



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE GRADO

Título: Mejoras en una app para la gestión de la gamificación (II)

Titulación: Grado en Ingeniería Telemática

Autor: Sumaya Benali Laarif

Director: Miguel Valero García

Fecha: 8 de julio del 2022

Título: Mejoras en una app para la gestión de la gamificación (I)

Titulación: Grado en Ingeniería Telemática

Autor: Sumaya Benali Laarif

Director: Miguel Valero García

Fecha: 8 de julio del 2022

Resumen

El proyecto, que se describe a continuación, está basado en la implementación de cinco juegos para una aplicación de gamificación llamada Classpip, orientada al sector educativo. Su uso fomenta e incrementa el dinamismo y rendimiento en las clases, ya que el objetivo es motivar a los alumnos con juegos competitivos a participar y aprender.

En este caso, el proyecto en el que se ha trabajado va dirigido en su totalidad a la experiencia de usuario del profesor, donde se ha trabajado en la implementación de cinco juegos para la aplicación móvil del profesor que se explicarán a continuación.

El objetivo de este proyecto era facilitar el uso de Classpip para los docentes en las clases. Classpip tiene implementado un Dashboard donde el profesor tiene el control de la creación, modificación y uso de los juegos desde esta una página web. Pero teniendo en cuenta las limitaciones que tiene el dashboard de cara a la movilidad o el uso de Classpip en las clases, una aplicación móvil para el docente era una opción a tener en cuenta.

Para cubrir esta necesidad se ha trabajado en la migración de cinco juegos que previamente existían en el Dashboard. Reutilizando así algunas funcionalidades ya existentes, aplicando varias mejoras estéticas y añadiendo otras funcionalidades nuevas y necesarias en varios juegos, para mejorar la experiencia de usuario del docente.

El equipo para llevar a cabo este proyecto está formado por dos personas. Ambos miembros del equipo han trabajado en la implementación de todos los juegos que se describen a continuación en este documento.

Para la gestión, organización de los tiempos y la repartición de tareas, se ha usado la metodología SCRUM. Y se realizó la repartición de las tareas en función de las diferentes funcionalidades que requiere tener cada uno de los juegos.

En este documento se podrá seguir detalladamente los objetivos, la estrategia de trabajo y el análisis previo de cada uno de los juegos a implementar con su correspondiente descripción. También el proceso de implementación con la correspondiente explicación de código, las decisiones tomadas y retos superados, y las pruebas realizadas para la demostración de un correcto funcionamiento.

Título: Mejoras en una app para la gestión de la gamificación (I)

Titulación: Grado en Ingeniería Telemática

Autor: Sumaya Benali Laarif

Director: Miguel Valero García

Fecha: 8 de julio del 2022

Overview

The project described below is based on the implementation of five games for a gamification application called Classpip, aimed at the educational sector. Its use encourages and increases dynamism and performance in classes, since the objective is to motivate students with competitive games to participate and learn.

In this case, the project that has been worked on is entirely aimed at the teacher's user experience, where they have worked on the implementation of five games for the teacher's mobile application that will be explained below.

The objective of this project was to facilitate the use of Classpip for teachers in the classroom. Classpip has implemented a Dashboard where the teacher has control of the creation, modification and use of the games from this web page. But taking into account the limitations that the dashboard has in terms of mobility or the use of Classpip in classes, a mobile application for the teacher was an option to consider.

To cover this need, we have worked on the migration of five games that previously existed in the Dashboard. Thus, reusing some existing functionalities, applying several aesthetic improvements and adding other new and necessary functionalities in several games, to improve the user experience of the teacher.

The team to carry out this project consists of two people. Both team members have worked on the implementation of all the games described below in this document.

For the management, organization of times and the distribution of tasks, the SCRUM methodology has been used. And the distribution of tasks was carried out based on the different functionalities that each of the games requires.

In this document it will be possible to follow in detail the objectives, the work strategy and the previous analysis of each of the games to be implemented with their corresponding description. Also, the implementation process with the corresponding code explanation, the decisions made and challenges overcome, and the tests carried out to demonstrate correct operation.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN.....	7
1.1. Proyecto Classpip	7
1.2. Estructura de la memoria	8
1.3. Equipo de trabajo	9
1.4. ¿Qué he hecho en este TFG?.....	9
1.5. Motivación personal.....	11
CAPÍTULO 2. INTRODUCCIÓN A CLASSPIP	12
2.1. ¿Qué es la gamificación?	12
2.2 Arquitectura del proyecto	12
2.3 Funcionalidades del ecosistema.....	14
2.4 ¿Qué herramientas del ecosistema hemos utilizado en nuestro proyecto?.....	17
CAPÍTULO 3: OBJETIVOS Y PLAN DE TRABAJO.....	19
3.1. Metodología Scrum.....	19
3.2. Herramienta Trello.....	24
CAPÍTULO 4: TAREAS PREVIAS.....	25
4.1 Pagina de registro	25
CAPÍTULO 5: ANÁLISIS DE LOS JUEGOS A IMPLEMENTAR.....	29
5.1 Creación del grupo.....	29
5.2. Juego de competición.....	29
5.2.1. Liga.....	30
5.2.2. Fórmula Uno	31
5.2.3. Torneo	32
5.3. Evaluación	33
5.3.1. Crear Juego	33
5.3.2. Rúbrica	33
5.3.3. Preguntas abiertas.....	34
5.3.4. Punto de vista del profesor.....	34
5.3.5. Punto de vista del alumno	35
5.4. Control de trabajo en grupo.....	35
5.4.1 Crear Juego	35
5.4.2. Punto de vista del profesor.....	35
5.4.3. Punto de vista del alumno	36
5.5. Juego de votaciones	36

5.5.1. Crear Juego	36
5.5.2. Uno a todos	36
5.5.3. Todos a uno	37
5.5.4. A opciones	38
5.6. Cuestionario de satisfacción	38
5.6.1. Crear Juego	39
5.6.2. Punto de vista del profesor	39
CAPÍTULO 6. JUEGO DE COMPETICIÓN	40
6.1. ¿En qué consiste el juego de competición?	40
6.2. Modalidades del juego de competición	40
6.3 Tipología del juego de competición	41
6.4. Página de Inicio	41
6.4.1 ¿En qué consiste la funcionalidad?	41
6.4.2 Planificación para la implementación	42
6.4.3 Implementación del juego	42
6.4.4 Retos destacados	44
6.5. Funcionalidad 1: Ver progreso	44
6.5.1 ¿En qué consiste la funcionalidad?	44
6.5.2 Planificación para la implementación	44
6.5.3 Implementación del juego	45
6.5.4 Retos destacados	47
6.6. Funcionalidad 2: Editar Jornadas	47
6.6.1 ¿En qué consiste la funcionalidad?	47
6.6.2 Planificación para la implementación	47
6.6.3 Implementación del juego	48
6.6.4 Retos destacados	50
6.7. Funcionalidad 3: Editar Puntos	50
6.7.1 ¿En qué consiste la funcionalidad?	50
6.7.2 Planificación para la implementación	50
6.7.3 Implementación del juego	51
6.7.4 Retos destacados	53
6.8. Funcionalidad 4: Desactivar	53
6.8.1 ¿En qué consiste la funcionalidad?	53
6.8.2 Planificación para la implementación	53
6.8.3 Implementación del juego	54
6.8.4 Retos destacados	54
6.9. Testing del correcto funcionamiento	55

CAPÍTULO 7. JUEGO DE CONTROL DE TRABAJO.....	61
7.1. ¿En qué consiste el juego de control de trabajo?	61
7.2. Funcionalidad 1: Visualización de la información	62
7.2.1 Planificación para la implementación	62
7.2.2 Implementación del juego.....	64
7.2.3 Retos destacados.....	67
7.3. Funcionalidad 2: Envío de correos electrónicos	69
7.3.1 Planificación para la implementación	69
7.3.2 Implementación del juego.....	70
7.3.3 Retos destacados.....	74
7.4. Testing del correcto funcionamiento	74
CAPÍTULO 8. JUEGO DE CUESTIONARIO DE SATISFACCIÓN.....	77
8.1. ¿En qué consiste el juego de cuestionario de satisfacción?	77
8.2. Funcionalidad 1: Visualización de la información	78
8.2.1 Planificación para la implementación	78
8.2.2 Implementación del juego.....	80
8.2.3 Retos destacados.....	82
8.3. Testing del correcto funcionamiento	82
CAPÍTULO 9. DISEÑO GENERAL DE LA APLICACIÓN.....	84
9.1. Cards juegos: Cambio de la estética para mejorar la funcionalidad.....	84
9.2. Pantalla de inicio de juego de competición	85
9.3. Botones: Estética según el tipo de función del botón.....	86
9.4. Tablas: Mejora estética para correcta visualización.....	87
CAPÍTULO 10. RETOS MÁS DESTACADOS	89
10.1. Nodemailer: Herramienta de envío de correos electrónicos.....	89
10.1.1. Gmail: Problemas de autenticación	89
10.1.2. Gmail: Problemas de autenticación	92
10.1.3. Gmail: Uso de una contraseña de aplicación.....	94
10.2. Problemática con las versiones	97
10.2.1. Primera propuesta: Realizar los pasos de instalación de nuevo	98
10.2.2. Segunda propuesta: Desinstalar Node completamente	98
10.2.3. Tercera propuesta: NVM.....	99
CAPÍTULO 11. CONCLUSIONES.....	101
11.1. Conclusiones técnicas	101
11.2 Valoración de los objetivos y el plan de trabajo	101
11.3. Valoración personal.....	102
REFERENCIAS.....	103

CAPÍTULO 1. INTRODUCCIÓN

En este capítulo podremos ver las ideas principales del proyecto. Se explicará en qué consiste el proyecto Classpip y también, como está estructurada la memoria. Seguidamente, algunos detalles del equipo de trabajo, cómo ha sido la organización y la división de tareas para llevar a cabo el objetivo del proyecto. A continuación, Se comentará qué se ha hecho exactamente en este TFG, y que diferencias y similitudes tiene este documento en comparativa al documento de mi compañero de trabajo, ya que este TFG se ha hecho por un equipo formado por dos personas. Finalmente, se detallarán las razones por las que decidí participar en dicho proyecto.

1.1. Proyecto Classpip

La idea del proyecto de Classpip se originó a causa del interés en mejorar algo del sector educativo, donde gracias a esta herramienta los alumnos y docentes pueden beneficiarse para mejorar el rendimiento académico en las escuelas y la participación de los alumnos en el aula. Gracias a esta herramienta, el dinamismo en las actividades en clase e incluso el aprendizaje puede verse afectado de forma positiva.

Classpip es una herramienta que utiliza la gamificación, la cual es clave para brindar la motivación necesaria para lograr un mejor rendimiento académico en la escuela a través de minijuegos u otras actividades. Ese es el objetivo de este proyecto, incrementar el rendimiento de los alumnos, creando situaciones para impulsar actitudes como prestar más atención, por el dinamismo que se cree en clase gracias al uso de Classpip.

Este proyecto fue creado por la Escuela de Ingeniería de Telecomunicación y Aeroespacial de Castelldefels (EETAC) de la Universidad Politécnica de Cataluña (UPC), la arquitectura de la aplicación fue desarrollada por un alumno del TFM (Trabajo Fin de Máster) en 2016 y ha seguido avanzando con la ayuda de varios alumnos que han añadido funcionalidades a Classpip en sus proyectos de graduación.

Classpip tiene dos modalidades diferentes de cara al tipo de usuario: Puede ser desde el punto de vista del profesor, donde utilizará todas las herramientas que le ofrece Classpip desde el Dashboard en su ordenador, o puede utilizar su dispositivo móvil para poder realizar algunas de las funciones que ofrece Classpip desde clase. La segunda modalidad, es desde el punto de vista del alumno, que podrá jugar a los juegos que active su profesor desde su dispositivo móvil.

1.2. Estructura de la memoria

La memoria que se presenta a continuación se divide por capítulos, y en cada capítulo se hará énfasis de un tema en concreto sobre el trabajo realizado. A continuación, se explicará de forma resumida la estructura y el contenido de cada uno de estos capítulos.

El primer capítulo es la introducción del proyecto, justo en el capítulo en el que estamos ahora, donde empezamos presentando el proyecto de Classpip y explicando la estructura de la memoria. Seguimos explicando el equipo de trabajo, es decir, cuantos miembros han formado parte de este proyecto, que tareas han realizado cada uno y como se ha llevado a cabo la organización del equipo. A continuación, se explica como he contribuido en el TFG, es decir, que tareas he realizado independientemente de mi compañero de trabajo. Y al final del capítulo 1, se explica la motivación personal y que razones me llevaron a tomar la decisión de elegir este proyecto como mi TFG.

En el capítulo 2 se realiza la introducción a Classpip, explicando que es la gamificación como concepto, cuál es la arquitectura del proyecto, y qué juegos y funcionalidades tiene Classpip para ofrecer a sus usuarios. Este capítulo es importante, ya que es la introducción necesaria para poder entender los siguientes capítulos del proyecto.

El tercer capítulo es la explicación de los objetivos y plan de trabajo de nuestro proyecto, donde se explica la metodología que utilizamos para la organización del equipo y las herramientas que nos han servido para cumplir con plazos de entrega de una manera más organizada.

El cuarto capítulo son las tareas previas que se han realizado antes de empezar con el proyecto 'real', para familiarizarnos con el entorno. Se explican las razones por las cuales quería hacer esa mejora como tarea previa, la implementación de la tarea y el resultado final.

El quinto capítulo es el análisis de los juegos a implementar, donde se analizan de manera muy detallada para poder entender el funcionamiento de cada uno de ellos completamente. El análisis se realiza tanto desde el punto de vista del profesor, que en este caso sería analizar los juegos desde el Dashboard. Como desde el punto de vista del alumno, que sería analizar los juegos desde la aplicación del alumno. A partir de este análisis, podemos obtener una lista de las funcionalidades de cada uno de los juegos, y así posteriormente poder elegir que funcionalidades se migrarían a la aplicación del profesor. Este análisis es muy importante, ya que nos sirvió para entender de manera muy completa, desde varios puntos de vista, el funcionamiento de la aplicación. Y también, gracias a este análisis fuimos capaces de ver que funcionalidades eran útiles para la aplicación del profesor y que funcionalidades no tenían sentido migrarlas. Por lo tanto, cada análisis finaliza concluyendo que funcionalidades se van a implementar en la aplicación del profesor.

El sexto capítulo es la explicación de uno de los juegos que se han migrado a la aplicación del profesor, en este caso es el juego de competición que se explicara posteriormente. Donde

se explica en que consiste el juego, que modalidades tiene, las funcionalidades que se han implementado y los retos más destacados.

El séptimo capítulo es también la explicación de uno de los juegos que se han migrado a la aplicación del profesor, en este caso es el juego de control de trabajo. Donde se explica en que consiste el juego, las funcionalidades que se han implementado y los retos más destacados.

El octavo capítulo es también la explicación de uno de los juegos que se han migrado a la aplicación del profesor, en este caso es el juego de cuestionario de satisfacción. Donde se explica en que consiste el juego, las funcionalidades que se han implementado y los retos más destacados.

El noveno capítulo es el análisis y la explicación de la mejora del diseño general de la aplicación del profesor. Donde se analizan algunos de los elementos que hemos optado que necesitaban algún cambio estético.

Y el último capítulo, el décimo, trata de los retos más destacados que han surgido durante todo el transcurso del proyecto. Se explica la problemática que surgió, las propuestas que se hicieron y la solución final que se puso al problema.

1.3. Equipo de trabajo

Este proyecto se ha realizado por un equipo formado por dos personas, donde los miembros somos Suhail Chaib Akhdim y yo, Sumaya Benali Laarif.

Para la organización del equipo se ha utilizado la metodología SCRUM y la herramienta Trello, para cumplir con los tiempos establecidos de cada una de las tareas. Estos conceptos de organización, se explicarán de forma más detallada en el capítulo 3.

Debido a que varias tareas las hemos trabajado de forma conjunta, ambas memorias tienen apartados muy parecidos, y se difieren en las tareas que se han hecho de forma independiente. Por lo tanto, se irá dejando claro a lo largo de la lectura las diferencias y similitudes que hay en ambas memorias.

1.4. ¿Qué he hecho en este TFG?

En este proyecto hemos rediseñado y mejorado la aplicación del profesor, desde la cual puede realizar varias tareas de gestión del ecosistema de Classpip. En particular, hemos trasladado funcionalidades, que se especificarán más adelante, de cinco juegos existentes en el Dashboard, también hemos mejorado la mecánica del envío de correos electrónicos y hemos realizado una revisión y arreglos del diseño de la aplicación del profesor.

A continuación, explicaré que tareas se han hecho conjuntamente y que tareas he realizado de forma independiente a mi compañero de forma resumida e introductoria.

Los primeros tres capítulos son muy parecidos en algunos apartados en ambas memorias, ya que es la introducción del proyecto en general y el plan de trabajo planeado. Los apartados que son iguales se redactaron de forma conjunta en una de las primeras fases del proyecto, y se difieren en los apartados más personales que se redactaron en la fase final del proyecto.

Las tareas previas que se implementaron del capítulo cuatro, son diferentes en cada una de las memorias, porque cada miembro del equipo hizo las tareas que eligió y le correspondían implementar.

El análisis de cada uno de los juegos del capítulo cinco, fue una tarea que se hizo de forma conjunta y también se redactó de esa manera, mientras se estaban realizando las pruebas pertinentes en cada uno de los juegos. Era necesario hacerlo en equipo, ya que las conclusiones después de cada análisis era una decisión a tomar en conjunto. Y también, era importante que cada miembro del grupo tuviera muy claro el funcionamiento de cada uno de los juegos, para poder realizar una migración o implementación con conocimiento.

En el capítulo seis, donde se encuentra el juego de competición, los primeros dos apartados donde se explica en que consiste el juego y las diferentes modalidades que tiene, son apartados comunes en ambas memorias. Esta información se escribió a partir de la información que se obtuvo en el análisis previo de forma conjunta, y era necesario tenerlo en ambas memorias para entender los siguientes apartados diferenciados que tiene cada uno. Los siguientes apartados, que son diferentes en ambas memorias, son las funcionalidades que ha hecho cada uno. Por lo tanto, en mi memoria solo estarán las funcionalidades que he implementado según las tareas que me correspondían y en la memoria de mi compañero se encuentran el resto de funcionalidades implementadas para este juego.

Los capítulos siete y ocho son totalmente diferentes en cada memoria, puesto que yo explico en estos capítulos dos de los juegos que he implementado de manera independiente, y mi compañero explica otros dos juegos en su memoria que ha implementado él. En mi caso, en estos capítulos se hablará del juego de control de trabajo y el juego de cuestionario de satisfacción.

