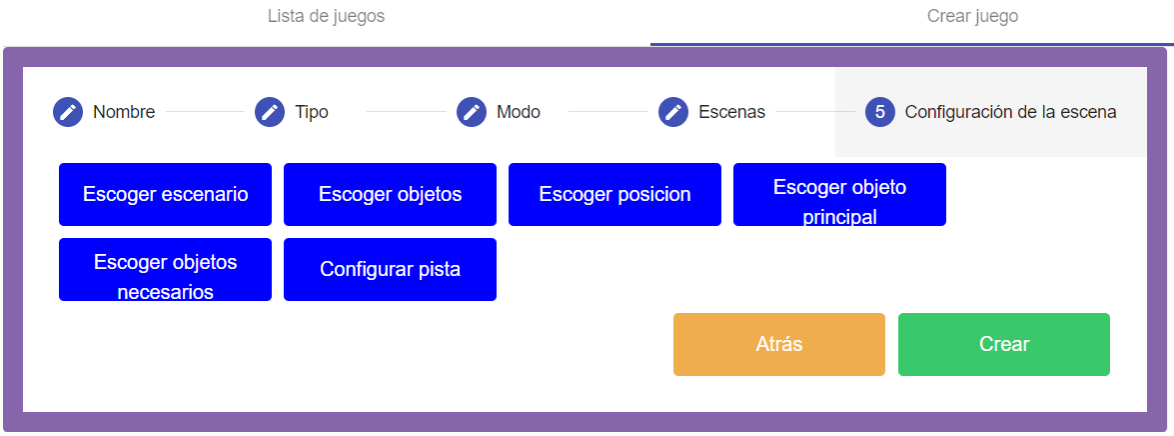


Figura 3.5A *Pantalla de configuración del escape room II*

- **Paso 7. *Escoger el escenario.*** El primer paso para configurar la escena es escoger el escenario. Para ello clicamos en el botón "*Escoger escenario*" y nos saldrá el siguiente Swal:

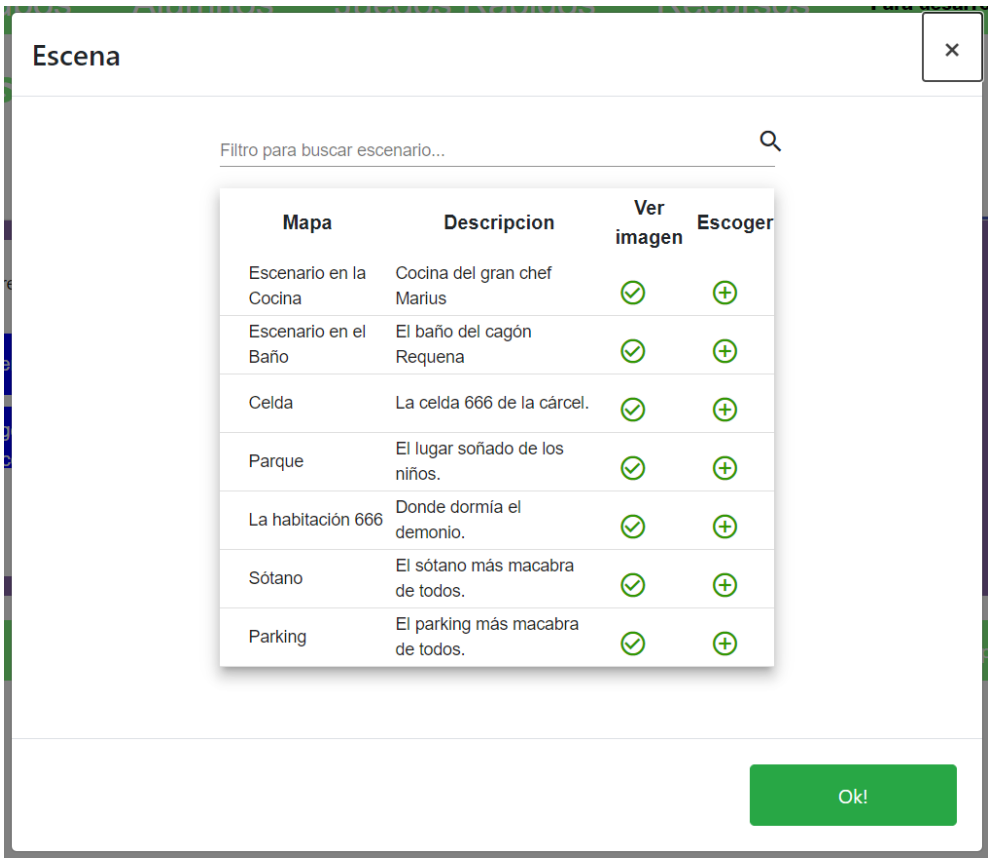


Figura 3.6A *Pantalla de escoger escenario*

Tenemos la opción de ver la imagen del escenario y escoger el escenario. Para nuestro ejemplo escogeremos el primero, "*Parque*". Primero clicamos en Ver imagen y posteriormente escogemos el escenario.

Figura 3.7A Swal alert para *ver escenario I*

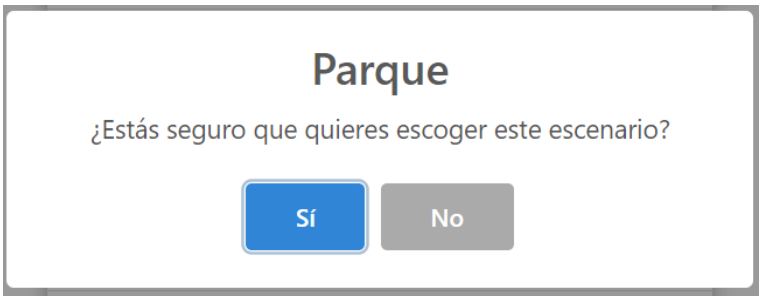


Figura 3.8A Swal alert para *ver escenario II*

- **Paso 8. *Escoger los objetos.*** El siguiente paso es escoger los objetos que vamos a utilizar para ese escenario. Tendremos que escoger tres objetos escape y dos objetos enigma de todos los objetos creados. Los objetos que escojamos para esta escena no se podrán escoger en la siguiente.

Figura 3.9A Swal alert para *ver escenario II*

En nuestro caso de objetos escape vamos a escoger: "*Pelota*", "*botella*" y "*peluche*", mientras que en los objetos enigma vamos a escoger: "*Mochila azul*" y "*mochila rosa*".

- **Paso 9. *Escoger la posición para cada objeto.*** La pantalla que nos debe salir es una lista con los objetos escogidos anteriormente.

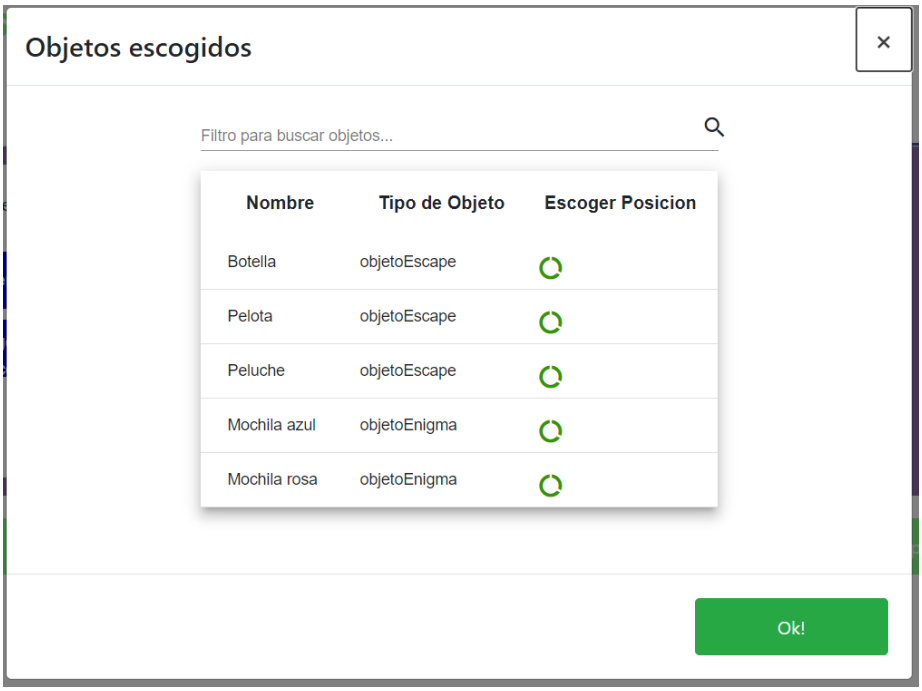


Figura 3.10A Ver objetos escogidos

Para los objetos escape únicamente nos dejará escoger desde la posición 1 a la posición 3, mientras que los objetos enigma irán desde la 4 hasta la 5. Las posiciones se pueden observar en las imágenes que hemos visto previamente del escenario. En nuestro caso, las posiciones serán las siguientes:

- ***Pelota*** → Posición 1.

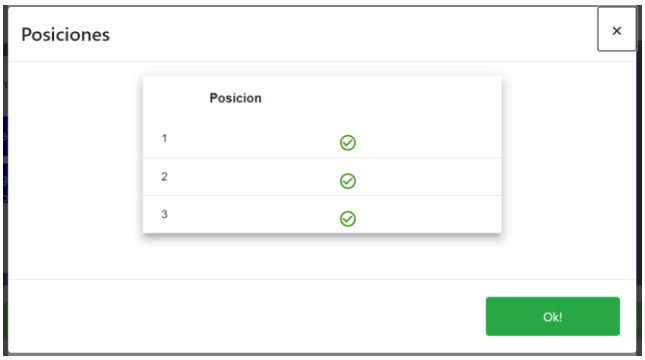


Figura 3.11A Pantalla posiciones I

- ***Botella*** → Posición 2.

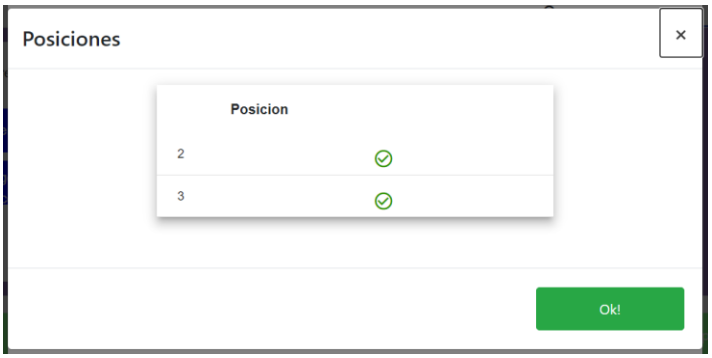


Figura 3.12A Pantalla posiciones II

- ***Peluche*** → Posición 3.

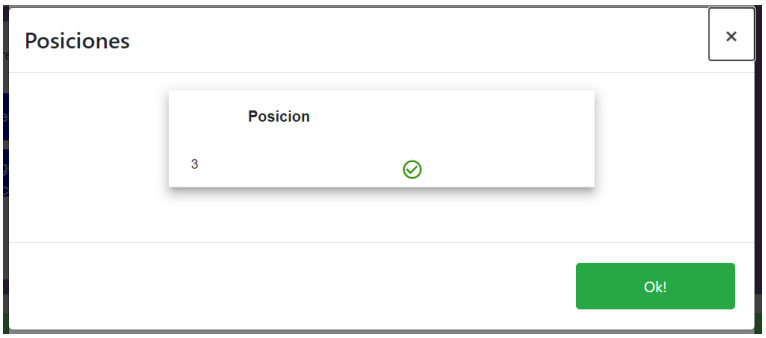


Figura 3.13 Pantalla posiciones III

- ***Mochila azul*** → Posición 4.

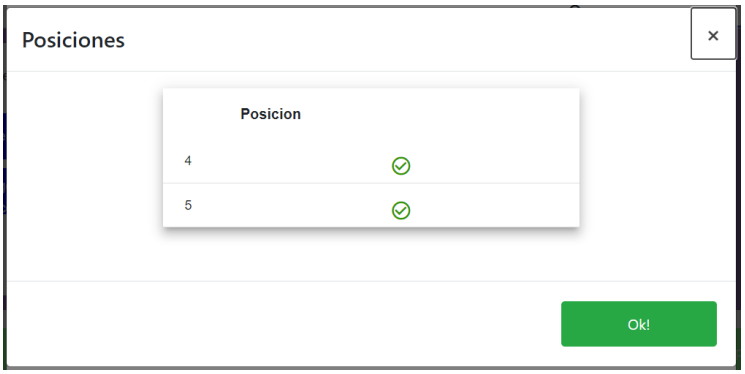


Figura 3.14 Pantalla posiciones IV

- ***Mochila rosa*** → Posición 5.

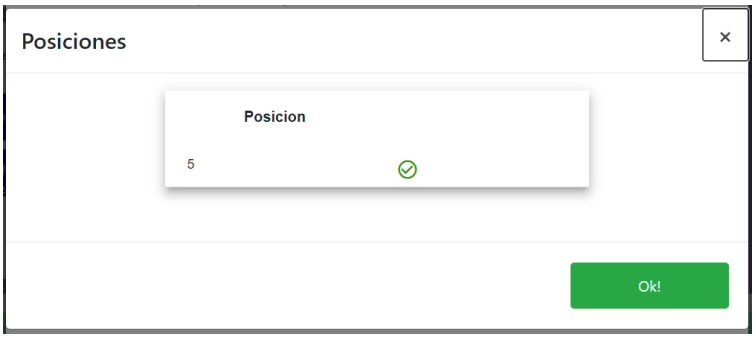


Figura 3.15A Pantalla posiciones V

En el siguiente paso lo que vamos a realizar es la configuración de los objetos enigma.