El diseño general de la aplicación, que se comenta en el capítulo nueve, es igual en ambas memorias. Fue una tarea que analizamos ambos y fuimos aplicando lo que se comentó respecto al diseño mientras se iban implementando los juegos sobre la marcha. Por lo tanto, la información que se encuentra en este capítulo, nos sirvió de guía para realizar de forma homogénea el diseño en cada uno de los juegos.

Y finalmente, el capítulo diez, donde se habla de los retos más destacados. Este capítulo es totalmente diferente en cada una de las memorias, ya que cada uno de nosotros tuvo unos retos u otros. Por lo tanto, es un capítulo más personal, son retos independientes al trabajo en equipo.

1.5. Motivación personal

En el momento en que se empezó a introducir la programación en las diferentes asignaturas de la universidad, supe que iba a ser una de las cosas que más me disfrutaría en un futuro. Una vez realicé la asignatura de “Ingeniería de las aplicaciones” (EA), tuve claro que quería realizar el TFG con algo relacionado con lo que aprendí en esa asignatura. Cuando supe de la existencia de Classpip como opción a realizar como TFG y leí la descripción detallada del proyecto, tuve muy claro que era lo que realmente quería hacer. Las razones fueron varias:

- Que fuera un proyecto de desarrollo de aplicaciones, fue interesante. Sabía que podía aprender más sobre este campo de la programación y que iba a disfrutar haciendo el proyecto.
- Las tecnologías que se usaban en Classpip. Como Ionic, MongoDB o Nodejs, eran tecnologías que había trabajado con ellas previamente en la asignatura de EA, y me sentía cómoda con ellas. Por lo tanto, tenía la posibilidad de aprender mucho más sobre estas tecnologías trabajando en un proyecto como Classpip, y mejorar mis habilidades como desarrolladora.
- Tener la posibilidad de aplicar lo que estaba aprendiendo en las prácticas de empresa, ya que también era relacionado con el desarrollo de páginas web y aplicaciones, a Classpip o viceversa. Y realmente, una vez finalizado el proyecto, me ha servido trabajar en ambos proyectos a la vez, puesto que aplicaba conocimientos o habilidades que obtenía tanto en un proyecto como en otro.
- Tener la posibilidad de compartir el proyecto con un compañero de la universidad, también fue un punto que me hizo decidirme sobre Classpip como TFG. El trabajo en equipo siempre me ha parecido una habilidad importante de cara al mundo laboral, y Classpip permitía tener de alguna manera esa conexión con la realidad. La colaboración en este proyecto, también era un punto importante, y mejorar en esa habilidad también era uno de mis objetivos.
- El proyecto era algo que iba a ayudar, tanto a alumnos como a docentes, y de alguna manera ya se estaba haciendo uso de la plataforma como prueba en algunas clases en la universidad. Era emocionante pensar que la aportación que ibas a hacer en este proyecto, iba a ayudar a hacer crecer un proyecto como el que es Classpip, y que en algún momento sería algo que pudiera ser útil para la decencia.

En general, por estas razones y algunas más, Classpip fue la elección que veía más afín a mí y a mis objetivos de futuro. Por eso, no tuve dudas en elegir este proyecto una vez leí de que trataba.

CAPÍTULO 2. INTRODUCCIÓN A CLASSPIP

En este capítulo vamos a hablar sobre la gamificación y veremos en qué consiste, también analizaremos en detalle la arquitectura que tiene el proyecto Classpip, tanto las herramientas que se utilizan como las plataformas que se han implementado. A continuación, se describirán brevemente las funcionalidades que tiene el ecosistema, y finalmente, se mencionarán qué herramientas del ecosistema hemos utilizado en nuestro proyecto y por qué.

Este apartado es bastante parecido en ambas memorias, aunque con algunas diferencias. La mayor parte del capítulo se trabajó en equipo en las primeras fases del proyecto, ya que mucha información de la que se presenta, era importante entenderla al principio del proyecto por parte de todos los miembros del grupo.

2.1. ¿Qué es la gamificación?

La gamificación es un método educativo que acerca la mecánica de los juegos al ámbito educativo-profesional para obtener mejores resultados: sirve para trabajar y adquirir algunos conocimientos, mejorar ciertas habilidades y para obtener recompensas por acciones concretas. Es un término que se ha vuelto muy popular en los últimos años, especialmente en el entorno educativo y digital.

Otra de sus ventajas es que consigue sus objetivos en el campo de la educación y la formación, fomentando un entorno más alegre y positivo para los alumnos y haciendo más amena la realización de las actividades educativas.

2.2 Arquitectura del proyecto

Classpip se ha dividido en cinco aplicaciones diferentes, que juntas forman el ecosistema de la herramienta. Cada tipo tiene las características que se describen a continuación:

Server: Indispensable para gestionar notificaciones en tiempo real entre distintas aplicaciones. Esto es posible gracias al uso de Sockets, un método de comunicación entre el servidor y el cliente. El servidor es parte del backend y fue desarrollado usando ExpressJS.

API-REST: La aplicación también forma parte del backend, desarrollado utilizando el framework LoopBack, que almacena la base de datos del sistema y todas las rutas o puntos finales a través de los cuales se realizan las operaciones CRUD en el registro. Estos registros se almacenan en un archivo JSON.

Dashboard: Es una herramienta de gestión de profesores que te permite realizar todo tipo de acciones para prepararte las clases donde se decida utilizar Classpip. Gracias a la aplicación web, los profesores configuran todo tipo de juegos, dan de alta a los alumnos y

gestionan sus grupos de trabajo. Esta aplicación es parte de la interfaz de usuario de la plataforma y está desarrollada usando Angular y TypeScript.

Aplicación móvil para docente: El docente tiene acceso a una aplicación móvil que le permite realizar acciones rápidas sin tener que ir al tablero. El uso de la aplicación móvil facilita acciones como tomar turnos, asignar puntos o administrar un juego que no requiere mucha molestia. Esta aplicación es parte de la interfaz de usuario de la plataforma y está desarrollada utilizando Ionic y TypeScript.

Aplicación móvil para el alumnado: Esta es la aplicación en la que el alumno juega los juegos que el profesor preparó para él. Además, permite a los alumnos modificar los datos de su perfil y obtener un historial de todos los juegos en los que han participado, estén activos o no. Esta aplicación es parte de la interfaz de usuario de la plataforma y está desarrollada utilizando Ionic y TypeScript.

Para desarrollar las diferentes partes del ecosistema hemos utilizado las siguientes herramientas:

Angular: Framework Open Source, permite crear aplicaciones web de una manera sencilla y organizada.

Ionic: Framework Open Source basado en Angular y Apache Cordova nos ha permitido desarrollar la interfaz gráfica tanto del móvil del estudiante como el del profesor.

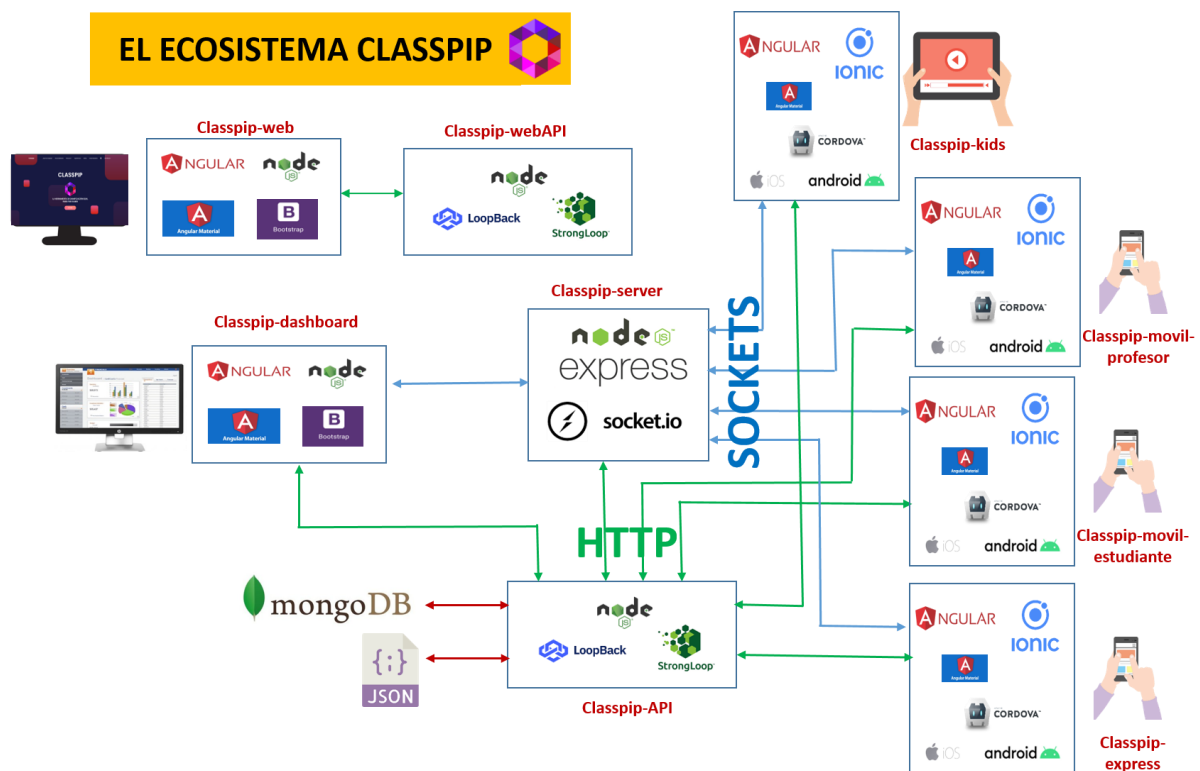


Fig.1. El ecosistema de Classpip

2.3 Funcionalidades del ecosistema

Classpip es una herramienta para crear escenarios de gamificación de una manera rápida, a continuación, se explicarán de una manera breve los diferentes juegos que ofrece esta herramienta que hemos analizado.

1. **Juego de competición:** Es un juego en el cual se involucra a los alumnos de una clase en una dinámica competitiva en la cual hay un único vencedor. De esta manera, la actividad llevada a cabo se hace más amena y los alumnos sienten que tienen la responsabilidad de participar activamente, si quieren realmente quedar en una buena posición en frente de sus otros compañeros y poder así obtener ese premio.

Existen tres modalidades: Liga, Fórmula Uno y torneo. En próximos capítulos de esta memoria se explicará el funcionamiento de estos juegos en detalle.



Fig.2. Una de las pantallas del juego de competición desde la aplicación del alumno

2. **Juego de control de trabajo:** El juego de control de trabajo consiste en una herramienta del profesor para la supervisión periódica y evolución del trabajo de los alumnos. Donde el profesor, a partir de un proyecto que se esté realizando en clase, puede dividirlo por partes para controlar el progreso de los alumnos que conformen ese grupo de trabajo.



Fig.3. Una de las pantallas del juego de control de trabajo desde la aplicación del alumno

3. **Juego de cuestionario de satisfacción:** El juego de cuestionario de satisfacción consiste en una herramienta del profesor para conocer la opinión de los alumnos respecto a un tema que el docente decida. En este caso, va enfocado a una opinión relacionada con la satisfacción que sienten los alumnos respecto a un tema relacionado con las clases, metodología del docente, interés sobre una asignatura o tema en concreto.

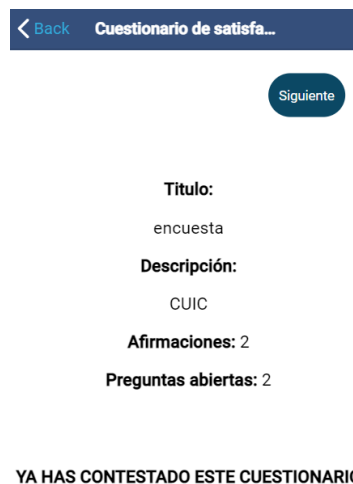
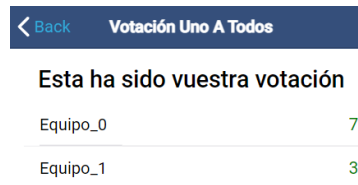


Fig.4. Una de las pantallas del juego de cuestionario de satisfacción desde la aplicación del alumno

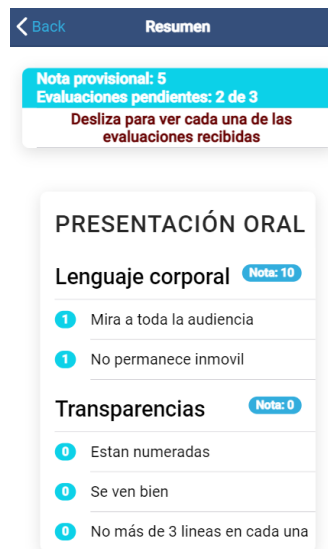
4. **Juego de votación:** El juego de votaciones es un juego en el cual se involucra a los alumnos de una clase en una dinámica, en la cual hay que tomar decisiones para elegir cuál es la mejor opción a tener en cuenta de entre la que hay. Los alumnos en ese caso pueden llegar a votarse entre sí o pueden votar opciones establecidas por el profesor.



Esta ha sido vuestra votación	
Equipo_0	7
Equipo_1	3

Fig.5. Una de las pantallas del juego de votación desde la aplicación del alumno

5. **Juego de evaluación:** El juego de evaluación es un juego en el cual se involucra a los alumnos de una clase en un ambiente exigente en el cual cada uno de los participantes será juzgado y puntuado por los demás alumnos. De esta manera, la actividad llevada a cabo se hace más estricta, pero a la vez más amigable. Serán evaluaciones cruzadas realizadas entre los alumnos.



PRESENTACIÓN ORAL	
Lenguaje corporal	Nota: 10
1	Mira a toda la audiencia
1	No permanece inmóvil
Transparencias	Nota: 0
0	Están numeradas
0	Se ven bien
0	No más de 3 líneas en cada una

Fig.6. Una de las pantallas del juego de evaluación desde la aplicación del alumno

Los juegos mencionados arriba son los juegos que hemos trabajado en este proyecto y los que hemos analizado en profundidad, pero también existen otros juegos que ofrece el ecosistema como se observa en la siguiente imagen.



Fig.7. Todos los juegos que ofrece el ecosistema

2.4 ¿Qué herramientas del ecosistema hemos utilizado en nuestro proyecto?

En este apartado, se explicarán que herramientas que ya tenía Classpip existentes en su ecosistema y las que se han utilizado para la implementación de nuestro proyecto. En nuestro caso, hemos utilizado las siguientes:

1. **Aplicación móvil del profesor** (Desarrollada en Ionic): Como el objetivo de nuestro proyecto era implementar cinco juegos en la aplicación del profesor y mejorar el diseño general también, era esencial utilizar esta parte del ecosistema. Básicamente, porque es la parte en la que más hemos trabajado, añadido y modificado cosas. Y es donde se encuentra la mayoría del trabajo realizado.
2. **Dashboard** (Página web, desarrollada en Angular): La utilización del Dashboard ha sido esencial para llevar a cabo el análisis de los juegos desde el punto de vista del profesor y poder decidir que funcionalidades se iban a mantener. Por lo tanto, ha sido un elemento que se ha usado bastante en las primeras fases del proyecto.
3. **Aplicación móvil del alumno** (Desarrollada en Ionic): Este elemento también ha sido esencial para llevar a cabo el análisis de los juegos desde el punto de vista del alumno. Analizar las funcionalidades desde este elemento, nos hizo entender mejor el funcionamiento de los juegos, ya que podíamos experimentar la experiencia del usuario del alumno. Por lo tanto, ha sido un elemento que se ha usado bastante en las primeras fases del proyecto.

4. **Base de datos** (En formato API): Entender el funcionamiento de la API, nos facilitó entender como realmente se hacían las consultas. Para poder crear las que necesitaríamos en las funcionalidades nuevas que hemos añadido en el proyecto. Ha sido un elemento que hemos utilizado a lo largo de todo el proyecto, ya que el móvil del profesor se conectaba constantemente a la API para recibir toda la información necesaria a mostrar.
5. **Server**: El servidor ha sido un elemento muy importante para la mejora de la mecánica del envío de correos electrónicos de la aplicación del profesor. La mayor parte de la configuración y programación de esta funcionalidad, se ha trabajado en el servidor utilizando la herramienta de Nodemailer. Aparte, de ser esencial para esta funcionalidad, también ha sido un elemento que se ha utilizado durante todo el proyecto, ya que el móvil del profesor se conectaba al servidor en ocasiones.

CAPÍTULO 3: OBJETIVOS Y PLAN DE TRABAJO

En este capítulo se presentarán los objetivos que se han asumido desde el inicio del proyecto por parte de sus dos participantes, terminando así de definirse con exactitud gracias a la ayuda de los tutores. Este capítulo es igual en ambas memorias, ya que es la explicación de los objetivos que teníamos como equipo y la metodología de trabajo que hemos utilizado para dividirnos las tareas entre los miembros del grupo. Por lo tanto, es información que se ha decidido y se ha escrito en conjunto.

Los objetivos en cuestión se definen a continuación:

Se necesitará en primer lugar definir una metodología de trabajo a seguir a lo largo de todo el proyecto. Como objetivo principal se quiere implementar un total de 5 juegos diferentes en la aplicación del profesor. Como siguientes pasos a tomar, se requerirá de añadir una nueva funcionalidad que no esté implementada en los juegos existentes y que los juegos que se vayan a implementar tengan un código optimizado con respecto al código del propio Dashboard. En último lugar, no solo se tiene como objetivo testear todos los juegos para darlos por funcionales, sino que también se requiere que todos los juegos tengan una unificación en cuanto a diseño.

Estos objetivos, que se han definido de forma genérica, se desglosarán en subcategorías y funcionalidades concretas en siguientes apartados para su posterior implantación y/o mejora.

Antes de establecer un método de trabajo necesitamos familiarizarnos con el entorno y el código, por ello se nos asignaron unas pequeñas tareas a realizar, con ellas pudimos concretar más la estructura de la aplicación y del propio código ya implementado. En el capítulo siguiente podremos en mayor detalle qué tareas fueron y cómo se desarrollaron. Una vez realizada la toma de contacto, establecimos el método de trabajo que queríamos seguir para este proyecto, dada su estructura y envergadura.

Una de las metodologías de trabajo más comunes en equipos de desarrolladores es la metodología Scrum que es la metodología que hemos escogido para este proyecto. Y para poder organizar las tareas que se asignaran se ha utilizado la herramienta Trello.

3.1. Metodología Scrum

El método Scrum es un proceso de trabajo en equipo en el que se aplican un buen conjunto de técnicas para maximizar la colaboración óptima entre los distintos miembros que forman parte del equipo. Este tipo de formación parte de grupos de trabajo de alto rendimiento.