- **Paso 10. *Configurar los objetos enigma.*** Los objetos enigma son los que tienen una pregunta y una solución, que el profesor lo configura en la pantalla que vemos a continuación.

Figura 3.16A

Otra funcionalidad que tenemos en esta pantalla además de añadir la pregunta y la respuesta del objeto, es la de escoger qué objeto de los dos es el principal. Hay dos tipos de objetos enigma, los principales y los secundarios. A la hora de resolver el enigma en el juego, el alumno obtendrá una pista si el objeto enigma es secundario y una llave si el objeto enigma es principal. En nuestro ejemplo el objeto enigma principal que esconderá la llave será la "*Mochila azu*l". A continuación, vamos a ver los objetos requeridos.

- **Paso 11. *Objetos requeridos.*** Para pasar de escena, el alumno va a necesitar encontrar la llave que se esconde tras uno de los objetos enigma y además lo va a tener que complementar con una serie de objetos escape que decida el profesor, es decir, en nuestro ejemplo podemos poner que para salir del primer escenario el alumno debe tener la llave y en su mochila los objetos botella y peluche. Esto significa que, si no coge la pelota, podrá salir igualmente del escenario, pero que si le falta alguno de los objetos no podrá salir. Los objetos que el profesor decida **NO UTILIZAR** como requeridos en una escena, se acumularán, quiere decir que cuando vaya a configurar la siguiente escena en el caso de que haya más de una escena, en la pantalla que vamos a mostrar a continuación para escoger objetos requeridos, le aparecerá todos los objetos configurados para el segundo escenario y los objetos que no se hayan requerido para el primero, como por ejemplo el caso de la pelota, que podría ser un objeto necesario para pasar del segundo al tercer escenario.

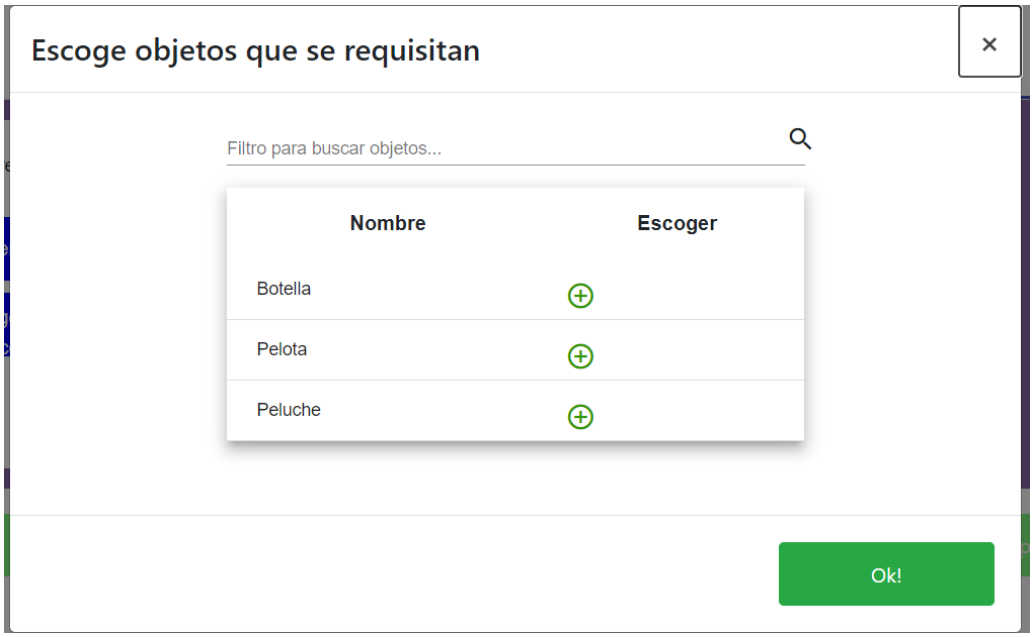


Figura 3.17A Pantalla posiciones I

- **Paso 12. *Configurar la pista.*** Por último, queda configurar la pista. El objeto pista estará escondido en uno de los objetos enigma. El profesor pondrá el texto que él crea conveniente para ayudar al alumno a superar la otra prueba o alguna indicación que él crea conveniente.

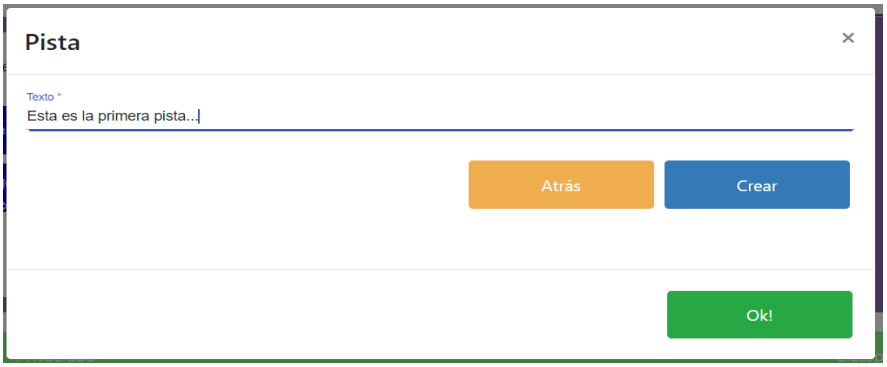


Figura 3.18A

Estos serían los pasos a seguir a la hora de configurar una escena, una vez acabada le damos a Crear y en el caso de que hayamos escogido una escena nos creará el juego y en el caso de que hayamos escogido más de una escena, como es el caso ejemplo, nos volverá a mostrar la pantalla para configurar la segunda escena.
Los datos que pondremos en el ejemplo para la segunda escena serán los siguientes:

Escenario	**Parking**		
Objetos escogidos y Posiciones	**Bidón de gasolina**	***Objeto escape***	**1**
	Trapo	***Objeto escape***	**2**
	Gafas de sol	***Objeto escape***	**3**
	Caja fuerte	***Objeto enigma***	**4**
	Maleta	***Objeto enigma***	**5**
Objetos requeridos	**Bidón de gasolina**		
	Trapo		
	Gafas de sol		
	Peluche		
Objetos enigma	***Nombre***	***Pregunta***	***Respuesta***
	Caja fuerte	¿Qué número va detrás del 2?	3
	Maleta	¿Qué llevo en la maleta?	Ropa

Pista	“Piensa un poco.”

Tabla1A. Tabla ejemplo

Una vez configurado el juego, en la página de inicio nos indicará que se ha creado correctamente y nos lo mostrará en la tabla de juegos activos.

Se puede realizar la comprobación para ver si se ha creado correctamente de dos maneras:

- Mirar en la base de datos si se ha creado como lo hemos configurado. Se nos deben haber creado los siguientes apartados:

  - ***Alumno juego escape room.***

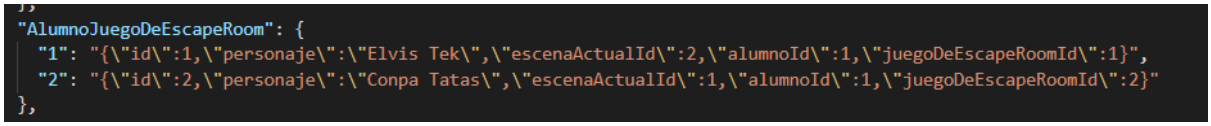


Figura 3.19A Alumno juego escape room en la base de datos

  - ***Juego de escape room.***

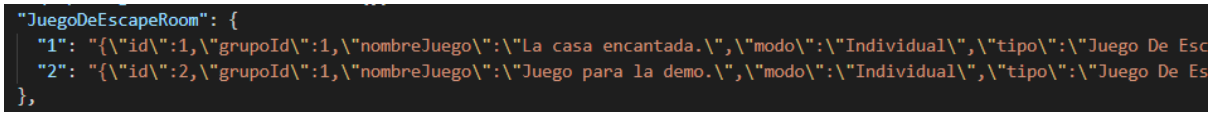


Figura 3.20A Juego de escape room en la base de datos

  - ***Escena de juego.***

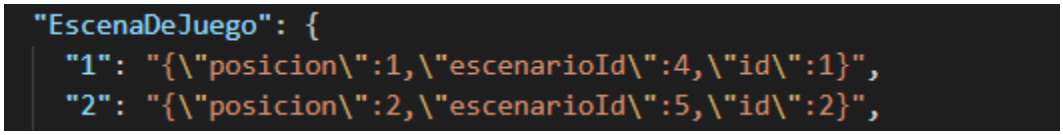


Figura 3.21A Escena de juego en la base de datos

  - ***Partida de escape.***

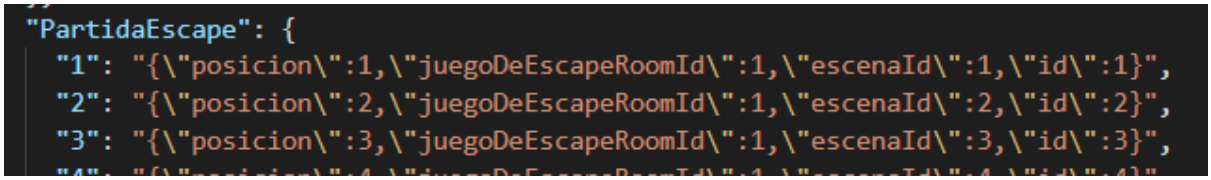


Figura 3.22A Partida escape en la base de datos

- ***Objetos juego.***


```
"ObjetoJuego": {
  "1": "{\"id\":1,\"pregunta\":\"string\",\"respuesta\":\"string\",\"usado\":false,\"recogido\
  "2": "{\"id\":2,\"pregunta\":\"string\",\"respuesta\":\"string\",\"usado\":false,\"recogido\
  "3": "{\"id\":3,\"pregunta\":\"string\",\"respuesta\":\"string\",\"usado\":false,\"recogido\
  "4": "{\"id\":4,\"pregunta\":\"Estate atento a todo tu alrededor.\",\"respuesta\":\"string\"
  "5": "{\"id\":5,\"pregunta\":\"string\",\"respuesta\":\"string\",\"usado\":false,\"recogido\
  "6": "{\"id\":6,\"pregunta\":\"Estate atento a todo tu alrededor.\",\"respuesta\":\"string\"
  "7": "{\"id\":7,\"pregunta\":\"string\",\"respuesta\":\"string\",\"usado\":true,\"recogido\"
```


Figura 3.23A Objeto juego en la base de datos

Los objetos globales, los escenarios y las imágenes del escenario no se modificarán al crear el juego, ya que son recursos que se pueden reutilizar en los diferentes Escape Rooms.

- Iniciar sesión en el móvil del alumno con el usuario alumno al que le has asignado el juego y jugar. Las pantallas que deberían salir con la configuración que hemos seguido en los ejemplos son las siguientes:

  - Escena 1.

Figura 3.23A

  - Escena 2.

Figura 3.24A

ANEXO 4 - Guía para crear un modelo en loopback

Este documento contiene los pasos a seguir para crear un modelo en Loopback siguiendo las directrices impuestas en el proyecto de Classpip.

Los requisitos previos para poder realizar la creación de un modelo es tener instaladas las herramientas de Classpip: Node, angular y loopback.

Las directrices son las siguientes:

Formato nombre del modelo	La primera letra en mayúscula
Formato atributos del modelo	La primera letra en minúscula
Formato relaciones del modelo	La primera letra en mayúscula y en plural

Figura 4.1A Tabla de las directrices a seguir

A continuación, vamos a crear un modelo ejemplo, el modelo se llamará Vendedor, y vamos a explicar paso por paso cada punto.

- **Paso 1: *Crear la base del modelo en Loopback***. Poner el comando *"lb model"* en el terminal. Nos aparecerá lo siguiente:

Figura 4.2A Paso 1 en la creación de un modelo

El nombre del modelo tendrá la primera letra en mayúscula, en nuestro caso, el nombre será "*Vendedor*".
Una vez escribamos el nombre, nos pedirá que seleccionemos el origen de datos en la base de datos. Pondremos "*baseDatos*".

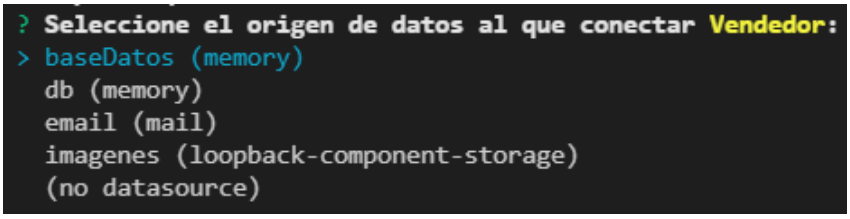


Figura 4.3A Paso 2 en la creación de un modelo

El siguiente paso será seleccionar la clase base Modelo, que será "*PersistedModel*" para poder configurarlo a posteriori.

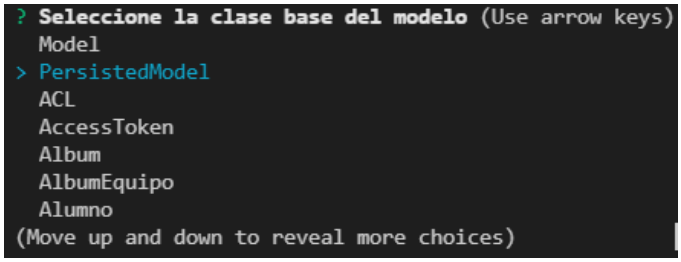


Figura 4.4A Paso 3 en la creación de un modelo

La siguiente opción que nos hace marcar es si queremos exponerlo en la API REST, la opción que marcaremos siempre es "Sí" ya que entre otras cosas podremos comprobar que se ha creado bien con la página del Loopback.