El funcionamiento de Scrum se basa en el cumplimiento parcial y la entrega regular del producto final. Esto la convierte en una metodología especialmente adecuada para proyectos implementados en entornos complejos. De esta manera, permite alcanzar resultados en un corto período de tiempo en el que priman la innovación, la competitividad, la flexibilidad y la productividad.

Los proyectos Scrum se implementan en ciclos cortos y fijos. En nuestro caso, el tiempo establecido fue de 2 semanas, aunque los tiempos de ciclo se ajustaron según los requisitos del proyecto. Al final de cada iteración o sprint, debe haber un resultado completo que contribuya al aumento del producto final, de modo que se pueda presentar un desarrollo sólido del proyecto en cada reunión con el tutor.

Este proceso implica hacer una lista de objetivos que servirán como hoja de ruta, y cada objetivo debe dividirse en varias tareas si es necesario. Para cada tarea, se asigna la prioridad correspondiente. Basándonos en esto, se elabora un cronograma con los resultados que se lograrán en cada sprint.

Al hablar de Scrum, debemos tener en cuenta una figura imprescindible en proyectos donde se aplica esta metodología: el Scrum Master. 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.

Cuando se trata de Scrum, es necesario tener una figura que se involucre activamente en la organización de las tareas, esa figura es la denominada “Scrum Master”.

Esta figura es responsable de que se aplique correctamente cada sprint, de asignar nuevas tareas a los diferentes miembros del equipo y de resolver los conflictos que puedan surgir entre ellos. Para este proyecto el Scrum Master elegido, ha sido Sumaya Benali Laarif.

A continuación, en las siguientes páginas se muestra en detalle cómo fue la división de los sprints, las tareas que se asignaron a cada sprint con la descripción correspondiente, y quien realizó dichas tareas con las correspondientes horas dedicadas de cada miembro del grupo.

Sprint	Fecha	Tareas	Descripciones	Horas realizadas	
				Suhail	Sumaya
1	17/02 - 24/02	Instalar entorno de trabajo	Descargar las 5 aplicaciones del repositorio principal de GitHub	3	3
			Instalar dependencias		
			Abrir las aplicaciones		
			Crear repositorios propios en GitHub		
			Enlazar ambos repositorios		
		Testear el entorno	Crear juegos desde cero mediante el dashboard	3	3
			Probar juegos desde el móvil del alumno	2	2
			Probar aplicación del profesor	1	1
			Creación de una lista con items a mejorar del test realizado	2	2
2	24/02 - 03/03	Pasos previos	Arreglar formulario del Register en el móvil del profesor	N/A	8
			Arreglar taps y sidebar en el móvil del profesor	7	N/A
3	03/03 - 17/03	Análisis de los juegos en detalle	Análisis de la creación de los juegos en el Dashboard	6	6
			Análisis de las funcionalidades implementadas en el Dashboard		
			Análisis de las funcionalidades implementadas en el móvil del alumno		
			Toma de decisión: Funciones a añadir en el móvil del profesor	1	1

Sprint	Fecha	Tareas	Descripciones	Horas realizadas	
				Suhail	Sumaya
4	17/03 - 31/03	Diseño de los juegos	Toma de decisión: Elementos que se deben incluir	6	6
			Creación de un Borrador	3	3
5	31/03 - 21/04	Juego de Competición	Implementar página principal	5	5
			Funcionalidad: Ver Jornadas y Resultados generales	18	N/A
			Funcionalidad: Desactivar el juego	N/A	6
			Funcionalidad: Editar Jornadas	N/A	12
			Funcionalidad: Editar puntos	N/A	12
			Funcionalidad: Asignar ganadores	30	N/A
6	21/04 - 12/05	Juego de Evaluación	Funcionalidad: Visualizar Rubrica	20	N/A
			Funcionalidad: Editar evaluaciones	12	N/A
			Funcionalidad: Borrar evaluaciones	12	N/A
		Juego Control de trabajo	Funcionalidad: Visualizar Tabla principal	N/A	18
			Funcionalidad: Envió de email	N/A	40
		Problemática Node	Compatibilidad versiones, arreglo con NVM	N/A	35

Sprint	Fecha	Tareas	Descripciones	Horas realizadas	
				Suhail	Sumaya
7	12/05 - 02/06	Juego de Votaciones	Funcionalidad: Visualizar Tabla principal	18	N/A
		Juego Cuestionario de satisfacción	Funcionalidad: Visualizar Tabla principal	N/A	12
8	02/06 - 23/06	Diseño genérico	Aplicar diseño a nivel de código	6	6
			Unificar diseño para todos los elementos	6	6
9	23/06 - 07/06	Redacción de la memoria	Redacción de los juegos y otros apartados	20	20
		Preparación presentación	Diseño y redacción de las transparencias	6	6

3.2. Herramienta Trello

Para organizar todas las tareas en un solo lugar y que sean accesibles para todos los miembros del equipo, las herramientas que se utilizan proporcionan paneles donde se ubican todas las tareas y subtareas.

Trello es una aplicación diseñada para organizar tareas en grupos de trabajo. Esta herramienta nos proporciona una tabla en la que podemos añadir listas y etiquetas, permitiéndonos organizar las tareas de nuestro proyecto de forma fácil, rápida y cómoda. Los diferentes elementos que forman parte de Trello son:

Lista de tareas: Son todas aquellas actividades o ejercicios que se acuerdan con el equipo, que se completarán para un sprint determinado.

En proceso: Son aquellas tareas que ya han sido asignadas a uno o varios miembros del grupo y ya están siendo desarrolladas y desplegadas.

Hecho: Son aquellas tareas que ya han sido finalizadas y testeadas y por ello se dan por acabadas.

En nuestro caso particular, no hemos querido solamente introducir los tres elementos básicos, sino que también una vez finalizadas todas las tareas de un mismo sprint, creábamos una categoría nueva con el número del sprint y pasábamos ahí las tareas que formaban parte de ella. De esta manera, era más fácil a simple vista detectar el progreso del proyecto en cada momento.

A continuación, podemos ver una imagen, del Trello en acción. Se pueden ver hasta dos sprints finalizados y el tercer sprint dividido entre tareas a realizar, tareas en curso y tareas hechas. También se puede ver a quien o quienes ha sido asignada cada tarea, por ello consideramos que era una de las mejores herramientas de organización que podíamos utilizar.

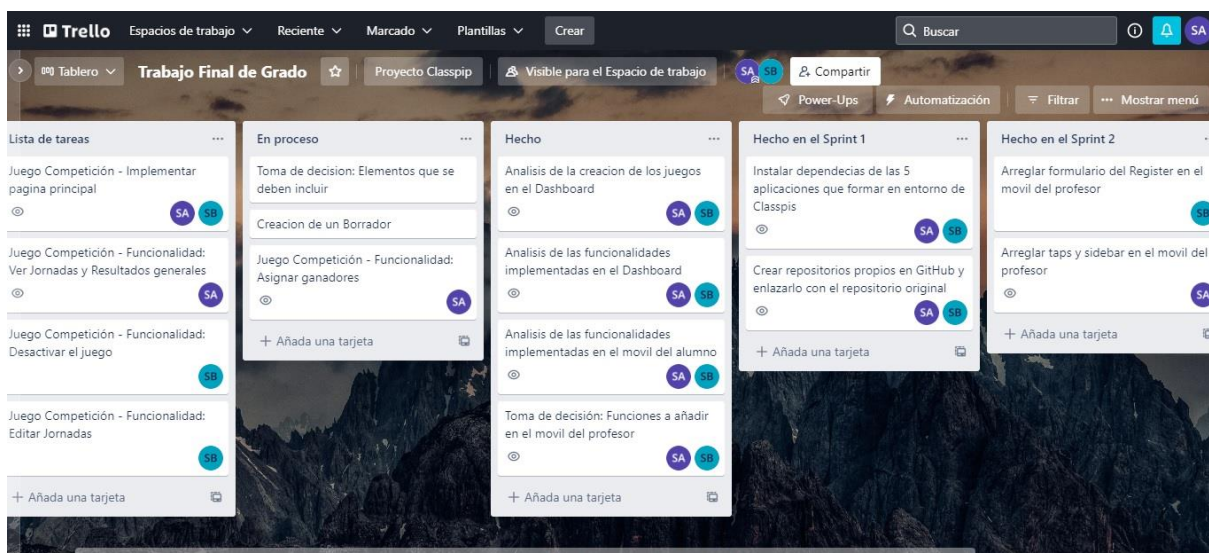


Fig.8. Captura de la herramienta Trello donde se muestra la organización de las tareas

CAPÍTULO 4: TAREAS PREVIAS

En este capítulo se van a explicar las tareas previas que se hicieron antes de empezar a trabajar en el proyecto real. Estas tareas se llevaron a cabo, con el objetivo de familiarizarnos con el entorno en la primera fase del proyecto.

Este capítulo es diferente en comparativa a la memoria de mi compañero de trabajo, puesto que ha sido algo que hemos hecho de forma independiente. Cada uno tiene las tareas previas que ha realizado explicadas en su memoria.

4.1 Pagina de registro

La página de registro del profesor fue una de las tareas finalmente elegidas como tarea previa, para practicar y conocer el entorno en el que íbamos a trabajar posteriormente.

La siguiente imagen se puede observar el estado en el que estaba la página de registro antes de cualquier tipo de arreglo o mejora. Como se puede observar hay bastantes cosas que se podrían mejorar:

1. La posición del formulario: Cuando la página se renderizaba en formato móvil, el formulario no estaba del todo bien centrado. Como se puede observar en la imagen de abajo, el formulario está demasiado hacia abajo e incluso el botón de 'Volver al login' se corta por la mala posición. Este fue uno de los puntos a arreglar, el objetivo era centrar el formulario y colocar los botones de manera correcta.
2. El nombre de cada campo: Como se muestra en la imagen, cada campo tiene un nombre para indicar al usuario el tipo de información que se espera, por ejemplo, el campo 'nombre', 'primer apellido' o 'email'.
3. Validaciones: El formulario no tenía mensajes que avisaran al usuario sobre si un campo es obligatorio o si algún campo no está en el formato correcto. Incluso si realmente algún campo no estaba bien, al clicar sobre el botón de registro daba la sensación de que no funcionaba, puesto que no hacía nada. Y también en ocasiones, si el formato en el que se rellenaba el campo no era correcto, se registraba correctamente con esos campos erróneos. Esto puede ser un poco confuso respecto a la experiencia del usuario, Por eso era necesario implementar las validaciones y ejecutarlas al clicar el botón de 'Registrar'.
4. Iconos: Dentro de cada input se puede observar un icono a la derecha. A primera vista no tienen mucho sentido, tampoco son necesario y no cumplen ninguna función. Por eso fue algo que se decidió quitar y mantener un estilo más elegante.



Fig. 9. Página del registro antes de ser modificada

Después de hacer algunos cambios en el código, tanto de forma funcional como de forma estética, obtenemos un resultado como el que se muestra en la siguiente imagen.

Se puede observar que la posición del formulario que comentamos anteriormente se ha arreglado, centrando el formulario en la parte central de la pantalla. Esto se ha conseguido editando los márgenes de la pantalla.

El siguiente objetivo a arreglar, era sacar fuera el nombre de los campos, puesto que previamente, al rellenar el campo con la información necesaria, desaparecía la información de que era el campo a rellenar. Por ejemplo, en la figura 9, al rellenar el campo de nombre de usuario con “talleres”, desaparece la información de que ese campo corresponde al nombre de usuario. Por lo tanto, el usuario en ese momento no sabe a qué corresponde ese campo una vez rellenado, y eso no era una buena experiencia del usuario. Como vemos, en la figura 10, ahora la información de los campos es visible fuera de donde se escriben los datos, y aun rellenando los campos la información sigue siendo visible.

También se han dejado solo los iconos más relevantes, como se muestran en la figura 10, a diferencia de la cantidad de iconos que teníamos previamente, como se muestra en la figura 9.



Nombre

Primer apellido

Segundo apellido

Email

Username

Contraseña

Repetir contraseña

REGISTRAR

VOLVER A LOGIN

Fig. 10. Página del registro después de las modificaciones

Y finalmente, se han implementado las validaciones, puesto que previamente no mostraba ningún tipo de mensaje, ni indicaba si un campo era correcto o no. Por lo tanto, como vemos en la figura 11, si el campo del email se rellena de forma errónea, tenemos una línea roja que indica que ese campo no se ha rellenado de forma correcta. La figura 12, vemos como si rellenamos el campo correctamente, la raya será de color verde, indicando que el formato es correcto. Y en la figura 13, si tenemos el campo con un mal formato y aun así clicamos sobre el botón de 'Registrar', saldrá un mensaje por debajo del campo indicando cuál es el error y la razón por la que no es válido.

Por lo tanto, con todos estos cambios que se han realizado, se ha mejorado la página de registro, tanto estéticamente, como de manera funcional.

Nombre
Sumaya

Primer apellido
Benali

Segundo apellido
Laarif

Email
sumaya-hotmail.com

Username
sumayabl

Contraseña
.....

Repetir contraseña
.....

REGISTRAR

VOLVER A LOGIN

Fig. 11. Campo de email con el formato incorrecto

Nombre
Sumaya

Primer apellido
Benali

Segundo apellido
Laarif

Email
sumaya@hotmail.com

Username
sumayabl

Contraseña
.....

Repetir contraseña
.....

REGISTRAR

VOLVER A LOGIN

Fig. 12. Campo de email con el formato correcto

Nombre
Sumaya

Primer apellido
Benali

Segundo apellido
Laarif

Email
sumaya-hotmail.com

Introduce un email válido.

Username
sumayabl

Contraseña
.....

Repetir contraseña
.....

REGISTRAR

VOLVER A LOGIN

Fig. 13. Campo de email con el formato incorrecto, clicando sobre el botón de registro

CAPÍTULO 5: ANÁLISIS DE LOS JUEGOS A IMPLEMENTAR

En este capítulo se va a explicar de forma detallada el análisis de cada uno de los juegos que vamos a implementar en nuestro TFG. Los juegos que se van a analizar son los siguientes: El juego de competición, el juego de evaluación, el juego de control de trabajo, el juego de votaciones y el juego de cuestionario de satisfacción. Cada uno de estos juegos se van a analizar siguiendo el mismo patrón, empezando por la descripción de cómo se crea el juego. Siguiendo por cómo se visualiza el juego desde el punto de vista del profesor y desde el punto de vista del alumno, indicando que funcionalidades existen. Y se finaliza el análisis de cada uno de los juegos con una conclusión de qué funcionalidades se migrarán al móvil del profesor.

Este capítulo es uno de los más importantes de la memoria, ya que, gracias al análisis realizado, pudimos crear conclusiones de qué funcionalidades existentes eran útiles para la migración, y que funcionalidades teníamos que crear de cero para la aplicación móvil del profesor.

Este capítulo es conjunto, y se encuentra en ambas memorias, puesto que el análisis fue algo que se trabajó en equipo para poder tomar las decisiones en conjunto, y así poder discutir qué puntos queríamos o no en nuestro proyecto.

5.1 Creación del grupo

El primer paso para empezar el análisis, fue abrir el Dashboard del profesor y entender cómo se realizaba la creación del juego desde cero. Por lo tanto, la siguiente información indica qué pasos se han seguido para crear el juego.

El paso principal en el Dashboard, fue crear un grupo (se considera que el grupo es el conjunto de alumnos de la clase que dirige el profesor) asignándole un nombre, una descripción y se seleccionan los alumnos que participarán en este grupo.

Este paso es importante, ya que sin un grupo no se pueden crear los juegos pertinentes, por lo tanto, el primer paso es siempre crear el grupo al que queremos crear los juegos que el profesor considere. Una vez creado el grupo, pasamos a la creación y análisis de cada uno de los juegos.

5.2. Juego de competición

Para crear el juego de competición, nos dirigimos al grupo al que lo queremos crear, indicamos que queremos crear un juego e indicamos que queremos que sea de tipo 'Juego de competición'.

Se abrirá una pestaña donde se tiene que asignar un nombre para el juego, se elige el modo individual y el tipo de competición (Liga, Fórmula Uno, Torneo). Como tenemos diferentes tipos de competición, en los siguientes apartados se explicará el análisis de cada uno de ellos.

5.2.1. Liga

Empezamos por el tipo de juego de Liga, que es una competición en la que se enfrentan entre sí todos los equipos o participantes de una misma categoría, siendo el vencedor el que obtiene mayor número de puntos. Todos los participantes juegan en todas las jornadas existentes.

5.2.1.1 Punto de vista del profesor

Al elegir la competición de tipo liga, mientras se está creando el juego, se debe asignar el número de jornadas que se jugarán. Este es el último paso para la creación del juego, por lo tanto, ahora el juego ya está creado en el apartado de juegos activos.

Una vez creada la liga, el profesor desde el Dashboard, desde juegos activos, puede seleccionar lo que ha creado previamente. Una vez seleccionado el juego de competición deseado, el profesor tiene la posibilidad de ver los alumnos que participan en el juego, la posición en la que se encuentran actualmente y los datos estadísticos del juego (Partidas totales, jugados, ganados, empatados, perdidos y puntos totales).

Las funcionalidades que el profesor puede hacer desde el Dashboard son:

- Ver la información de cada jornada en detalle
- Editar jornadas (Seleccionando la fecha y el criterio ganador).
- Asignar los ganadores o empates (de forma manual, aleatoria o a partir de los resultados de un juego) de cada jornada.
- Desactivar el juego, es decir, que el juego pase de activo a inactivo.

5.2.1.2. Punto de vista del alumno

El alumno desde la aplicación solo tiene la posibilidad de ver estadísticas, las suyas y la de sus compañeros. Puede ver el ranking de los jugadores de su grupo en función de los puntos obtenidos, y de cada uno de ellos puede ver los datos estadísticos (Partidas jugadas, ganadas, empatadas y pérdidas). También tiene la posibilidad de ver el ganador de cada una de las jornadas.

5.2.1.3. Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Ver la información de cada jornada en detalle
- Editar jornadas (Seleccionando la fecha y el criterio ganador).
- Asignar los ganadores o empates (de forma manual, aleatoria o a partir de los resultados de un juego) de cada jornada.
- Desactivar el juego, es decir, que el juego pase de activo a inactivo.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, añadimos algunas funcionalidades más:

- Ver progreso (Visión global de la evolución de la competición, según se vayan produciendo las jornadas)
- Creación de una página de inicio (Visión global de todas las herramientas o funcionalidades que tenga este modo)

5.2.2. Fórmula Uno

El tipo de juego Fórmula Uno es una competición en la que se enfrentan entre sí todos los equipos o participantes de una misma categoría, con un número limitado de vencedores, descartando así el resto. Cada vencedor según su posición se lleva una puntuación predeterminada. Todos los participantes juegan en todas las jornadas existentes y pueden acumular puntos.