Figura 4.5A Paso 4 en la creación de un modelo

En el siguiente apartado se tendrá que poner el nombre con el que se expone la clase en la API, que es el que utilizaremos para realizar las peticiones. Como marca el estándar, se tendrá que poner en plural con la primera letra en Mayúscula, en nuestro caso será "*Vendedores*".

Figura 4.6A Paso 5 en la creación de un modelo

Por último, en la configuración básica del modelo, tendremos que indicar el tipo de modelo, en el que pondremos "*Común*".

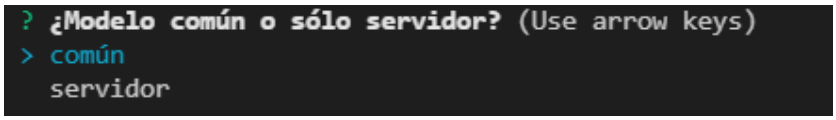


Figura 4.7A Paso 6 en la creación de un modelo

- **Paso 2: *Añadir atributos al modelo***. Una vez creada la base del modelo, el siguiente paso es añadir los atributos. La regla para los atributos es que todas las letras estarán en minúscula. En nuestro caso añadiremos a la clase "*Vendedor*" los atributos de "*nombre*" (string) y "*edad*" (number).
  Cuando creas un atributo, te permite indicar el nombre, el tipo de atributo que será, si es necesario o no y el valor predeterminado. En el caso de "*nombre*" serán los siguientes:

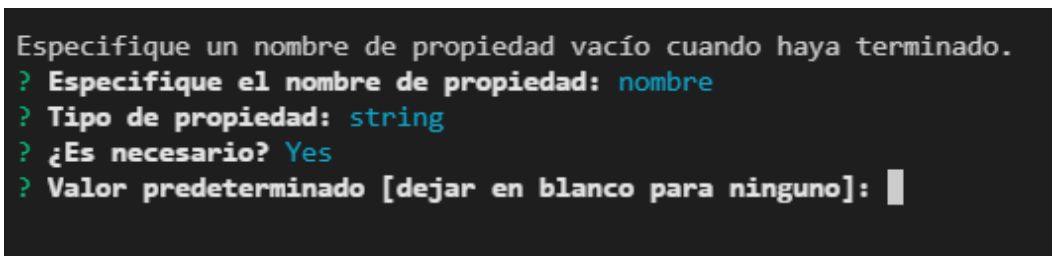


Figura 4.8A Paso 7 en la creación de un modelo

En el caso de *"edad"* serán:

```
Vamos a añadir otra propiedad de Vendedor.
Especifique un nombre de propiedad vacío cuando haya terminado.
? Especifique el nombre de propiedad: edad
? Tipo de propiedad: number
? ¿Es necesario? Yes
? Valor predeterminado [dejar en blanco para ninguno]: 5
```

Figura 4.9A Paso 8 en la creación de un modelo

Una vez finalizado el modelo, clicamos enter para que se cree en loopback. En la base de datos se habrán creado los siguientes ficheros:

- ***vendedor.js***

Figura 4.10A Fichero vendedor.js

- ***vendedor.json***

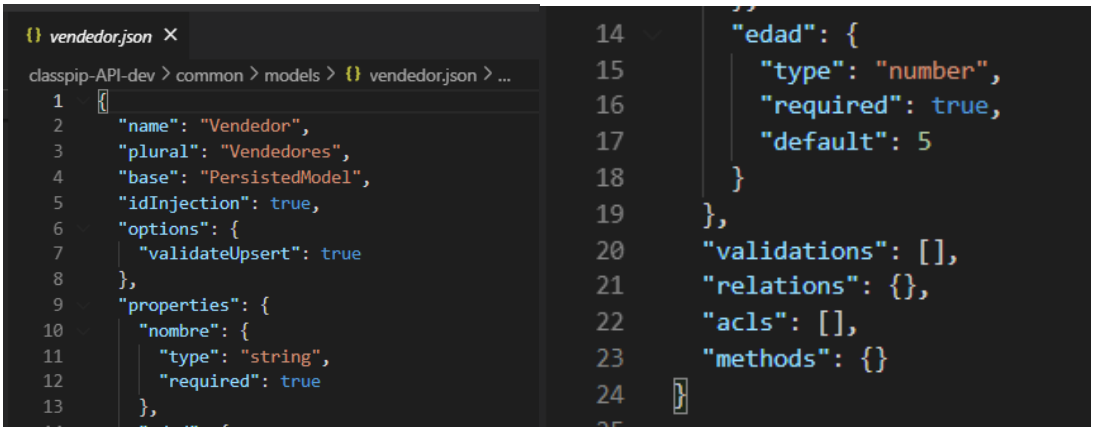


Figura 4.11A y 4.12A Fichero vendedor.json

Como podemos observar en la imagen, se ha creado el modelo con los parámetros que hemos ido indicando.
Para comprobar que hemos creado el modelo correctamente, realizaremos el paso 3.

- **Paso 3:** ***Entrar en la API y comprobar que se ha creado correctamente.*** Primero ejecutaremos npm start para ejecutar la API. Una vez la API esté corriendo, ponemos en el navegador la siguiente URL: *http://localhost:3000/explorer/*

Esta es la parte visual de la clase *Vendedor* en la API.

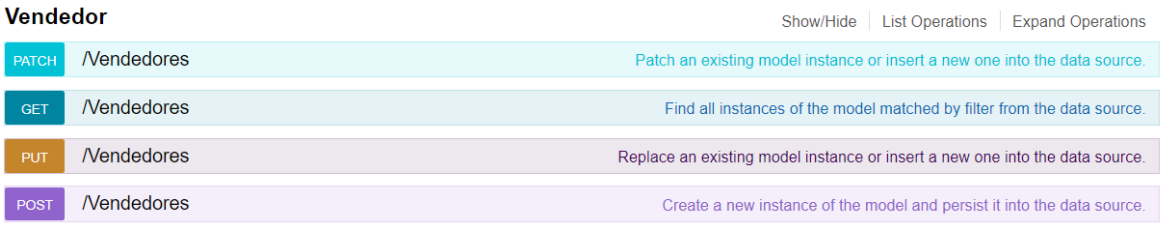


Figura 4.13A Modelo vendedor en loopback

Como podemos observar, el nombre de la clase sale en mayúscula y en singular, "*Vendedor*", mientras que la URL a la que tenemos que hacer la petición es el plural, "*Vendedores*". El modelo se verá de la siguiente manera:

Figura 4.14A Atributos de la clase Vendedor

El siguiente paso va a ser crear una relación entre la clase "*Vendedor*" y "*JuegoDeEscapeRoom*". Hay dos maneras de crear una relación, una es con el comando lb relation y la otra es crear la directamente en el JSON, que es la más fácil y la que vamos a enseñar a continuación.