5.2.2.1. Punto de vista del profesor

Al elegir la competición de tipo Fórmula Uno, se debe asignar el número de jornadas que se jugarán. Y posteriormente, el profesor deberá elegir cuantas posiciones finales deberá haber y los puntos totales que se deben repartir entre ellos.

Una vez creado el juego de competición “Fórmula Uno”, el profesor desde el Dashboard, desde juegos activos, puede seleccionar lo que ha creado previamente. Una vez seleccionado el juego de competición deseado, el profesor tiene la posibilidad de ver los alumnos que participan en el juego, la posición en la que se encuentran actualmente y los puntos totales.

Las funcionalidades que el profesor puede hacer son:

- Ver la información de cada jornada en detalle
- Editar jornadas (Seleccionando la fecha y el criterio ganador) y editar puntuación (Editar los puntos de cada posición si se desea).
- Asignar los ganadores (de forma manual, aleatoria o a partir de los resultados de un juego) de cada jornada.
- Desactivar el juego, es decir, que el juego pase de activo a inactivo.

5.2.2.2. Punto de vista del alumno

El alumno desde la aplicación solo tiene la posibilidad de ver estadísticas, las suyas y la de sus compañeros. Puede ver el ranking de los jugadores de su grupo en función de los puntos obtenidos. También tiene la posibilidad de ver los ganadores de cada una de las jornadas y los puntos asignados.

5.2.2.3. Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Ver la información de cada jornada en detalle
- Editar jornadas (Seleccionando la fecha y el criterio ganador) y editar puntuación (Editar los puntos de cada posición si se desea).
- Asignar los ganadores (de forma manual, aleatoria o a partir de los resultados de un juego) de cada jornada.
- Desactivar el juego, es decir, que el juego pase de activo a inactivo.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, añadimos algunas funcionalidades más:

- Ver progreso (Visión global de la evolución de la competición, según se vayan produciendo las jornadas)
- Creación de una página de inicio (Visión global de todas las herramientas o funcionalidades que tenga este modo)

5.2.3. Torneo

El tipo de juego Torneo es una competición con una serie de juegos en los que compiten entre sí varias personas o equipos que se eliminan unos a otros progresivamente.

5.2.3.1. Punto de vista del profesor

Al elegir la competición de tipo Torneo, se debe elegir el modelo: Clásico, Doble eliminación o suizo. Y a continuación, se deben elegir las parejas que competirán entre sí en la primera jornada. En este caso el único funcional en este momento es el clásico, que es el que vamos a analizar a continuación.

Una vez creado el juego de competición "Torneo", el profesor desde el Dashboard, desde juegos activos, puede seleccionar lo que ha creado previamente. Una vez seleccionado el juego de competición deseado, el profesor tiene la posibilidad de ver los alumnos que participan en el juego, la posición en la que se encuentran actualmente en el cuadro de torneo.

Las funcionalidades que el profesor puede hacer son:

- Ver la información de cada jornada en detalle.
- Editar jornadas (Seleccionando la fecha y el criterio ganador).
- Asignar los ganadores (de forma manual, aleatoria o a partir de los resultados de un juego) de cada jornada. El número de jornadas existentes vendrá delimitado por el número de participantes.
- Desactivar el juego, es decir, que el juego pase de activo a inactivo.

5.2.3.2. Punto de vista del alumno

El alumno desde la aplicación solo tiene la posibilidad de ver los resultados del torneo. El resumen de los contrincantes y del ganador de cada fase (En modo cuadro de torneo).

5.2.3.3. Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Ver la información de cada jornada en detalle.
- Editar jornadas (Seleccionando la fecha y el criterio ganador).
- Asignar los ganadores (de forma manual, aleatoria o a partir de los resultados de un juego) de cada jornada. El número de jornadas existentes vendrá delimitado por el número de participantes.
- Desactivar el juego, es decir, que el juego pase de activo a inactivo.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, añadimos algunas funcionalidades más:

- Ver progreso (Visión global de la evolución de la competición, según se vayan produciendo las jornadas)
- Creación de una página de inicio (Visión global de todas las herramientas o funcionalidades que tenga este modo)

5.3. Evaluación

Para crear el juego de evaluación, nos dirigimos al grupo al que lo queremos crear, indicamos que queremos crear un juego e indicamos que queremos que sea de tipo 'Evaluación'. Se abrirá una pestaña donde se tiene que asignar un nombre para el juego y se seguirán una serie de pasos que se explican a continuación.

5.3.1. Crear Juego

El primer paso, es elegir el modo entre individual y el modo equipos (donde se podrá elegir si la evaluación de alumnos será individual o por equipos). A continuación, se introduce la descripción del juego y se elige entre dos formatos de evaluación: Rúbrica o preguntas abiertas.

5.3.2. Rúbrica

Las rúbricas son una herramienta que ayuda a evaluar el aprendizaje del alumnado, haciendo que los propios estudiantes también conozcan sus errores mediante la autoevaluación. Es un documento que describe distintos niveles de calidad de un proyecto, dando un feedback informativo al alumnado sobre el desarrollo de su trabajo.

En el siguiente apartado de la creación del juego basado en rúbrica, se debe seleccionar la rúbrica que se quiere utilizar. Existe la posibilidad de evaluar presentaciones orales, donde se tendrá que ajustar el peso de cada criterio (porcentaje). Y también existe la posibilidad de evaluar retos donde se tendrá que ajustar el peso de cada criterio (porcentaje).

A continuación, se debe seleccionar la forma de evaluación: 1 a N o Todos con todos.

- **1 a N:** Cuando le toca presentar a una persona, un número determinado de personas son las que evalúan la presentación. Por lo tanto, se deberá determinar el parámetro N y podrá ser de forma aleatoria o manual.
- **Todos con todos:** Cuando le toca presentar a una persona, todos los miembros del equipo evalúan esa presentación.

Una vez elegida la forma de evaluar se procede a elegir si el profesor tomará parte de la evaluación. En el caso de que sea que sí, se debe indicar si la evaluación del profesor se considerará igualitaria en cuanto al resto de evaluadores o si contará un 50% sobre la evaluación final.

A continuación, se debe elegir el criterio de puntuación, que será mediante pesos o mediante penalización. Y se debe elegir el porcentaje de los sub criterios de los criterios elegidos previamente.

En el siguiente apartado, se debe elegir la visualización de los resultados de cara a la persona evaluada. Donde la persona evaluada solo podrá ver el resultado global o podrá ver el resultado global y cada una de las evaluaciones recibidas.

5.3.3. Preguntas abiertas

Una vez seleccionada la opción de preguntas abiertas, se deberán añadir las preguntas que se consideren necesarias para la evaluación.

Los siguientes apartados de la creación de esta opción de evaluación son exactamente igual a la de la rúbrica. A partir del apartado de forma de evaluación.

5.3.4. Punto de vista del profesor

Funcionalidades a continuación:

- Desactivar el juego
- Añadir evaluación del profesor en el caso de que participe.
- Visualizar la evaluación de los participantes.
- Visualizar las coincidencias de evaluación entre evaluadores (Únicamente en la rúbrica).

5.3.5. Punto de vista del alumno

Se visualiza un resumen de las características del juego de evaluación (Modo y formato). Al alumno le saldrán los diferentes compañeros que le toca evaluar y los podrán evaluar según los criterios.

Si eres el alumno evaluado, aparecerá una nota provisional y los detalles correspondientes.

5.3.6. Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Desactivar el juego
- Añadir evaluación del profesor en el caso de que participe.
- Visualizar la evaluación de los participantes.
- Visualizar a los alumnos que tienen que evaluar.
- Eliminar la evaluación de un alumno.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, hemos decidido no añadir nada más porque lo que había ya nos parecía suficiente e interesante.

5.4. Control de trabajo en grupo

Para crear el juego de control de trabajo, nos dirigimos al grupo al que lo queremos crear, indicamos que queremos crear un juego e indicamos que queremos que sea de tipo 'Control de trabajo'. Se abrirá una pestaña donde se tiene que asignar un nombre para el juego y se seguirán una serie de pasos que se explican a continuación.

5.4.1 Crear Juego

El primer paso, es introducir el número de controles que habrá y seleccionar qué información podrá visualizar el alumno (Cada alumno verá solo los puntos que él ha asignado y sus comentarios, o los alumnos verán los puntos y comentarios de todos sus compañeros de equipo).

5.4.2. Punto de vista del profesor

Funcionalidades a continuación:

- Desactivar el juego
- Visualización de los equipos y estado de cada uno de los controles del equipo.
- Control del progreso de los controles, con información de cada una de las valoraciones de los alumnos en detalle.

5.4.3. Punto de vista del alumno

El alumno podrá ver los miembros del equipo (incluido el mismo) y deberá repartir 10 puntos entre los miembros del equipo, donde también podrá escribir observaciones.

Una vez asignados los puntos, se visualizará un resumen de la repartición y un mensaje de si falta alguien por responder.

5.4.4. Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Desactivar el juego
- Visualización de los equipos y estado de cada uno de los controles del equipo.
- Control del progreso de los controles, con información de cada una de las valoraciones de los alumnos en detalle.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, añadimos algunas funcionalidades más:

- Mostrar la opción de envío de notificación a los alumnos que no han evaluado el control aún (envío manual).

5.5. Juego de votaciones

Para crear el juego de votaciones, nos dirigimos al grupo al que lo queremos crear, indicamos que queremos crear un juego e indicamos que queremos que sea de tipo 'Votación'. Se abrirá una pestaña donde se tiene que asignar un nombre para el juego y se seguirán una serie de pasos que se explican a continuación.

5.5.1. Crear Juego

Se crea un juego de votaciones, eligiendo el tipo entre individual o por equipos, seguidamente se elige uno de los tres modos (Uno a Todos, Todos a Uno y Votar Opciones). Como son modos distintos, se analizarán de forma independiente en los siguientes apartados.

5.5.2. Uno a todos

En esta modalidad cada participante reparte puntos a todos los demás. Cuando se crea se puede elegir que los participantes pueden votarse a sí mismos o no. Y se establece si el reparto será fijo según la posición que ocupa el participante o si el reparto será libre. En el caso de reparto fijo, se debe establecer qué posiciones si puntuaran y con cuantos puntos lo harán. Y en el caso de reparto libre se debe establecer cuándo puntos máximos se dan para repartir.

5.5.2.1. Punto de vista del profesor

Se puede visualizar en formato de tabla la posición a nivel global que ocupa cada alumno. Se ve el nombre y los apellidos del alumno y los puntos que lleva recaudados. Cuanto mayor puntuación posee el alumno en mayor posición estará en el ranking.

5.5.2.2. Punto de vista del alumno

Se puede visualizar en formato de lista los alumnos involucrados en el juego. El alumno puede asignar los puntos a cada uno de ellos y enviar sus resultados.

5.5.2.3 Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Visualizar en formato de tabla la posición a nivel global que ocupa cada alumno.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, hemos decidido no añadir nada más porque lo que había ya nos parecía suficiente e interesante.

5.5.3. Todos a uno

En esta modalidad cada participante valora a cada uno de los demás. Cuando se crea el juego se deben definir en qué conceptos se va a basar esa evaluación y que porcentaje de la nota representa cada uno de ellos. Se pueden poner tantos conceptos como se quiera y asignar así sus porcentajes sin superar el 100%.

5.5.3.1. Punto de vista del profesor

Se puede visualizar en formato de tabla la posición a nivel global que ocupa cada alumno. Se ve el nombre y los apellidos del alumno y la nota media global que lleva hasta ese momento. También se puede ver para cada alumno el número de votos que ha recibido por parte de otros alumnos. Cuanto mayor sea la nota media, mayor posición estará en el ranking.

5.5.3.2. Punto de vista del alumno

Se puede visualizar en formato de lista los alumnos involucrados en el juego. El alumno puede asignar los puntos a cada uno de ellos en base los criterios que han sido establecidos previamente por el profesorado. Cada uno de los criterios puede ser evaluado con una nota del 1 al 10. Al finalizar se pueden enviar el resultado.

5.5.3.3 Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Visualizar en formato de tabla la posición a nivel global que ocupa cada alumno.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, hemos decidido no añadir nada más porque lo que había ya nos parecía suficiente e interesante.

5.5.4. A opciones

En esta modalidad cada participante vota por sus opciones preferidas. Cuando se crea el juego se deben definir una lista de opciones sobre las cuales los alumnos elegirán la que más les guste. Y se establece si el reparto de puntos hacia las opciones será fijo, según la posición que ocupa la opción en la lista o si el reparto será libre. En el caso de reparto fijo, se debe establecer qué posiciones si puntuaran y con cuantos puntos lo harán. Y en el caso de reparto libre se debe establecer cuándo puntos máximos se dan para repartir.

5.5.4.1. Punto de vista del profesor

Se puede visualizar en formato de tabla las opciones que se han configurado por parte del profesor y los puntos que lleva sumadas esa opción. También a nivel informativo te muestra el número de alumnos que participan en esta votación y el número de alumnos que ya han votado.

5.5.4.2. Punto de vista del alumno

Se pueden visualizar en formato de lista las opciones que hay en juego. Y el alumno puede asignar su opción favorita por orden y enviar sus resultados. El alumno podrá ver los miembros del equipo (incluido el mismo) y deberá repartir 10 puntos entre los miembros del equipo, donde también podrá escribir observaciones.

Una vez asignados los puntos, se visualizará un resumen de la repartición y un mensaje de si falta alguien por responder.

5.5.4.3 Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Visualizar en formato de tabla las opciones que se han configurado por parte del profesor y los puntos que lleva sumadas esa opción.
- Mostrar el número de alumnos que participan y el número de alumnos que ya han votado.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, hemos decidido no añadir nada más porque lo que había ya nos parecía suficiente e interesante.

5.6. Cuestionario de satisfacción

Para crear el juego de cuestionario de satisfacción, nos dirigimos al grupo al que lo queremos crear, indicamos que queremos crear un juego e indicamos que queremos que sea de tipo 'Cuestionario de satisfacción'. Se abrirá una pestaña donde se tiene que asignar un nombre para el juego y se seguirán una serie de pasos que se explican a continuación.

5.6.1. Crear Juego

Para la creación del juego, primero seleccionamos el grupo al que queremos asignar la creación. A continuación, le damos a la opción 'Juegos' y 'Crear juego', donde en ese momento nos redirigirá a la página para asignar el nombre del juego y el tipo. En este caso vamos a elegir 'Juego de cuestionario de satisfacción', donde existen dos modos: individual y de equipos. Pero en este caso el juego en modo equipos no existe aún, por lo tanto, solo vamos a analizar el juego en modo individual.

El siguiente paso es seleccionar el tipo de cuestionario de satisfacción. En este caso existen dos, el denominado como 'Incidencias críticas' y el de 'Final de curso'. Una vez seleccionado el tipo de cuestionario deseado, te dan la opción de modificar la descripción si es necesario.

5.6.2. Punto de vista del profesor

Funcionalidades a continuación:

- Desactivar el juego
- Visualización del número de participantes, el número de respuestas y una tabla con la afirmación y la media de los resultados.
- Visualización de las respuestas a las preguntas abiertas.

5.6.3. Punto de vista del alumno

Se visualizarán las afirmaciones para responder valorando del 1 al 5, y las preguntas abiertas del profesor a responder como quiera el alumno. Una vez respondida la encuesta, se visualizará un resumen de las respuestas.

5.6.4. Conclusión

Las funcionalidades que se añadirán al móvil del profesor son las siguientes:

- Desactivar el juego
- Visualización del número de participantes, el número de respuestas y una tabla con la afirmación y la media de los resultados.
- Visualización de las respuestas a las preguntas abiertas.

Aparte de estas funcionalidades que ya existían en el Dashboard del profesor, hemos decidido no añadir nada más porque lo que había ya nos parecía suficiente e interesante.

CAPÍTULO 6. JUEGO DE COMPETICIÓN

En este capítulo podremos ver con más detalle en qué consiste el juego de competición, en que diferentes modalidades se dividen y qué funcionalidades nos puede aportar. Además, podremos ver como se ha creado este aplicativo desde cero hasta su estado final, pasando así por muchos detalles interesantes.

Se detallarán también algunas dificultades y decisiones tomadas en el proceso, según las diferentes situaciones en las que nos hemos encontrado. Este capítulo estará dividido en once partes, explicando así el propio juego y sus diferentes funcionalidades. En ambas memorias de Trabajo de Fin de Grado de los dos miembros de este equipo, figuran varios puntos en común y otras funcionalidades distintas.

6.1. ¿En qué consiste el juego de competición?

El juego de competición es un juego en el cual se involucra a los alumnos de una clase en una dinámica competitiva en la cual hay un único vencedor. De esta manera, la actividad llevada a cabo se hace más amena y los alumnos sienten que tienen la responsabilidad de participar activamente, si quieren realmente quedar en una buena posición en frente de sus otros compañeros. Eso hace que el alumno esté más atento a los conceptos impartidos en clase, con tal de poder usarlos en la competición que se proponga y poder así obtener ese premio.

En este proyecto se ha desarrollado este juego de competición en sus diferentes modalidades, pero en esta ocasión nos centraremos en la aplicación del profesor y así podremos verlo desde su punto de vista. El desarrollo de la aplicación se basa en que el juego principal ya ha sido creado mediante el Dashboard principal y que con el móvil se necesita realmente un mero monitoreo o configuraciones específicas que se requieran en el proceso del juego.

Al profesor se le permitirá visualizar las diferentes competiciones que hay en progreso y las puntuaciones o posiciones que ocupan los alumnos que participan en ellas. El progreso de estas competiciones se podrá visualizar mediante diagramas o tablas según la competición. También se le permitirá realizar ciertas modificaciones mediante algunas funcionalidades, aparte de haber configurado el mismo los parámetros principales a través del Dashboard a la hora de crear el propio juego.

6.2. Modalidades del juego de competición

Para el juego de competición existen un total de tres modalidades diferentes, cada una de ellas tiene su propia dinámica. Hay ciertas funcionalidades que comparten todas las modalidades, ya que son básicas y esenciales para el correcto funcionamiento de la misma. Aunque alguna modalidad requiere de alguna más por su propia estructura de juego.

En primer lugar, tenemos la competición en modalidad “**Fórmula Uno**”, consiste en que cada jornada llevada a cabo del juego, únicamente puntúan unos pocos alumnos y con cantidades

variables según la posición que lleguen a ocupar, la persona que sume más puntos al finalizar todas las jornadas, es la declarada como ganadora.

En segundo lugar, tenemos la competición en modalidad “**Torneo**”, consiste en que cada jornada llevada a cabo del juego va disminuyendo el número de alumnos que participan, ya que son descalificados, hasta quedar un solo alumno en la última jornada, este es el declarado como ganador.

En tercer lugar, tenemos la competición en modalidad “**Liga**”, consiste en que cada jornada llevada a cabo del juego, se generen de forma aleatoria encuentros uno contra uno, de entre los alumnos que participen. El encuentro se puede resolver habiendo un vencedor o un empate. Se puede determinar un único vencedor basándonos en el alumno que más encuentros haya ganado al finalizar todas las jornadas.

Más adelante, en los siguientes apartados se podrán ver las diferentes funcionalidades existentes y cómo pueden llevar a cabo el monitoreo en tiempo real de este tipo de modalidades en el contexto de juegos de competición. Cada una de las funcionalidades será explicada en detalle, desde cómo se pensó, diseñó a como se montó y desarrollo, incluyendo así los retos y problemáticas que surgieron en el proceso.

6.3 Tipología del juego de competición

Para el juego de competición igual que para otros juegos, existen un total de dos tipologías diferentes, cada una de ellas tiene su propia dinámica.

En primer lugar, tenemos la competición de tipo “**Individual**”, consiste en que cada alumno participa de forma solitaria.

En segundo lugar, tenemos la competición de tipo “**Grupo**”, consiste en que un número determinado de alumnos forman parte de un mismo equipo y por ello compiten como equipo.

6.4. Página de Inicio

6.4.1 ¿En qué consiste la funcionalidad?

El juego de competición, como ya se ha nombrado en anteriores apartados, es un juego en el cual independientemente de la modalidad que se juegue, hay un número determinado de funcionalidades que se comparten.

Por ello, para la aplicación del profesor necesitamos de una funcionalidad que esté en las tres modalidades del juego y que sea para ver todas las funcionalidades disponibles que hay según la modalidad del juego.

Para ello se ha diseñado una página de inicio que reúne todas las funcionalidades según el caso. De esta manera, se puede tener una visión global de todas las herramientas que se tienen para ese juego de competición en concreto.

6.4.2 Planificación para la implementación

Para la implementación de esta funcionalidad, nos basamos en primer lugar en el diseño ya estipulado en el Dashboard principal. Analizando el Dashboard, nos damos cuenta de que hay 4 botones, uno al lado del otro en la parte superior de la pantalla de inicio, la cual se muestra al seleccionar un juego de competición, de la lista de juegos activos.

Estos cuatro botones se denominan: Información, Editar Jornadas, Resultados y Desactivar. Justo en la parte inferior de esta página de inicio, debajo de los botones, nos encontramos con una tabla o gráfica según la modalidad del juego, que nos indica de forma rápida como avanza el juego en cuestión, según las jornadas vayan pasando. Por ello se puede tener una visión global del progreso en tiempo real.

Una vez analizado el diseño estipulado en el Dashboard, decidimos realizar un diseño similar, pero con algunos cambios, ya que a priori nos parece entendible e intuitivo, pero algunos elementos creemos que deberían ser eliminados, reubicados, añadidos o modificados.

En primer lugar, se ha decidido reubicar los botones de forma vertical en vez de horizontal, de tal manera que se adapte mejor a la pantalla de un teléfono móvil. Los botones se han dividido en dos grupos, y se les ha añadido un icono para cada grupo. Cada grupo de botones va encabezado por un icono distinto.

En segundo lugar, se ha decidido modificar algunos los nombres de los botones para un mejor entendimiento del usuario. Para Información se ha modificado a “Ver Jornadas” y para resultados se ha modificado a “Asignar Ganadores”. Se ha mantenido el nombre de “Editar Jornadas” y “Desactivar”.

En siguientes apartados de este mismo capítulo, se podrá ver como se añaden más botones como “Ver Progreso” y “Editar Puntos”, esta última funcionalidad solo es válida para el juego de competición con modalidad “Fórmula Uno”.

Y en último lugar, se ha decidido eliminar la tabla o gráfica que hay en la parte inferior de la pantalla de inicio, ya que se ha decidido por estética que no concuerda con el modelo que se está realizando en el móvil del profesor. Se reubicará en una nueva página y se le asignará un botón en esta misma página para su redireccionamiento. Es el botón mencionado anteriormente como “Ver progreso”. Se detalla con exactitud en siguientes apartados.

6.4.3 Implementación del juego

Para la implementación del juego hemos tenido que crear una página en exclusiva para visualizar las jornadas existentes, la página se ha denominado “juego-competición”. Para establecer todo lo mencionado anteriormente hemos tenido que modificarlo todo, tanto en el archivo html como en el archivo ts, como en el archivo module.ts.

En el código del archivo **ts**, se ha añadido en la sección OnInit la petición de los datos del propio juego seleccionado, que tiene la siguiente forma:

```
this.juegoSeleccionado = this.sesion.DameJuego();
```

También como elemento adicional en el **ts.** fuera del **OnInit** se han añadido la funcionalidad de los propios botones mencionados anteriormente junto con sus peticiones de datos al "Service", estos elementos tienen la siguiente forma:

```
EditarPuntuacion(){}
```

```
this.TablaeditarPuntos =
this.calculos.DameTablaeditarPuntos(this.juegoSeleccionado);
```

```
EditarJornadas() {}
```

```
this.JornadasCompeticion =
this.calculos.GenerarTablaJornadas(this.juegoSeleccionado, this.jornadas,
this.enfrentamientosDelJuego);
```

```
Desactivar() {}
this.peticionesAPI.CambiaEstadoJuegoDeCompeticionLiga(this.juegoSeleccionado)
.subscribe(res => {})
```

```
InformacionJornadas() {}
```

```
this.JornadasCompeticion =
this.calculos.GenerarTablaJornadas(this.juegoSeleccionado, this.jornadas,
this.enfrentamientosDelJuego);
```

```
AsignacionGanador(){
this.JornadasCompeticion =
this.calculos.GenerarTablaJornadasLiga(this.juegoSeleccionado, this.jornadas,
this.enfrentamientosDelJuego);
```

```
VerProgreso(){ this.navCtrl.navigateForward('/progreso-competicion');}
```

En el código **html** se ha añadido los iconos que separan los dos grupos de botones, mediante la siguiente herramienta:

```

```

Y también se ha añadido el propio botón con la siguiente herramienta:

```
<ion-button color="medium" style="width: 90%;" (click)="VerProgreso();">Ver
Progreso</ion-button>
```

Para llevar a cabo el uso de las anteriores herramientas ha sido necesario importar en el archivo **module.ts** ciertas librerías como son las siguientes:

```
import {MatExpansionModule} from '@angular/material/expansion';
import {MatIconModule} from '@angular/material/icon';
```

6.4.4 Retos destacados

Para esta funcionalidad en concreto no ha habido ninguna dificultad a destacar.

6.5. Funcionalidad 1: Ver progreso

6.5.1 ¿En qué consiste la funcionalidad?

El juego de competición, como ya se ha nombrado en anteriores apartados, es un juego en el cual independientemente de la modalidad que se juegue, hay un número determinado de jornadas que se juegan y para ellas se debe de poder realizar un monitoreo de forma global para saber cómo evoluciona el juego de competición en concreto.

Por ello, para la aplicación del profesor necesitamos de una funcionalidad que esté en las tres modalidades del juego y que sea para ver el progreso del juego de competición. Para ello se ha diseñado la funcionalidad de “Ver Progreso”.

De esta manera se puede tener una visión global de la evolución de la competición que avanza según se vayan produciendo las jornadas.

6.5.2 Planificación para la implementación

Para la implementación de esta funcionalidad nos basamos en primer lugar en el diseño ya estipulado en el Dashboard principal. Para acceder a esta funcionalidad hemos de acceder a través del botón “Información” que hay en la página de inicio del juego de competición. Recordemos que en apartados anteriormente ya se ha mencionado esta funcionalidad, pero que se implementara en el móvil del profesor de forma separada.

En este apartado procederemos a explicar en detalle como son las tablas o esquemas para la visión global de la evolución del juego según las modalidades existentes para el juego de competición.

Analizando el Dashboard nos damos cuenta para la modalidad de “Torneo” hay un esquema el cual está dividido en varias fases, en cada fase podemos ver que el número de participantes se disminuye hasta quedar solo un único vencedor. Los participantes realizan enfrentamientos cada dos de ellos y eso se puede ver reflejado en las flechas que se dibujan entre los nombres de los participantes. El esquema está realizado de forma horizontal.

Para la modalidad de “Fórmula Uno”, se utiliza para visualizar el progreso del juego de competición, una tabla. Esta tabla consta de 5 columnas, tres de ellas son para el nombre, primer y segundo apellido de los participantes. La primera columna nos muestra el número de la posición que ocupa ese participante en la clasificación general.

Y la última columna es para indicarnos la cantidad de puntos que tiene cada uno de los participantes. El participante con más punto y que por ello se encuentra en la primera posición de la tabla, se le añade un icono de una copa de vencedor. Todos los datos de la primera posición de la tabla se encuentran escritos de forma gruesa y de color verde.

Y como última modalidad tenemos “Liga”, se utiliza para visualizar el progreso del juego de competición, una tabla. Esta tabla consta de 8 columnas, una de ellas es para el nombre, primer y segundo apellido de los participantes. La primera columna nos muestra el número de la posición que ocupa ese participante en la clasificación general.

Y la última columna es para indicarnos la cantidad de puntos que tiene cada uno de los participantes. El participante con más punto y que por ello se encuentra en la primera posición de la tabla, se le añade un icono de una copa de vencedor. Todos los datos de la primera posición de la tabla se encuentran escritos de forma gruesa y de color verde.

Y las 5 columnas restantes nos muestran datos relacionados con lo sucedido en el propio juego, como lo son: partidas totales, partidas jugadas, partidas ganadas, partidas empatadas y partidas perdidas.

Una vez analizado el diseño estipulado en el Dashboard decidimos realizar un diseño similar, pero con algunos cambios, ya que a priori nos parece entendible e intuitivo, pero algunos elementos creemos que deberían ser modificados.

En primer lugar, se ha decidido mantener las tablas exactamente iguales, con los mismos componentes tanto de la modalidad de “Liga” como de la modalidad de “Fórmula Uno”, ya que son esenciales para el correcto seguimiento del juego de competición.

Pero para la modalidad de Torneo, se respetará la estructura horizontal, pero se eliminarán las flechas relacionales. La razón es, que los nombres estarán en tarjetas y se dejará suficiente margen entre ellas, en la sección de diseño se podrá ver con más detalle.

6.5.3 Implementación del juego

Para la implementación del juego hemos tenido que crear una página en exclusiva para visualizar las jornadas existentes, la página se ha denominado “progreso-competición”. Para establecer todo lo mencionado anteriormente hemos tenido que modificarlo todo, tanto en el archivo html como en el archivo ts, como en el module.ts.

En el código del archivo **ts**. se ha añadido en la sección OnInit la petición de los datos del propio juego seleccionado y se ha ejecutado una funcionalidad que está fuera de la sección OnInit, ambos componentes tienen la siguiente forma:

```
this.juegoSeleccionado = this.sesion.DameJuego();
this.DameJornadasDelJuegoDeCompeticionSeleccionado();
```

En el **ts**. fuera del OnInit se encuentra la funcionalidad anteriormente ejecutada, la cual incluye una petición a la API de las jornadas de competición según la modalidad del juego en cuestión.

```
this.peticionesAPI.DameJornadasDeCompeticion(this.juegoSeleccionado.id).subscribe(
  inscripciones => {})
```

También como elemento adicional en el **ts.** fuera del **OnInit** se han añadido la funcionalidad de recoger todos los enfrentamientos del juego en cuestión, lo cual incluye una petición a la API que tiene la forma siguiente:

```
DameEnfrentamientosDelJuego() {}
this.peticionesAPI.DameEnfrentamientosDeCadaJornada(this.jornadas[i].id)
  .subscribe((enfrentamientosDeLaJornada: Array<EnfrentamientoLiga>) => {})
```

También se ha recogido el nombre de los participantes del juego de competición mediante la siguiente funcionalidad, la cual ejecuta una petición a la API para ello.

```
AlumnosDelJuego() {}
this.peticionesAPI.DameAlumnosJuegoDeCompeticionLiga(this.juegoSeleccionado.i)
  .subscribe(alumnosJuego => {})
```

Y finalmente se recoge el nombre del ganador para el caso del juego de competición de “Torneo” con la siguiente funcionalidad.

```
DameNombreGanador() {}
```

Aquí podemos ver la última funcionalidad, la cual con los datos anteriores se montan las tablas a visualizar en pantalla.

```
TablaClasificacionTotal() {}
```

En el código **html** se ha añadido las tablas o esquema según la modalidad del juego de competición. Para la modalidad de “Liga” y “Fórmula Uno”, se ha usado la siguiente herramienta:

```
<table style="width: 100%" mat-table [dataSource]="dataSource">
```

Cada uno de los datos que irá en las columnas se introduce de la siguiente manera:

```
<ng-container matColumnDef="select">
```

Y para la modalidad de “Torneo” se ha utilizado la siguiente herramienta:

```
<ion-card>
```

Para llevar a cabo el uso de las anteriores herramientas ha sido necesario importar en el **module.ts** ciertas librerías como son las siguientes:

```
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatTableModule } from '@angular/material';
import { MatTooltipModule } from '@angular/material';
import { MatButtonModule, MatCardModule, MatToolbarModule } from
  '@angular/material';
import { MdePopoverModule } from '@material-extended/mde';
import { MatListModule } from '@angular/material/list';
import { MatInputModule } from '@angular/material';
```

6.5.4 Retos destacados

Para esta funcionalidad en concreto no ha habido ninguna dificultad a destacar.

6.6. Funcionalidad 2: Editar Jornadas

6.6.1 ¿En qué consiste la funcionalidad?

El juego de competición, como ya se ha nombrado en anteriores apartados, es un juego en el cual independientemente de la modalidad que se juegue, hay un número determinado de jornadas que se realizan que son establecidas en un comienzo por parte del profesor.

Además, estas jornadas deberían poderse modificar por parte del profesor, tanto el criterio por el cual se establecerá el vencedor como la fecha en la que se llevará a cabo.

Por ello, para la aplicación del profesor necesitamos de una funcionalidad que esté en las tres modalidades del juego y que sea para editar las jornadas que existen. Para ello se ha diseñado la funcionalidad de “Editar Jornada”.

De esta manera se puede tener una visión global de las jornadas de esa competición y poder modificarlas.

6.6.2 Planificación para la implementación

Para la implementación de esta funcionalidad nos basamos en primer lugar en el diseño ya estipulado en el Dashboard principal. Analizando el Dashboard nos damos cuenta en primer lugar que hay un formulario para la fecha, consta de su propio título justo arriba y de un icono a la derecha en el propio formulario. El icono tiene forma de agenda y al clicar sobre él se despliega un calendario por meses, donde se puede elegir manualmente un día de algún mes y con solo seleccionar el día se establece por escrito en el espacio del formulario. Ya que la fecha se puede introducir o por escrito o mediante el desplegable del calendario, lo cual resulta más amigable y visual.

En segundo lugar, podemos ver a continuación del primer formulario un segundo formulario con su correspondiente título en la parte superior, en el formulario podemos introducir por manualmente y por escrito el criterio por el cual será elegido el vencedor.

Una vez introducido uno o los elementos que queremos editar se necesitan seleccionar para qué jornada deben efectuarse esas modificaciones. Por ello, a continuación, en la parte de abajo, se puede observar una tabla con todas las jornadas existentes, incluyendo así las fechas y criterios que tendrás, en el caso de haber sido establecidas en un principio por defecto, de no ser así el espacio correspondiente aparece en blanco a establecer.

La tabla incluye para cada jornada un checkbox el cual nos permite seleccionar una jornada en concreto a la cual afectará los cambios establecidos anteriormente. Una vez seleccionada la jornada a editar, se habilita el botón de guardado que está justo al lado de del segundo formulario. El cual al ser pulsado envía la información a la API y es registrado.

También se incluye en la parte inferior un botón para poder volver hacia atrás, es decir, a la lista de juegos activos.

Para el caso concreto de la modalidad de Fórmula uno, esta funcionalidad, tal y como ha sido descrita anteriormente, va dentro de un deslizable y en la segunda parte del deslizable se puede ver una segunda funcionalidad que se explicará más adelante en otro apartado.

Una vez analizado el diseño estipulado en el Dashboard decidimos realizar un diseño similar, pero con algunos cambios, ya que a priori nos parece entendible e intuitivo, pero algunos elementos creemos que deberían ser eliminados y reubicados.

En primer lugar, se mantendrán los dos formularios con sus respectivos títulos y el desplegable de calendario que otorga el icono, puesto que nos parece esencial para la funcionalidad en cuestión. El botón inferior que nos permite volver atrás será eliminado, ya que esa funcionalidad ya la tenemos establecida por defecto en todas las páginas de la aplicación del profesor, por ello se considera irrelevante.

También se mantendrá las tablas de las jornadas con todos sus elementos mencionado anteriormente porque va acorde con la funcionalidad a implementar. En esta ocasión será modificada la posición de los elementos para que se pueda adaptar a la pantalla de un teléfono móvil. Por ello los elementos no están de forma adyacente uno con el otro, sino de forma vertical, uno debajo del otro. Creando así una dinámica por pasos entendible y dirigida para el profesor. De tal manera que el botón final de guardado quede en la parte inferior, es decir, al finalizar la edición y no en la superior.

Para el caso concreto de la modalidad de Fórmula uno, se eliminará el deslizable y solo dejará los elementos mencionados anteriormente como en cualquier otra modalidad de juego y la funcionalidad adicional se explicará en apartados siguientes.

6.6.3 Implementación del juego

Para la implementación del juego hemos tenido que crear una página en exclusiva para visualizar las jornadas existentes, la página se ha denominado "editar-jornadas". Para establecer todo lo mencionado anteriormente hemos tenido que modificarlo todo, tanto en el archivo html como en el archivo ts, como en el module.ts.