- **Paso 4: *Crear una relación entre modelos.*** A continuación vamos a crear una relación 1:N entre "*Vendedor*" y "*JuegoDeEscapeRoom*". Dependiendo en la clase que creemos la relación, será donde tengamos la URL, vamos a mostrarlo a continuación.
  Creamos la relación en la clase "*Vendedor*", "*Vendedor*" has many "*JuegosDeEscapeRoom*". El nombre de la relación será el nombre de la clase con la que se hace la relación pero en plural, en este caso se llamará "*JuegosDeEscapeRoom*". Cabe remarcar que el plural será en el objeto, es decir, en este caso "*Juegos*", no será por ejemplo "*JuegoDeEscapeRooms*".

Figura 4.15A Relación JuegosDeEscapeRoom

La foreignKey será un atributo de la clase con la que se hace la relación, en este caso "*JuegoDeEscapeRoom*", se escribirá en minúscula.
El resultado será el siguiente:

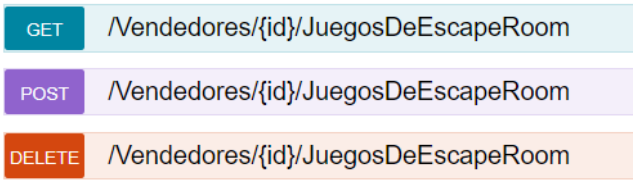


Figura 4.16A URL vendedores / juegos de escape

Figura 4.17A Atributos de la clase JuegoDeEscapeRoom

Como hemos creado la relación en el modelo de "Vendedores", la relación en la URL la encabeza "Vendedores". El otro aspecto es que se ha creado una foreign key en "JuegoDeEscapeRoom" con el nombre que le hemos indicado: "vendedorId".

ANEXO 5 - Documento de pruebas de escape room

En este escrito está toda la documentación referente a las pruebas sobre el juego escape room.

CREACIÓN DEL JUEGO ESCAPE ROOM EN EL DASHBOARD

Usuarios que realizan la prueba:

- **User 1**
- **User 2**
- **User 3**
- **User 4**

El objetivo de la prueba es detectar anomalías en la creación del juego. Para ello, 4 personas vamos a logearnos en el dashboard con cuentas de profesores diferentes y vamos a configurar cuatro Escape Rooms de maneras distintas para comprobar la robustez del código.

Requisitos previos para realizar la prueba:

- Se deberá crear un alumno y asociarlo a un grupo.
- Se deberán haber creado los siguientes escenarios según el usuario:
  - ***User 1:*** Celda.
  - ***User 2:*** Parque y habitación.
  - ***User 3:*** Parking, habitación y baño.
  - ***User 4:*** Parque, habitación, sótano y cocina.
- Se deberán haber creado los siguientes objetos según el usuario.
  OEs → Objeto escape.
  OEn → Objeto enigma.
  - ***User 1:*** Lima, cadenas y bolígrafo (OEs). Baúl y libreta (OEn).
  - ***User 2:*** Pelota, botella, peluche, jarrón, vaso y lámpara. (OEs). Mochila azul, mochila rosa, cuadro y caja fuerte. (OEn).
  - ***User 3:*** Gasolina, trapo, gafas de sol, jarrón, botella, lámpara, cepillo de dientes, jabón y bote de pastillas (OEs). Maleta, libreta, caja fuerte, cuadro, báscula y espejo (OEn).
  - ***User 4:*** Pelota, botella, peluche, jarrón, botella, bolígrafo, olla, aceite, vaso, cadena de pie, trapo y bandeja (OEs). Mochila azul, mochila rosa, caja fuerte, libreta, reloj, microondas, mapa y baúl.

Al final del documento están los JSON que han de cargar en la base de datos cada usuario para ahorrarse el proceso de creación de los elementos.

- USER 1

Nombre del escape room: Prisión.
Modo: Individual.
Número de escenas: 1

ESCENA 1
***Escenario*:** Celda
Objetos escogidos:
- ***Objetos tipo escape:*** Lima, bolígrafo y cadenas.
- ***Objetos tipo enigma:*** Baúl y libreta.

Posiciones:
1. Cadenas.
2. Lima.
3. Bolígrafo.
4. Baúl.
5. Libreta.

Modificar objetos enigma:
- ***Baúl****:* Escoger como **principal**.
  - Pregunta: ¿Soy el baúl de tus recuerdos?
  - Respuesta: No, soy el de los de Mario.
- ***Libreta:***
  - Pregunta: ¿Quién es el freestyler más libretero?
  - Respuesta: Papo.

Escoger objetos que se requisitan para la escena:
- Cadenas.
- Lima.

Pista:
- Texto: Los recuerdos son del diseñador...

- USER 2

Nombre del escape room: Poblado.
Modo: Individual.
Número de escenas: 2

ESCENA 1

***Escenario*:** Parque
Objetos escogidos:
- ***Objetos tipo escape:*** Pelota, botella y peluche.
- ***Objetos tipo enigma:*** Mochila azul y mochila rosa.

Posiciones:
1. Pelota.
2. Botella.
3. Peluche.
4. Mochila azul.
5. Mochila rosa.

Modificar objetos enigma:
- ***Mochila azul****:* Escoger como **principal**.

    - Pregunta: ¿De qué es el bocata que guardo?
    - Respuesta: De chorizo.
- ***Mochila rosa:***
    - Pregunta: ¿Quién es el que ha copiado más durante la carrera?
    - Respuesta: Carlos.

Escoger objetos que se requisitan para la escena:

- Pelota.
- Botella.

Pista:

- Texto: El apellido del que copia es Vazquéz...

ESCENA 2

***Escenario*:** Habitación
Objetos escogidos:

- ***Objetos tipo escape:*** Jarrón, vaso y lámpara.
- ***Objetos tipo enigma:*** Cuadro y caja fuerte.

Posiciones:

1. Lámpara.
2. Vaso.
3. Jarrón.
4. Cuadro.
5. Caja fuerte.

Modificar objetos enigma:

- ***Cuadro****:* Escoger como **principal**.
    - Pregunta: ¿Quién pintó este cuadro?
    - Respuesta: Da vinci.
- ***Caja fuerte:***
    - Pregunta: ¿2x10x5-5 és …?
    - Respuesta: 95.

Escoger objetos que se requisitan para la escena:

- Lámpara.
- Vaso.
- Peluche.

Pista:

- Texto: El pintor Da cosas...