En el código del archivo **ts**. se ha añadido en la sección OnInit la petición de los datos de las jornadas relacionadas con ese juego, en concreto de la forma siguiente:

```
this.sesion.DameDatosJornadas();
```

Llamando así a una funcionalidad que se encuentra en el Service nombrado "sesión". Esta funcionalidad realiza una petición directamente a la API. Y también se introduce esta información la tabla que se mostrará en pantalla, la introducción de los dos tiene la siguiente forma:

```
this.dataSource = new MatTableDataSource(this.JornadasCompeticion);
```


Como último elemento en esta sección podemos ver cómo se inicializa el formulario, este elemento tiene la siguiente forma:

```
this.myForm = this._formBuilder.group({})
```

También como elemento adicional en el ts. fuera del OnInit se ha añadido la propia funcionalidad de editar jornada, que se muestra a continuación:

```
EditarJornada() {}
```

Esta funcionalidad recoge los datos introducidos en los dos formularios y los guarda mediante una petición a la API, esta gestión se realiza de la siguiente manera:

```
NuevoCriterio = this.myForm.value.CriterioGanador;
NuevaFecha = this.myForm.value.picker;
```

```
this.peticionesAPI.ModificarJornada (this.jornadas[i], this.IDJornada)
    .subscribe(JornadaCreada => {})
```

En el código **html** se ha añadido un formulario donde contendrá dos entradas de datos posibles. Para ello, se ha utilizado la siguiente herramienta:

```
<form [formGroup]="myForm" class = "enFila" style="width: 80%">
```

Los datos a introducir se recogen con la siguiente herramienta dentro del formulario:

```
<input matInput [matDatepicker]="picker"
(dateChange)="onChangeEvent($event)" placeholder="Seleccione una
Fecha">
```

Y para gestionar el desplegable del calendario se ha utilizado la siguiente herramienta:

```
<mat-datepicker-toggle matSuffix[for]="picker"></mat-datepicker-toggle>
<mat-datepicker #picker></mat-datepicker>
```

Se incluye también la tabla de las jornadas con la siguiente herramienta:

```
<table style="width: 100%" mat-table [dataSource]="dataSource">
```

Cada una de los datos que ira en las columnas se introduce de la siguiente manera:

```
<ng-container matColumnDef="select">
```

Y los seleccionadores utilizan la siguiente herramienta:

```
<mat-checkbox>
```

Para llevar a cabo el uso de las anteriores herramientas ha sido necesario importar en el `module.ts` ciertas librerías como son las siguientes:

```
import { MatSnackBarModule } from '@angular/material/snack-bar';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatButtonModule } from '@angular/material/button';
import { MatInputModule } from '@angular/material/input';
import { MatDatepickerModule } from '@angular/material/datepicker';
import { MatNativeDateModule } from '@angular/material/core';
import { MatTableModule } from '@angular/material/table';
import { MatCheckboxModule } from '@angular/material/checkbox';
```

El código de las funcionalidades ya presentadas se encuentra en el propio GitHub.

6.6.4 Retos destacados

Para esta funcionalidad en concreto no ha habido ninguna dificultad a destacar.

6.7. Funcionalidad 3: Editar Puntos

6.7.1 ¿En qué consiste la funcionalidad?

El juego de competición, como ya se ha nombrado en anteriormente, tiene hasta tres modalidades y una de ellas tiene una particular dinámica, como lo es Fórmula un. De los participantes del juego solo puntuarán unos pocos y con unos puntos preestablecidos desde un inicio en el Dashboard.

Por ello, para la aplicación del profesor necesitamos de una funcionalidad adicional para esta modalidad de juego, ya que el profesor debe poder modificar esos valores, es decir, tanto el número de alumnos que realmente puntuaron como la puntuación que percibirán.

Para ello se ha diseñado la funcionalidad de “Editar Puntos”. De esta manera se puede tener una visión global de los puntos que se otorgarán y poder modificarlas.

6.7.2 Planificación para la implementación

Para la implementación de esta funcionalidad nos basamos en primer lugar en el diseño ya estipulado en el Dashboard principal. Analizando el Dashboard nos damos cuenta de que esta funcionalidad comparte página con otra funcionalidad ya mencionada en otro apartado. Ambas funcionalidades están separadas por un deslizable. En esta ocasión analizamos solo esta funcionalidad en cuestión.

En primer lugar, vemos que hay un formulario donde introducir manualmente y en formato numérico la cantidad de puntos a asignar, este formulario incluye un título en la parte superior. Justo al lado se puede ver dos botones, uno de ellos es para establecer los puntos en la posición que se haya elegido de la tabla que veremos a continuación, de esta manera se puede ver el resultado final como queda sin guardar los cambio aún.

Y el segundo botón guarda los cambios como tal mandando esta información a la API mediante una petición. Estos botones únicamente se habilitan cuando los datos ya han sido introducidos y elegido de la tabla a qué posición afectará.

Más abajo y como elemento central tenemos una tabla que nos muestra cuántas posiciones como máximo habrá, es decir, cuántos ganadores habrá para cada jornada. Aparte de las posiciones también nos muestra los puntos que se otorgarán a cada posición. En la parte de izquierda se añade un checkbox el cual se debe seleccionar cuando se ha introducido la cantidad de puntos a otorgar en el formulario, de esta manera se puede seleccionar a qué posición afectará ese cambio.

Al final de la página podemos ver dos botones más, ya que no solo se puede modificar la cantidad de puntos que se otorgara a las posiciones que aparecen en la tabla, sino que también se pueden añadir o quitar posiciones en la tabla, es decir, se puede modificar el número de participantes que puntuaran en esa jornada. Uno de los botones es para añadir una posición más y el otro botón es para borrar la última posición que haya en la tabla. También se incluye en la parte inferior un botón para poder volver hacia atrás, es decir, a la lista de juegos activos.

Una vez analizado el diseño estipulado en el Dashboard decidimos realizar un diseño similar, pero con algunos cambios, ya que a priori nos parece entendible e intuitivo, pero algunos elementos creemos que deberían ser eliminados y reubicados.

En primer lugar, se mantendrá el formulario donde introducir de forma manual los puntos a asignar junto a los dos botones, establecer los puntos y guardar los cambios porque es algo esencial para el correcto funcionamiento del mismo.

El botón inferior que nos permite volver atrás será eliminado, ya que esa funcionalidad ya la tenemos establecida por defecto en todas las páginas de la aplicación del profesor, por ello se considera irrelevante.

También se mantendrá las tablas con todos sus elementos mencionados anteriormente, puesto que va acorde con la funcionalidad a implementar. Incluyendo así los dos botones de añadir y quitar posiciones.

En esta ocasión será modificada la posición de los elementos para que se pueda adaptar a la pantalla de un teléfono móvil. Por ello los elementos no están de forma adyacente uno con el otro, sino de forma vertical, uno debajo del otro. Creando así una dinámica por pasos entendible y dirigida para el profesor. De tal manera que los botones de guardado queden en la parte inferior, es decir, al finalizar la edición y no en la superior.

6.7.3 Implementación del juego

Para la implementación del juego hemos tenido que crear una página en exclusiva para visualizar los puntos existentes, la página se ha denominado “editar-puntos”. Para establecer todo lo mencionado anteriormente hemos tenido que modificarlo todo, tanto en el archivo html como en el archivo ts, como en el module.ts.

En el código del archivo **ts.** se ha añadido en la sección **OnInit** la petición de los datos de la tabla que se presenta de la forma siguiente:

```
this.TablaPuntuacion = this.sesion.DameTablaeditarPuntos();
```

Justo a continuación se añaden esos datos a la tabla que se mostrará. Los datos son introducidos de la siguiente manera:

```
this.dataSource = new MatTableDataSource (this.TablaPuntuacion);
```

También como elemento adicional en el **ts.** fuera del **OnInit** se ha añadido las funcionalidades de los botones para añadir y eliminar posiciones en la tabla. Estas funciones tienen la siguiente forma:

```
AnadirJugadorconPuntos() {}
```

```
EliminarJugadorconPuntos() {
```

Como funcionalidad principal tenemos la del botón donde se establecen los puntos, esta función recoge los datos introducidos y tiene la siguiente forma:

```
AnadirPuntos() {
```

```
NuevaPuntuacion = Number(this.myForm1.value.NuevaPuntuacion);
```

También como función principal tenemos la del botón de guardado de datos, que los envía a la API para ser almacenados mediante una petición. Esta función tiene la forma siguiente:

```
goBackandguardar() {
```

```
this.peticionesAPI.ModificaJuegoDeCompeticionFormulaUno(this.JuegoModificado, this.JuegoModificado.id)
    .subscribe(JuegoModificado => {})
```

En el código **html** se ha añadido un formulario donde contendrá una entrada de datos posible. Para ello se ha utilizado la siguiente herramienta:

```
<form [formGroup]="myForm1" class = "enFila" style="width: 80%" >
```

Los datos a introducir se recogen con la siguiente herramienta dentro del formulario:

```
<input matInput formControlName="NuevaPuntuacion" placeholder="Escriba los Puntos" required [(ngModel)]="NuevaPuntuacion">
```

Se incluye también la tabla de las puntuaciones con la siguiente herramienta:

```
<table style="width: 100%" mat-table [dataSource]="dataSource">
```

Cada una de los datos que irá en las columnas se introduce de la siguiente manera:

```
<ng-container matColumnDef="select">
```

Y los seleccionadores utilizan la siguiente herramienta:

```
<mat-checkbox>
```

Para llevar a cabo el uso de las anteriores herramientas ha sido necesario importar en el **module.ts** ciertas librerías como son las siguientes:

```
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { MatCheckboxModule } from '@angular/material/checkbox';
import { MatTableModule } from '@angular/material/table';
```

6.7.4 Retos destacados

Para esta funcionalidad en concreto no ha habido ninguna dificultad a destacar.

6.8. Funcionalidad 4: Desactivar

6.8.1 ¿En qué consiste la funcionalidad?

El juego de competición como ya se ha nombrado en anteriores apartados, es un juego en el cual, independientemente de la modalidad que esté en uso, el juego debe poder desactivarse y dejar de ser accesible a los alumnos. Por ello es necesario pasar el juego de modo activado ha desactivado.

Por ello, para la aplicación del profesor necesitamos de una funcionalidad que esté en las tres modalidades del juego y que sea para desactivar un juego. Para ello se ha diseñado la funcionalidad de “Desactivar”.

De esta manera se puede desactivar un juego en cualquier momento y que, por lo tanto, los alumnos no tengan acceso.

6.8.2 Planificación para la implementación

Para la implementación de esta funcionalidad nos basamos en primer lugar en el diseño ya estipulado en el Dashboard principal. Analizando el Dashboard nos damos cuenta de que la funcionalidad de desactivar viene dada por un botón que hay en la página principal del juego de competición, por ello no tiene página propia.

Al ser pulsado el botón nos aparece un popup, el cual nos pregunta si estamos seguros de desactivar el juego. Al continuar en el mismo popup hay dos botones, uno para aceptar la desactivación y otro para desestimarlo.

Una vez analizado el diseño estipulado en el Dashboard decidimos realizar un diseño similar con los mismos elementos.

Se mantendrá el popup con la pregunta principal y los dos botones con el mismo diseño y las posiciones de los elementos.

6.8.3 Implementación del juego

Como para la implementación del juego no es necesario crear una página en exclusiva para la funcionalidad, se ha añadido código para su implantación en la misma página de inicio del juego de competición. Para establecer todo lo mencionado anteriormente hemos tenido que modificarlo todo, tanto en el archivo html como en el archivo ts, como en el module.ts.

En el código del archivo **ts**. de la página principal del juego de competición se ha añadido fuera de la sección OnInit la funcionalidad con forma siguiente:

```
Desactivar() {}
```

En su interior se ha utilizado la siguiente herramienta para generar el popup:

```
Swal.fire({})
```

En el cual se ha introducido la pregunta mencionada anteriormente. Después de tener en cuenta en qué modalidad estamos trabajando, se hace una petición directamente a la API para borrar esa información mediante una petición. Esta petición tiene la forma siguiente:

```
this.peticionesAPI.CambiaEstadoJuegoDeCompeticionLiga(this.juegoSeleccionado).subscribe(res => {})
```

En el código **html** de la página principal del juego de competición se ha añadido un botón que activa la funcionalidad directamente y ha sido implementado mediante la siguiente herramienta:

```
<ion-button color="danger" style="width: 90%;"  
(click)="Desactivar();">Desactivar juego</ion-button>
```

Para llevar a cabo el uso de la anterior herramienta ha sido necesario importar en el **module.ts** cierta librería como es la siguiente:

```
import Swal from 'sweetalert2';
```

6.8.4 Retos destacados

Para esta funcionalidad en concreto no ha habido ninguna dificultad a destacar.

6.9. Testing del correcto funcionamiento

En este punto, se explicará de forma detallada como se comprobó el correcto funcionamiento del juego, una vez implementado con respecto al código. Se realiza mediante una guía por pasos a seguir, que se ejecutó y realizó en su momento y que cualquier usuario con acceso al mismo puede realizar para su comprobación.

Para este juego en concreto se realizará la comprobación de un solo modo, ya que las funcionalidades básicas son las mismas para los tres casos. La modalidad escogida será fórmula 1, puesto que tiene una funcionalidad adicional que las demás no tienen y de esta manera se pueden abarcar la totalidad de funciones existentes.

La comprobación del correcto funcionamiento del juego, comienza con la creación de un juego nuevo mediante el Dashboard, cosa que ya se ha mencionado y explicado en capítulos anteriores, por ello en este punto se comenzará la comprobación desde el punto de vista del profesorado, con la aplicación del profesor, mediante pasos a seguir.

1. Abrimos la aplicación del profesor.
2. Introduciendo un usuario y su respectiva contraseña en el formulario.
3. Accedemos a la propia aplicación, pulsamos el botón de “Log In”.
4. Deslizamos hasta encontrar el juego seleccionado, para ello se dispone de botones para adelante y para atrás.
5. Accedemos al juego en cuestión pulsando en el botón “Ver más”.

En este punto podemos ver la página principal del juego donde se nos muestran todas las funcionalidades disponibles para el profesor. Para este caso, se empezará por las funcionalidades de configuración, más tarde por la de asignación de ganadores y así de esta más adelante podremos monitorizar esos mismos resultados en las funcionalidades restantes. Esto se puede observar en: Figure 14,15,16 y 17: Capturas de los pasos a seguir.

6. Seleccionamos la opción de “Editar Puntos”, mediante su respectivo botón.

En “editar puntos” podemos ver el formulario principal donde añadir manualmente la numeración de puntos que se quiere asignar. A continuación, se puede ver la tabla con las distintas posiciones posibles, estas posiciones son seleccionable mediante sus respectivos checkbox's, también se incluye la cantidad de puntos ya asignados a cada posición. Se procederá a añadir una nueva posición y a asignarle una cantidad de puntos determinada.

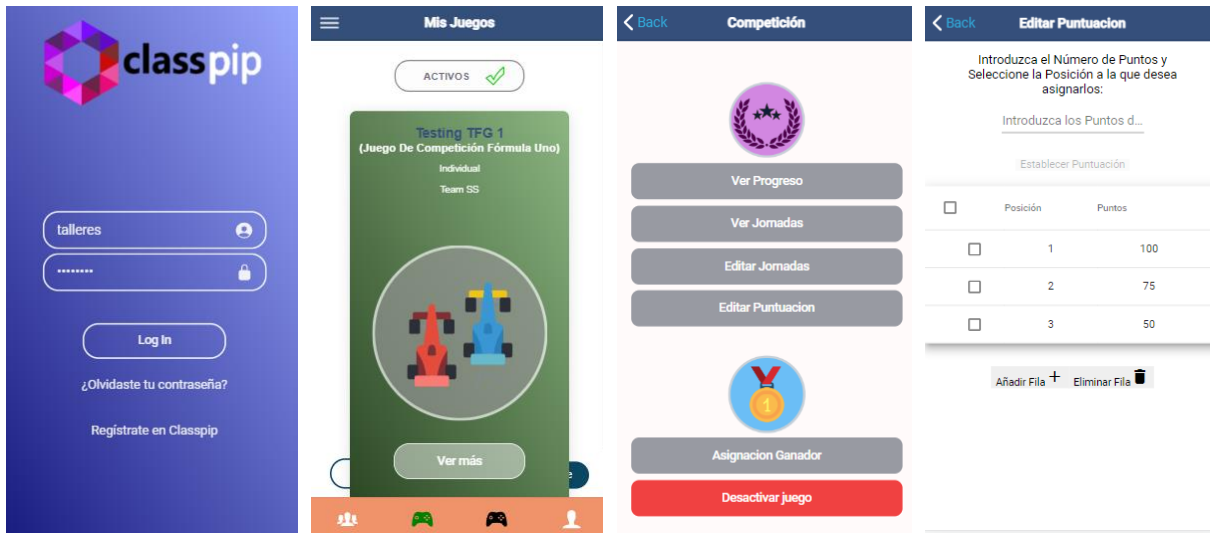


Fig. 14,15,16 y 17: Capturas de los pasos a seguir

7. Pulsamos el botón “Añadir Fila” que hay en la parte inferior.
8. Escribimos en el formulario “25”, introduciendo así los puntos.
9. Seleccionamos la última posición creada, seleccionado el checkbox.
10. Guardamos los cambios pulsando el botón de “Establecer puntos”
11. Pulsamos el botón “Back” para volver a la página de inicio.
12. Seleccionamos la opción de “Editar Jornadas”, mediante su respectivo botón.

En “editar jornadas” podemos ver dos formularios, el primero para seleccionar una fecha tanto manualmente como a través de un calendario y el otro para escribir manualmente el criterio por el cual será elegido el ganador. A continuación, se puede ver la tabla con las distintas jornadas posibles, estas jornadas son seleccionable mediante sus respectivos checkbox’s, también se una enumeración y la fecha y el criterio asignado. Se procederá a modificar una jornada y a asignarle una nueva fecha y criterio. Esto se puede observar en: Figure 18, 19, 20 y 21: Capturas de los pasos a seguir.

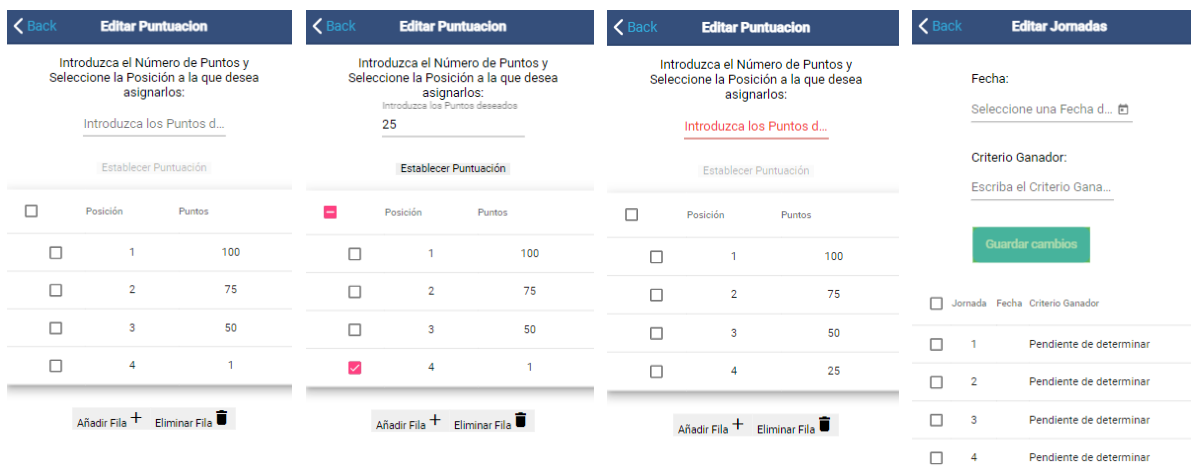


Fig. 18, 19, 20 y 21: Capturas de los pasos a seguir

13. Desplegamos el calendario mediante el botón que hay al lado del formulario, llamado “Fecha”.
14. Seleccionamos un día, con ellos se autocompletará este formulario.
15. Escribimos en el siguiente formulario “Intuición”
16. Seleccionamos la primera posición, seleccionando el checkbox.
17. Guardamos los cambios pulsando el botón de “Guardar Cambios”
18. Pulsamos el botón “Back” para volver a la página de inicio.
19. Seleccionamos la opción de “Asignar Ganadores”, mediante su respectivo botón.

En “Asignar Ganadores” podemos ver hasta tres formularios según el caso, uno para seleccionar la jornada, el segundo donde se puede seleccionar la metodología a seguir para asignar los ganadores y un tercer formulario para seleccionar en que juego nos queremos basar para asignar la puntuación, este último caso solo se da si se elige la metodología “según resultados de juegos”.

También se puede ver una tabla con las diferentes posiciones según la puntuación recibida, cada posición incluye el nombre y los apellidos del participante. Cada metodología tiene su propia dinámica. Esto se puede observar en desde la figura 22 a la 40.



Fig. 22, 23, 24 y 25: Capturas de los pasos a seguir

20. Seleccionamos la primera jornada mediante el desplegable.
21. Seleccionamos el modo de asignación (Manual) mediante el siguiente desplegable.
22. Escribimos el nombre y los dos apellidos de las personas que van a puntuar en el orden que van a puntuar.
23. Guardamos los cambios pulsando el botón de “Asignar Ganadores Manualmente”.

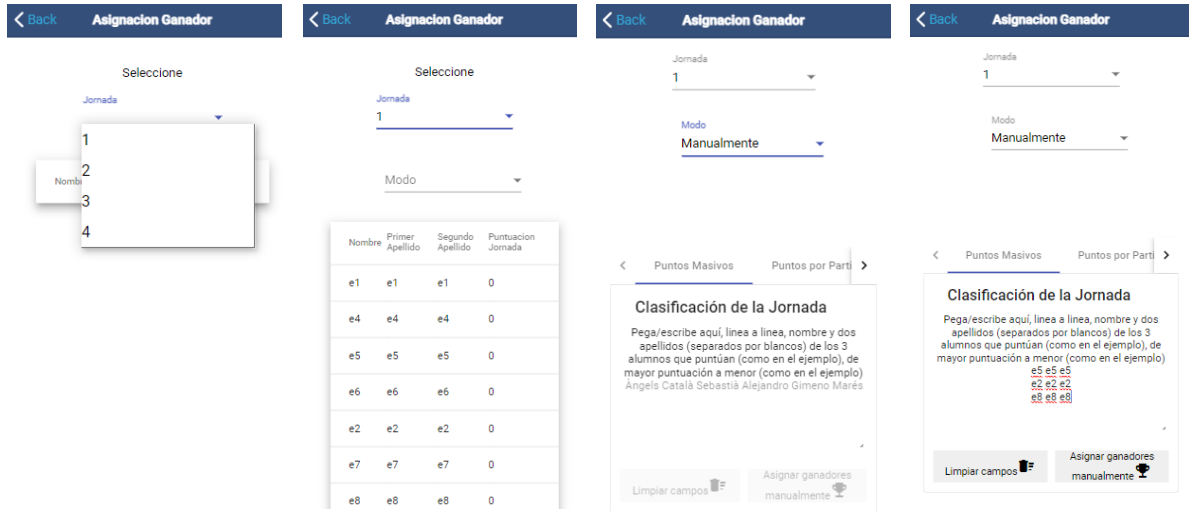


Fig. 26, 27, 28 y 28: Capturas de los pasos a seguir

Para esta metodología en concreto existen dos opciones, la siguiente opción va a continuación:

24. Deslizamos hacia la derecha para acceder a otra metodología de asignación de ganadores de forma manual.
25. Pulsamos el botón de “+” que hay al lado de cada participante para que puntué y los pulsamos en el orden que queremos que puntuén también.
26. Guardamos los cambios pulsando el botón de “Asignar Ganadores Manualmente”
27. Seleccionamos la segunda jornada mediante el desplegable.

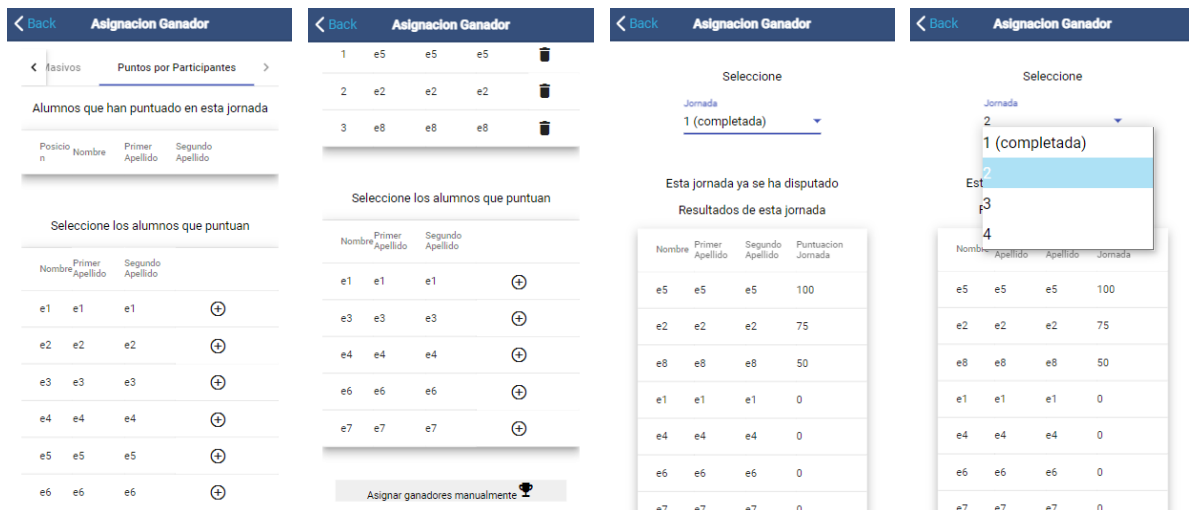


Fig. 29, 30, 31 y 32: Capturas de los pasos a seguir

28. Seleccionamos el modo de asignación (Aleatorio) mediante el siguiente desplegable.
29. Guardamos los cambios pulsando el botón de “Asignar Ganadores Aleatoriamente”
30. Seleccionamos la tercera jornada mediante el desplegable.

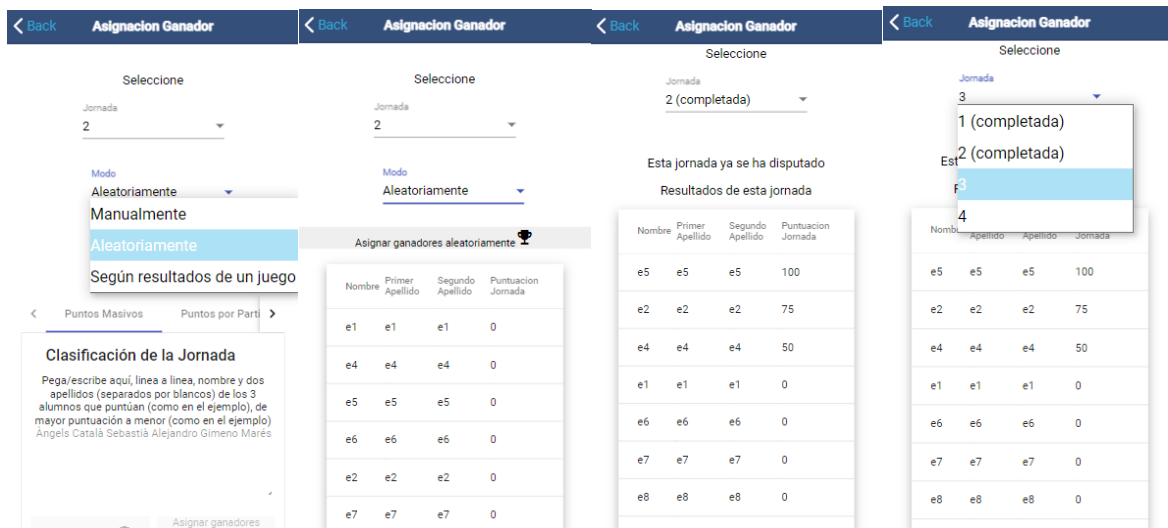


Fig. 33, 34, 35 y 36: Capturas de los pasos a seguir

31. Seleccionamos el modo de asignación (Según resultados de un juego) mediante el siguiente desplegable.
32. Seleccionamos el juego ya finalizado en el que queremos basar los ganadores.
33. Guardamos los cambios pulsando el botón de “Asignar Ganadores Mediante Juego Seleccionado”.
34. Pulsamos el botón “Back” para volver a la página de inicio.

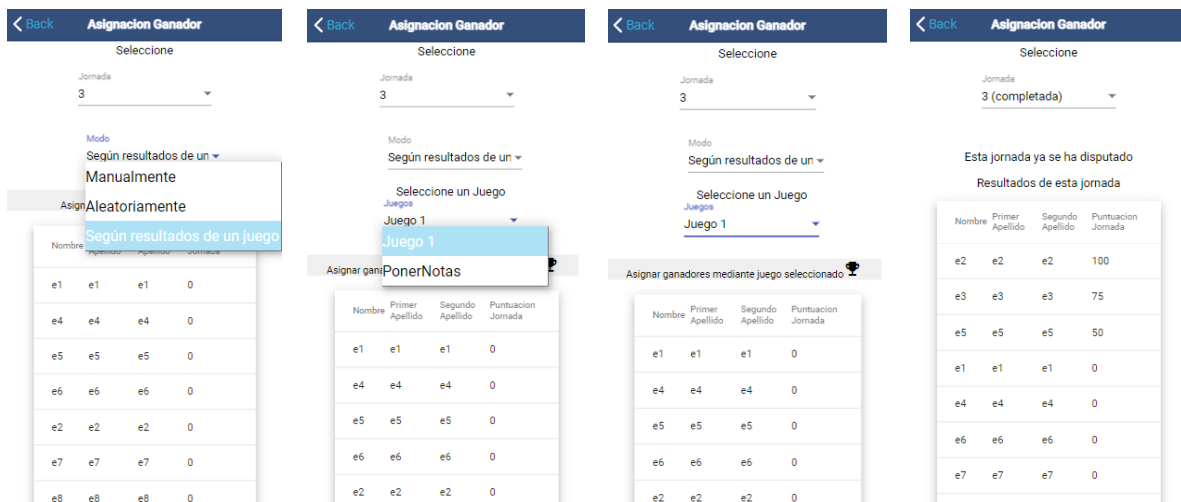


Fig. 37, 38, 39 y 40: Capturas de los pasos a seguir

35. Seleccionamos la opción de “Ver Jornada”, mediante su respectivo botón.
36. Pulsamos el botón “Back” para volver a la página de inicio.
37. Seleccionamos la opción de “Ver Progreso”, mediante su respectivo botón.
38. Pulsamos el botón “Back” para volver a la página de inicio.
39. Pulsamos el botón desactivar para eliminar el juego de la lista de juegos activos.

En “Ver Jornada” podemos ver que hay desplegable que en el mismo título nos indica la numeración y el icono que nos indica el estado y al abrirlos nos muestra la información de esa jornada como la fecha y el criterio del vencedor. Esto se puede observar en: *Figure 41 y 42.*

En “Ver Progreso” podemos ver una tabla con las diferentes posiciones según la puntuación recibida, cada posición incluye el nombre y los apellidos del participante. Esto se puede observar en: *Figure 43.*

Y en “desactivar juego” podemos ver una ventana emergente que nos pregunta si queremos realmente desactivar el juego, para ello le acompañan dos botones, uno para aceptar la desactivación y otro para desestimarla. Esto se puede observar en: *Figure 44.*



CAPÍTULO 7. JUEGO DE CONTROL DE TRABAJO

En este capítulo se explicará en qué consiste el juego de control de trabajo, las diferentes funcionalidades implementadas en el juego, que estrategia o planificación se utilizó para su implementación, los retos más destacados a lo que me he enfrentado y finalmente las soluciones que se aplicaron según el reto presentado.

Este capítulo es totalmente diferente al de mi compañero de proyecto, puesto que en este capítulo cada uno explica el juego que ha implementado de forma independiente.

7.1. ¿En qué consiste el juego de control de trabajo?

El juego de control de trabajo consiste en una herramienta del profesor para la supervisión periódica y evolución del trabajo de los alumnos. Donde el profesor, a partir de un proyecto que se esté realizando en clase, puede dividirlo por partes para controlar el progreso de los alumnos que conformen ese grupo de trabajo.

En la aplicación, podrá controlar el progreso mediante la creación de un número de controles que el docente considere necesario para cada equipo de proyecto. Cuando sea necesario llevar a cabo ese control, los alumnos deben repartir 10 puntos entre los miembros del equipo (ellos incluidos), donde también podrán escribir observaciones en relación con el trabajo en grupo.

Una vez asignados los puntos, se mostrará una tabla con el icono correspondiente al estado del control.

Hay diferentes estados en los que un control de trabajo puede estar, como se puede ver en la imagen anterior. Puede estar en estado completado, es decir, todos los miembros del equipo han completado la asignación de puntos de ese control. También puede estar en estado deficiente, es decir, todos los miembros han completado el control, pero alguno ha recibido una calificación especialmente baja. Y por último puede estar en estado en progreso, es decir, solo algunos miembros del equipo han completado el control. En el caso, de que no haya ningún icono y, por lo tanto, el campo esté vacío, el estado es no evaluado, es decir, la fecha de inicio de ese control de trabajo no ha sido acordada aún y, en consecuencia, ningún miembro del grupo ha realizado la asignación de puntos de ese control aún.

Mediante un clic en los iconos de cada uno de los controles, el docente podrá ver en detalle la puntuación de los diferentes miembros del grupo hacia sus otros compañeros y los comentarios que hayan podido dejar. Se explicará con más detalle esta funcionalidad en el siguiente apartado.

Gracias a la información proporcionada por esta herramienta, el profesor será capaz de saber cuál es el progreso y estado de cada uno de los equipos de proyecto, y así poder resolver cualquier problema que se esté produciendo en el grupo antes de que finalice el trabajo. O también puede ser información útil de cara a la evaluación final del proyecto, ya que el docente tendrá la información necesaria para tomar decisiones de qué miembros del equipo han contribuido más o menos en el proyecto propuesto.

El profesor también tiene la opción de enviar un recordatorio a los alumnos que no han hecho la repartición de puntos en el tiempo acordado, mediante una de las funcionalidades que se explicarán a continuación.

7.2. Funcionalidad 1: Visualización de la información

7.2.1 Planificación para la implementación

Antes de proceder con la implementación de la funcionalidad, se ha hecho un análisis previo a la migración.

El primer paso fue decidir qué elementos, que se muestran en el Dashboard en la siguiente imagen, se mantendrían para el móvil del profesor también. Como vemos en la figura 45, tenemos 3 elementos: el botón de desactivación del juego, información sobre los iconos y la tabla en la que se muestra la información de valor.

El botón desactivación se decidió mantenerlo, puesto que es un elemento esencial para pasar el juego de activo a inactivo cuando el profesor lo considere. Por lo tanto, es necesario tenerlo en la aplicación del móvil del profesor para poder desactivar el juego desde cualquier parte y no esperar a desactivarlo una vez pueda acceder al Dashboard desde el ordenador.

El segundo elemento, que es la visualización de la información sobre los iconos, nos pareció interesante. Consideramos que era interesante mantenerlo como un elemento informativo, por si era necesario recordar cuál era el significado de los iconos que aparecen en la tabla en algún momento. Como vemos en la figura 45, en el Dashboard esta información se mostraba en todo momento por encima de la tabla, esto fue algo que se propuso cambiar, puesto que no era necesario visualizarlo todo el tiempo. La propuesta fue mostrar esta información en el móvil del profesor en forma de desplegable, donde clicando sobre un botón llamado 'Ver más' se visualizará. En el siguiente apartado, donde se explica la implementación del juego, se profundizará en detalle cómo se implementó este cambio finalmente.

Y finalmente el tercer elemento, que corresponde a la tabla donde se encuentra la información relevante e importante de este juego. Este elemento era esencial tenerlo en el móvil del profesor por su importancia, por lo tanto, no hubo duda en cuanto a su migración. Sobre el diseño se decidió mantenerlo en formato tabla debido a su fácil lectura y visualización, aunque ya se preveía un ajuste estético en cuanto la tabla debido a que la visualización se produciría en un móvil. En este momento, se tuvo en cuenta la problemática que habría en el caso de que hubiera más de x número de controles, ya que en un móvil la tabla no cabría con este formato en el que se presenta en el Dashboard. En el siguiente apartado se explicará en detalle la resolución de esta problemática.



Fig.45. Juego de control de trabajo visualizado desde el Dashboard

La figura que se ve a continuación, es la información que se despliega al clicar sobre uno de los iconos que se muestran en la tabla de la figura 46. En este caso, por el título que vemos en la imagen, sabemos que esta información que está desplegada es del icono situado en el 'Equipo B' del 'Control 1'. Como vemos se muestran los diferentes participantes del equipo B, los correspondientes puntos que han repartido a sus compañeros, y finalmente el número total de puntos que tiene cada miembro del equipo según la repartición de puntos previa. Y a la derecha tenemos los comentarios que ha hecho cada uno de los miembros del equipo en relación con el control que estamos analizando en detalle.

Tener esta información en el móvil del profesor es esencial, ya que tiene la posibilidad de acceder al detalle de cada uno de los controles que desee en cualquier lugar, como puede ser en clase. Por eso se consideró interesante migrar cada uno de los elementos que se muestran, tanto la tabla de puntos como los comentarios relacionados con el control.

Respecto al diseño, en un principio se decidió mantener algo similar, teniendo en cuenta la resolución de la problemática de la tabla aquí también. Ya que también es posible que en un equipo haya más de x miembros, y eso puede provocar que la tabla crezca y no encaje bien en el formato móvil. Pero por lo general, la visualización de elementos iba a ser bastante similar.



Fig.46. Información desplegada al clicar sobre un icono de la tabla

7.2.2 Implementación del juego

Para la implementación del juego hemos tenido que crear una página en exclusiva para visualizar la información del juego de control de trabajo, la página se ha denominado “control-de-trabajo”. Para establecer todo lo mencionado anteriormente hemos tenido que modificarlo todo, tanto en el archivo html como en el archivo ts, como en el module.ts.