- USER 3

Nombre del escape room: Mansión.
Modo: Individual.
Número de escenas: 3

ESCENA 1

***Escenario*:** Parking
Objetos escogidos:

- ***Objetos tipo escape:*** Gasolina, trapo y gafas de sol.

- ***Objetos tipo enigma:*** Maleta y libreta.

Posiciones:

1. Gasolina.
2. Trapo.
3. Gafas de sol.
4. Maleta.
5. Caja fuerte.

Modificar objetos enigma:

- ***Maleta****:* Escoger como **principal**.
  - Pregunta: ¿Qué día es el cumpleaños del personaje?
  - Respuesta: El 17 de Febrero de 1998.
- ***Caja fuerte:***
  - Pregunta: ¿Quién canta mejor de los cuatro?
  - Respuesta: Mario.

Escoger objetos que se requisitan para la escena:

- Gasolina.
- Trapo.

Pista:

- Texto: 17 manzanas, en el mes de los enamorados, justo dos años antes de cambiar de milenio, sucedió una cosa muy extraña...

ESCENA 2

Escenario**:** Habitación

Objetos escogidos:

- ***Objetos tipo escape:*** Jarrón, botella y lámpara.
- ***Objetos tipo enigma:*** Caja fuerte y cuadro.

Posiciones:

1. Jarrón.
2. Botella.
3. Lámpara.
4. Caja fuerte.
5. Cuadro.

Modificar objetos enigma:

- ***Caja fuerte****:* Escoger como **principal**.
  - Pregunta: ¿Si te pregunto por la raíz cuadrada de 16?
  - Respuesta: Es 4.
- ***Cuadro:***
  - Pregunta: ¿Con qué pintura está trazado el cuadro?
  - Respuesta: Con pinturas de colores.

Escoger objetos que se requisitan para la escena:

- Botella.
- Gafas de sol.

Pista:

- Texto: Es como multiplicar el nombre de un modelo de coche…

ESCENA 3

***Escenario*:** Baño
Objetos escogidos:
- ***Objetos tipo escape:*** Cepillo de dientes, jabón y bote de pastillas.
- ***Objetos tipo enigma:*** Báscula y espejo.

Posiciones:
1. Cepillo de dientes.
2. Jabón.
3. Bote de pastillas.
4. Báscula.
5. Reloj.

Modificar objetos enigma:
- ***Báscula****:* Escoger como **principal**.
  - Pregunta: ¿Qué pesa más, 1 kilo de paja o 1 kilo de plomo?
  - Respuesta: Pesan lo mismo.
- ***Reloj:***
  - Pregunta: ¿Espejito espejito, quién es el más bonito?
  - Respuesta: Iván.

Escoger objetos que se requisitan para la escena:
- Cepillo de dientes.
- Jabón.

Pista:
- Texto: La paja siempre engaña...

- USER 4

Nombre del escape room: La casa encantada.
Modo: Individual.
Número de escenas: 4

ESCENA 1

***Escenario*:** Parque
Objetos escogidos:
- ***Objetos tipo escape:*** Pelota, botella y peluche.
- ***Objetos tipo enigma:*** Mochila azul y mochila rosa.

Posiciones:
1. Pelota.
2. Botella.
3. Peluche.
4. Mochila azul.
5. Mochila rosa.

Modificar objetos enigma:
- ***Mochila azul****:* Escoger como **principal**.
  - Pregunta: ¿De qué es el bocata que guardo?
  - Respuesta: De chorizo.
- ***Mochila rosa:***
  - Pregunta: ¿Quién es el que ha copiado más durante la carrera?

- Respuesta: Carlos.

Escoger objetos que se requisitan para la escena:
- Pelota.
- Botella.

Pista:
- Texto: El apellido del que copia es Vázquez…

ESCENA 2

***Escenario*:** Habitación
Objetos escogidos:
- ***Objetos tipo escape:*** Jarrón, lámparay bolígrafo.
- ***Objetos tipo enigma:*** Caja fuerte y espejo.

Posiciones:
1. Jarrón.
2. Lámpara.
3. Bolígrafo.
4. Caja fuerte.
5. Espejo.

Modificar objetos enigma:
- ***Caja fuerte****:* Escoger como **principal**.
  - Pregunta: ¿Qué día es el cumpleaños del personaje?
  - Respuesta: El 17 de Febrero de 1998.
- ***Espejo:***
  - Pregunta: ¿Quién canta mejor de los cuatro?
  - Respuesta: Mario.

Escoger objetos que se requisitan para la escena:
- Bolígrafo.
- Peluche.

Pista:
- Texto: 17 manzanas, en el mes de los enamorados, justo dos años antes de cambiar de milenio, sucedió una cosa muy extraña…

ESCENA 3

***Escenario*:** Cocina
Objetos escogidos:
- ***Objetos tipo escape:*** Olla, aceite y vaso.
- ***Objetos tipo enigma:*** Reloj y microondas.

Posiciones:
1. Olla. (botea)
2. Aceite. (gaf)
3. Vaso.(pel)
4. Reloj.(mal)
5. Microondas. (pantal)

Modificar objetos enigma:
- ***Reloj****:* Escoger como **principal**.
  - Pregunta: ¿Am?

    - Respuesta: 11.
- ***Microondas:***
    - Pregunta: ¿Cuántos grados son 30+25-15? AL REVEEES
    - Respuesta: 40.

Escoger objetos que se requisitan para la escena:

- Olla.
- Aceite.
- Vaso.
- Jarrón.

Pista:

- Texto: Te voy a dar dos pistas por ser especial, la primera es una canción, la segunda es que puede que tengas que dar un buen paseo para salir…

ESCENA 4

***Escenario*:** Sótano

Objetos escogidos:

- ***Objetos tipo escape:*** Cadena de pie, trapo y bandeja.
- ***Objetos tipo enigma:*** Mapa y baúl.

Posiciones:

1. Cadena de pie.
2. Trapo.
3. Bandeja.
4. Mapa.
5. Baúl.

Modificar objetos enigma:

- ***Mapa****:* Escoger como **principal**.
    - Pregunta: ¿Dónde estamos?
    - Respuesta: En España.
- ***Baúl:***
    - Pregunta: ¿Qué crees que guardo aquí?
    - Respuesta: Ropa.

Escoger objetos que se requisitan para la escena:

- Cadena de pie.
- Trapo.
- Bandeja.

Pista:

- Texto: Dentro de mi abanico hay rebecas...

COMPROBACIÓN DE LA MECÁNICA JUEGO ESCAPE ROOM EN EL MÓVIL DE ESTUDIANTE

Usuarios que realizan la prueba:

- **User 1**
- **User 2**
- **User 3**
- **User 4**

El objetivo de la prueba es detectar anomalías en la mecánica del juego y comprobar que se ha creado el juego correctamente desde el Dashboard. Para ello, 4 personas vamos a loguearnos con el alumno que hemos asignado en la prueba de *"Creación del juego escape room en el Dashboard"* y cada usuario va a realizar una serie de pruebas de maneras distintas para comprobar la robustez del código.

Requisitos previos para realizar la prueba:

- Todos los usuarios deberán haber hecho la prueba de *"Creación del juego escape room en el Dashboard".*

- USER 1

Nombre del escape room: Prisión.
Personaje: Elvis Tek.

ESCENA 1

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- **Escenario**: Celda.
- ***Objetos***:
    1. Cadenas
    2. Lima
    3. Bolígrafo
    4. Baúl
    5. Libreta

1. El primer paso que debemos realizar es clicar el botón de guardar arriba a la izquierda. Cuando pregunte, confirmar el guardado y salir del juego.
2. Volvemos a entrar, ahora ya no nos deberá salir la pantalla de elección de personaje, si no la de *Continuar historia*.
3. Coger cadenas y bolígrafo, mirar la mochila. *Debería aparecer los objetos cadenas y bolígrafo.*
4. Resolver la libreta y el baúl e intentar salir. *Debería decir que faltan objetos requeridos.*
5. Coger la lima y salir, *debería decir que se ha acabado el juego.*

- USER 2

Nombre del escape room: Poblado.
Personaje: Elvis Tek.

ESCENA 1

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Parque.
- ***Objetos***:
    1. Pelota.
    2. Botella.
    3. Peluche.
    4. Mochila azul.
    5. Mochila rosa.

1. El primer paso que debemos realizar es clicar el botón de guardar arriba a la izquierda. Cuando pregunte, confirmar el guardado y salir del juego.
2. Volvemos a entrar, ahora ya no nos deberá salir la pantalla de elección de personaje, si no la de *Continuar historia*.
3. Coger pelota y botella, mirar la mochila. *Debería aparecer los objetos pelota y botella.*
4. Resolver la mochila azul y la mochila rosa e intentar salir. *Debería decir que puedes pasar al siguiente escenario.*

ESCENA 2

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Habitación.
- ***Objetos***:
    1. Lámpara.
    2. Vaso.
    3. Jarrón.
    4. Cuadro.
    5. Caja fuerte.

1. Coger lámpara y vaso, mirar la mochila. *Debería aparecer los objetos lámpara y vaso.*
2. Guardar el juego, salir y volver a entrar. *Debería aparecer el personaje en el segundo escenario, con los objetos en la mochila y el resto en el escenario.*
3. Resolver el cuadro y la caja fuerte e intentar salir. *Debería decir que te faltan objetos requeridos.*
4. Volver al primer escenario, coger el peluche y volver al segundo escenario. Una vez dentro, mirar la mochila. *Debería aparecer los objetos lámpara y peluche.*
5. Intentar salir. *Debería decir que se ha acabado el juego.*

- USER 3

Nombre del escape room: Mansión.
Personaje: Elvis Tek.

ESCENA 1

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Parking.
- ***Objetos***:
    1. Gasolina.
    2. Trapo.
    3. Gafas de sol.
    4. Maleta.
    5. Libreta.

1. El primer paso que debemos realizar es clicar el botón de guardar arriba a la izquierda. Cuando pregunte, confirmar el guardado y salir del juego.
2. Volvemos a entrar, ahora ya no nos deberá salir la pantalla de elección de personaje, si no la de *Continuar historia*.
3. Coger gasolina, trapo y gafas de sol, mirar la mochila. *Debería aparecer los objetos gasolina, trapo y gafas de sol.*
4. Intentar salir. *Debería decir que falta la llave.*
5. Resolver la maleta y la libreta e intentar salir. *Debería decir que puedes pasar al siguiente escenario.*

ESCENA 2

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Habitación.
- ***Objetos***:
    1. Jarrón.
    2. Botella.
    3. Lámpara.
    4. Caja fuerte.
    5. Cuadro.

1. Coger el jarrón y la botella, mirar la mochila. *Debería aparecer los objetos jarrón, botella y gafas de sol.*
2. Resolver el cuadro y la caja fuerte e intentar salir. *Debería dejarte salir al tercer escenario.*

ESCENA 3

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Baño.
- ***Objetos***:

1. Cepillo de dientes.
2. Jabón.
3. Bote de pastillas.
4. Báscula.
5. Espejo.

1. Coger el cepillo de dientes, jabón y el bote de pastillas, mirar la mochila. *Debería aparecer los objetos jarrón, cepillo de dientes, jabón y bote de pastillas.*
2. Resolver el espejo y la báscula e intentar salir. *Debería poner final del juego.*

- USER 4

Nombre del escape room: La casa encantada.
Personaje: Elvis Tek.

ESCENA 1

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Parque.
- ***Objetos***:
    1. Pelota.
    2. Botella.
    3. Peluche.
    4. Mochila azul.
    5. Mochila rosa.

1. El primer paso que debemos realizar es clicar el botón de guardar arriba a la izquierda. Cuando pregunte, confirmar el guardado y salir del juego.
2. Volvemos a entrar, ahora ya no nos deberá salir la pantalla de elección de personaje, si no la de *Continuar historia*.
3. Coger pelota, botella y peluche, mirar la mochila. *Debería aparecer los objetos pelota, botella y peluche.*
4. Intentar salir. *Debería decir que falta la llave.*
5. Resolver la mochila azul y mochila rosa e intentar salir. *Debería decir que puedes pasar al siguiente escenario.*

ESCENA 2

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Habitación.
- ***Objetos***:
    1. Jarrón.
    2. Lámpara.
    3. Bolígrafo.
    4. Caja fuerte.
    5. Libreta.

3. Coger el jarrón y la botella, mirar la mochila. *Debería aparecer los objetos jarrón, botella y gafas de sol.*
4. Resolver el cuadro y la caja fuerte e intentar salir. *Debería dejarte salir al tercer escenario.*

ESCENA 3

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Cocina.
- ***Objetos***:
    1. Olla.
    2. Aceite.
    3. Vaso.
    4. Reloj.
    5. Microondas.

1. Coger la olla, el aceite y el vaso, mirar la mochila. *Debería aparecer los objetos olla, aceite, vaso y jarrón.*
2. Resolver el reloj y el microondas e intentar salir. *Debería dejarte salir y pasar al cuarto escenario.*

ESCENA 4

El primer escenario, si se ha realizado correctamente la creación del juego, deberá aparecer lo siguiente:

- ***Escenario***: Sótano.
- ***Objetos***:
    1. Cadena de pie.
    2. Trapo.
    3. Bandeja.
    4. Mapa.
    5. Baúl.

1. Coger la cadena de pie, el trapo y la bandeja, mirar la mochila. *Debería aparecer los objetos cadena de pie, trapo y bandeja.*
2. Resolver el mapa y el baúl e intentar salir. *Debería indicar el final del escape room.*