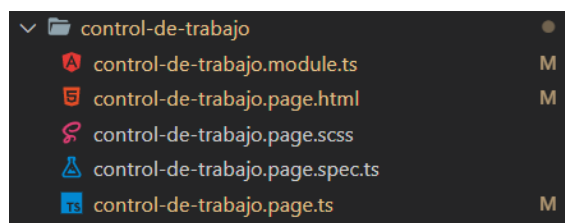


Fig.47. Estructura y archivos del juego de control de trabajo

El primero paso fue visualizar las tablas con información estática, es decir, no era la información dinámica que llega de la API que se muestra en la tabla ahora mismo. Esto se realizó modificando el archivo .html del juego. Una vez, las tablas se mostraban como habíamos planeado, pasábamos a hacer las peticiones a la API correspondientes para obtener la información necesaria.

Como se muestra en la siguiente imagen, este es el código de la primera tabla, donde se muestra el número de controles, el nombre de los equipos y los iconos que representan en el estado del control. El código marcado en azul es la columna que muestra el nombre de los equipos, en este caso, por ejemplo, serían ‘Equipo A’ y ‘Equipo B’. Y a continuación, el código marcado en verde es la columna de la tabla donde se muestran el número de control con los correspondientes iconos. Podemos ver que cada icono ejecuta la función ‘MouseOver()’ al clicar sobre él, esta función es la encargada de desplegar en detalle las puntuaciones de cada uno de los controles.

```
<table mat-table [dataSource]="dataSource" style = "width: 100%;">
  <ng-container matColumnDef="equipo">
    <th mat-header-cell class= "titulocolumnaTabla" *matHeaderCellDef> Equipo </th>
    <td mat-cell style= "text-align: left" *matCellDef="let element"> {{element.equipo.Nombre}} </td>
  </ng-container>
  <ng-container [matColumnDef]="control" *ngFor="let control of controles; let i = index">
    <th mat-header-cell class= "titulocolumnaTabla" *matHeaderCellDef> control {{i+1}} </th>
    <td mat-cell style= "text-align: left" *matCellDef="let element; let j = index">
      <i *ngIf = "controlTerminado[j][i] === 1" (click)="MouseOver(i, element)" class="material-icons" style = "color: green"> done</i>
      <i *ngIf = "controlTerminado[j][i] === 2" (click)="MouseOver(i, element)" class="material-icons" style = "color: red"> done</i>
      <i *ngIf = "controlTerminado[j][i] === 3" (click)="MouseOver(i, element)" class="material-icons" style = "color: black"> hourglass_bottom</i>
    </td>
  </ng-container>
  <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
  <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
</table>
```

Fig.48. Parte del código del archivo html del juego de control de trabajo

En la siguiente figura, se muestra el código de la tabla que se despliega cuando se hace clic sobre el icono. La sección azul es la columna donde muestra el nombre y apellido de cada uno de los miembros de ese equipo y el icono del email para enviar un recordatorio al alumno (funcionalidad que se explica en el siguiente apartado). La sección verde es la columna donde se muestra la repartición de puntos de cada uno de los miembros a sus compañeros de equipo, y donde finalmente se calcula la suma total de puntos que tiene cada miembro.

```

<table mat-table [dataSource]="dataSourceControl" style = "width: 100%" >
  <ng-container matColumnDef="alumno">
    <th mat-header-cell class= "tituloColumnaTabla" *matHeaderCellDef> </th>
    <td mat-cell style= "text-align: left" *matCellDef="let alumno">
      {{alumno.Nombre}} {{alumno.PrimerApellido}}

      <i (click)="EnviarEmail(alumno.Nombre);" class="material-icons" style = "color: blue">mail</i>

      <i class="material-icons" style = "color: black">forward</i>
    </td>
    <td mat-footer-cell *matFooterCellDef style= "text-align: left; font-size:large; font-weight: bold ;"> Total </td>
  </ng-container>

  <ng-container [matColumnDef]="nombre" *ngFor="let nombre of nombresMiembros; let i = index">
    <th mat-header-cell class= "tituloColumnaTabla" *matHeaderCellDef style= "text-align: center" > {{nombre}} </th>
    <td mat-cell style= "text-align: center" *matCellDef="let element; let j = index" >
      <span *ngIf = "datosControl[j]"> {{datosControl[j].puntuaciones[i].puntos}}</span>
    </td>
    <td mat-footer-cell *matFooterCellDef style= "text-align: center; font-size:large; font-weight: bold ;">
      {{ Suma (i)}}
    </td>
  </ng-container>

  <tr mat-header-row *matHeaderRowDef="displayedColumnsControl"></tr>
  <tr mat-row *matRowDef="let row; columns: displayedColumnsControl;"></tr>
  <tr mat-footer-row *matFooterRowDef="displayedColumnsControl"></tr>
</table>

```

Fig.49. Parte del código del archivo html del juego de control de trabajo

Para obtener la información que introducimos en cada uno de los campos mencionados anteriormente, programamos las funcionalidades en el archivo .ts del juego de control de trabajo.

El primer paso fue decidir qué información queríamos obtener nada más abrir la página del juego y, por lo tanto, que se ejecutaran esas funciones, y esto se programa en el 'ngOnInit()' del archivo. Las funciones más destacables que se implementaron, fueron 'this.sesion.DameJuego()', donde se llama a la funcionalidad que se encuentra en el Service nombrado "sesión", llamada 'DameJuego()'. Esta funcionalidad realiza una petición directamente a la API, para obtener el juego seleccionado en el que estamos. A partir de allí, mediante la función 'this.peticionesAPI.DameEquiposDelGrupo(this.juegoSeleccionado.grupold)' obtenemos los equipos que necesitamos, a partir Id de grupo que obtuvimos previamente al obtener el juego seleccionado. Gracias a estas dos funciones, podemos mostrar los equipos en la tabla del grupo con el que estamos trabajando.

A continuación, implementamos una función llamada 'TraeRespuestasAlControl()' donde se prepara el vector con todas las respuestas de los equipos. El vector tendrá una posición por cada equipo, donde habrá un objeto con el equipo y las respuestas del equipo. Hay que tener en cuenta que las respuestas del equipo, es un vector con tantas posiciones como miembros tiene el equipo y en cada una de esas posiciones hay un vector con las respuestas de cada alumno a los controles.

Cada respuesta a un control es un objeto con dos campos: puntuaciones y comentario. El campo de puntuaciones es un vector con tantas posiciones como alumnos tiene el grupo, y en cada posición hay el id del alumno y los puntos que se le han asignado. Gracias a esta información que recibimos de la API podemos rellenar la mayor parte de los campos de las tablas mencionados previamente.

Dentro de esta función, a partir de la matriz 'numeroDeRespuestas[i][j]', que nos dice cuántos controles ha contestado ya el alumno j del equipo i. Creamos la matriz 'numeroRespuestasPorControl[i][k]' que indica cuantos alumnos del equipo i han contestado ya el control k. Gracias a esta información podemos mostrar el estado de cada uno de los controles mediante iconos. Los requisitos para la decisión de mostrar un estado u otro son los siguientes:

1. Se considera que un control no ha tenido ningún problema cuando ningún miembro ha recibido de alguno de sus compañeros un número de puntos inferior a la mitad de lo que hubiera sido un reparto perfecto de los 10 puntos.

En ese caso se mostrará el icono de check en color verde, que indica que todos los miembros han completado el control.

2. Se considera que un control indica posibles problemas si alguno de los alumnos ha recibido de alguno de sus compañeros un número de puntos inferior a la mitad de lo que hubiera sido un reparto perfecto de los 10 puntos.

Por ejemplo, si el grupo es de 4 alumnos, el reparto perfecto es 2.5 para cada uno. Habrá problemas si un alumno recibe menos de 1.25 puntos por parte de algún compañero.

En ese caso se mostrará el icono de check en color rojo, que indica que todos los miembros han completado el control, pero que algún miembro ha recibido una nota inferior respecto a los demás.

3. Se considera que un control es incompleto cuando faltan miembros del grupo por evaluar. En ese caso se mostrará el icono de reloj de arena.

En la matriz 'controlTerminado [i][k]' indicaremos el estado del control k del equipo i, con estos códigos:

- 0 control no contestado aun nadie del grupo
- 1 control contestado por todos y sin problemas
- 2 control contestado por todos, pero con problemas
- 3 control contestado solo por algunos del grupo

Con toda esta información recibida de la API podemos rellenar de forma dinámica la primera tabla que queremos visualizar, con el nombre de los equipos, el número de controles correspondiente con cada uno de los iconos indicando su estado.

El segundo paso, es obtener la información necesaria para rellenar la tabla que se despliega al clicar sobre uno de los iconos de estado, y que muestra todos los detalles de puntuación de cada uno de los miembros.

La primera función a implementar es la del `'MouseOver()'`, función que hemos visto previamente en el archivo `.html` para desplegar los detalles del control al clicar sobre el icono. A partir del número del control y el nombre del equipo, obtenemos todos los alumnos de ese equipo mediante una petición a la API. A partir de esto, llamamos a otra función que tenemos creada en el archivo `.ts`, llamada `'PreparaDatosDelControl'` donde se preparan las puntuaciones y los comentarios de los alumnos del equipo identificado al contestar el control `i`. La respuesta es un vector con una posición para cada alumno del grupo, donde posteriormente recorreremos todos los alumnos y tomamos sus respuestas en el control elegido. En ese momento, también es donde se hacen la comprobación de si el alumno `x` ha puntuado o comentado algo en relación con ese control.

Y gracias a esta información recibida de la API podemos rellenar de forma dinámica la segunda tabla que queremos visualizar, con el nombre de los miembros del equipo, las puntuaciones de cada uno de ellos, el total de puntos recibidos de cada alumno y los comentarios respecto a la repartición de esos puntos.

Finalmente, para llevar a cabo el uso de las anteriores herramientas ha sido necesario importar en el `module.ts` ciertas librerías para la correcta visualización de todos los elementos descritos anteriormente:

```
import { FormsModule } from '@angular/forms';
import { MatTableModule } from '@angular/material/table';
import { MatExpansionModule } from '@angular/material/expansion';
import { MatListModule } from '@angular/material/list';
```

El código de las funcionalidades mencionadas se encuentra en el repositorio de GitHub llamado `'classpip-movil-profesor-dev'`.

7.2.3 Retos destacados

Se dedicó mucho tiempo a pensar como mostrar la información de una manera clara y útil para el docente. Al principio se planteó sustituir las tablas por expandibles, donde fuera posible expandir la información de cada equipo. Para ello, se importó un componente que se creó previamente para utilizarlo en el juego de competición, pero empezó a dar problemas de duplicación, puesto que este componente ya era utilizado en el juego de competición.

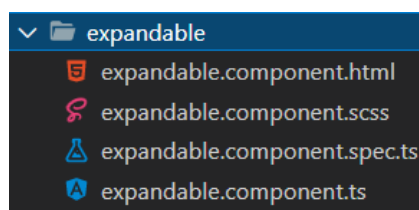


Fig.50. Estructura y archivos del componente llamado expandible

Esto nos obligaba a duplicar el componente y cambiarle el nombre para poder ser utilizado en el juego de control de trabajo sin ningún problema relacionado con la duplicación. Pero esta solución nos parecía muy poco óptima, ya que se tenía que duplicar este componente en cada uno de los juegos en los que lo queríamos utilizar. Aparte, que al añadirlo no nos convenció la manera en la que se veía todo estéticamente. Por ello se decidió mantener las tablas, aunque las tablas también nos generaban problemas al pasarlas al tamaño móvil si eran muy anchas, ya que se cortaba parte de la información y no se veía.

La solución que se propuso finalmente sobre la problemática con las tablas, fue añadir un estilo al 'div' que englobaba la tabla que queríamos arreglar.

Este estilo era el 'overflow-x', donde esta propiedad específica si se recorta el contenido, se agrega una barra de desplazamiento o se muestra el contenido de desbordamiento de un elemento a nivel de bloque, cuando se desborda en los bordes izquierdo y derecho. Al especificar que queríamos que la propiedad fuera 'auto', en ese momento se debe proporcionar un mecanismo de desplazamiento para las celdas que se desbordan.

```
<div *ngIf = "tablaPreparada" style="width: 100%; overflow-x: auto;">
```

Fig.51. Style añadido para solucionar la problemática con las tablas

Por lo tanto, finalmente obtenemos un 'scrolling' horizontal que nos permite visualizar la parte de la tabla que previamente se cortaba por el tamaño del dispositivo.

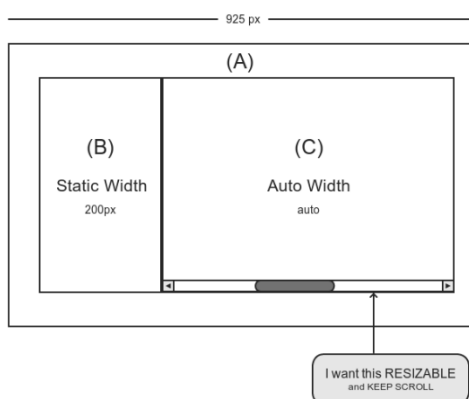


Fig.52. Ejemplo del scrolling horizontal que obtenemos mediante overflow-x: auto

7.3. Funcionalidad 2: Envío de correos electrónicos

7.3.1 Planificación para la implementación

Desde el análisis de cada uno de los juegos, teníamos claro que queríamos añadir una funcionalidad adicional que no existía en el Dashboard. Al realizar el análisis en detalle del juego de control de trabajo, vimos interesante la implementación de una funcionalidad para recordar al alumno la necesidad de completar la asignación de puntos en el tiempo predeterminado por el docente. En este caso, consideramos que el envío de un correo electrónico en formato recordatorio era una idea interesante a añadir como funcionalidad adicional.

El objetivo era conseguir que cuando el profesor desplegara un control con icono de reloj de arena (Indica que solo algunos miembros del equipo han completado la evaluación) y se vieran los detalles de puntuación de cada uno de los miembros de equipo, que el docente tuviera la posibilidad de recordar a los alumnos que tienen el apartado de puntuación vacío, la necesidad de evaluar a los miembros del grupo en el plazo preestablecido. Ya que previamente en el Dashboard esa posibilidad no existía y si el docente visualizaba que una cantidad de alumnos no lo habían completado, tenía que informarlo personalmente o enviando un comunicado de manera independiente a Classpip. Añadiendo esta funcionalidad, se facilita al docente la comunicación con sus alumnos en este tipo de casos.

De cara a que tipo de tecnología se iba a utilizar para dicha implementación, al realizar el análisis general de la aplicación del profesor al inicio, observamos que estaba implementada la funcionalidad de 'Recuperación de contraseña' en la página de inicio de sesión del profesor. Esta funcionalidad también realizaba el envío de un mail cuando el profesor solicitaba la recuperación de la contraseña. Por lo tanto, se decidió utilizar la misma tecnología, que era 'Nodemailer', para mantener la semejanza del código y también para no tener dos implementaciones de envío de mails distintas y que conlleven a confusión.

En cuanto al proveedor, desde un principio se decidió utilizar Gmail. Las razones de esta decisión fueron varias:

1. Homogeneidad: La funcionalidad previamente implementada, que utilizaba la tecnología de nodemailer, utilizaba este proveedor. Y Classpip como producto, cambiar constantemente de proveedor no era una solución cómoda de cara a la experiencia del usuario. Por ello, para mantener la homogeneidad se decidió utilizar Gmail en todos los casos en los que se implemente el envío de correos electrónicos.
2. Documentación: La información de la implementación de Gmail con Nodemailer estaba mucho más documentada en internet que cualquier otra opción. Eso significaba que cualquier problema al que nos íbamos a enfrentar, teníamos más información y armas con las que solucionarlo.
3. Seguridad: La seguridad que nos ofrece Gmail respecto a otros proveedores era mucho mayor, por lo que era un punto que era interesante si en algún momento la seguridad en cuanto al envío de mail tuviera que incrementarse.

4. Compatibilidad: Gmail es uno de los proveedores más compatibles con cualquier tecnología STMP que se quiera implementar en un proyecto de programación, en este caso, era bastante compatible con Nodemailer. Además, investigando sobre la combinación de uso de Gmail y Nodemailer, las opiniones eran muy positivas.

En el capítulo 10, en el apartado donde se entra en detalle de las problemáticas que ha conllevado implementar esta funcionalidad, se explica por qué se decidió cambiar de proveedor en un momento dado. Aunque finalmente, se acabó usando Gmail como proveedor en este proyecto.

Una vez sabíamos que tecnología íbamos a utilizar, pasamos a la parte estética de esta funcionalidad. Decidimos añadir la funcionalidad en forma de icono representando el sobre de un correo electrónico. Se decidió colocarlo en la tabla de detalles de puntos, ya que es exactamente donde obtenemos la información de que alumnos falta por votar. Además, se decidió ponerlo al lado del nombre para que de una manera más visual se sepa a qué alumno se le está enviando ese recordatorio.

Una vez teníamos planeada como queríamos la implementación de la funcionalidad de envío de correos electrónicos, el siguiente paso era programarlo.

7.3.2 Implementación del juego

El primer paso es crear el botón con el icono correspondiente en el archivo .html del juego de control de trabajo. Como se observa en la siguiente figura, al hacer clic sobre el icono se ejecutará la función de 'EnviarEmail()' donde se pasa como parámetro el nombre de usuario del alumno al que queremos enviar el recordatorio.

```
<ng-container matColumnDef="alumno">
  <th mat-header-cell class=" tituloColumnaTabla" *matHeaderCellDef> </th>
  <td mat-cell style=" text-align: left" *matCellDef="let alumno">
    {{alumno.Nombre}} {{alumno.PrimerApellido}}
    <i (click)="EnviarEmail(alumno.Nombre);" class="material-icons" style = "color: blue">mail</i>
    <i class="material-icons" style = "color: black">forward</i>
  </td>
  <td mat-footer-cell *matFooterCellDef style= "text-align: left; font-size:large; font-weight: bold ;"> Total </td>
</ng-container>
```

Fig. 53. Parte del código del archivo html del juego de control de trabajo

La función que hemos visto previamente que se ejecuta al hacer clic al botón del email, se encuentra en el archivo .ts del juego de control de trabajo. Esta función, como vemos en la siguiente figura, le llega como parámetro el nombre de usuario del alumno y a continuación, se hace una petición a la API para recuperar el objeto 'Alumno' al que corresponde ese nombre de usuario pasado como parámetro previamente (Función marcada en azul).

En ese momento el objeto 'Alumno' lo tenemos en la variable 'res', donde comprobamos si esa variable es 'undefined' o no, y depende del resultado, se muestra una notificación en la pantalla del usuario. En el caso de que todo vaya bien, se mostrara una notificación que indica que el 'Recordatorio ha sido enviado', por el contrario, indicara otro mensaje.

Si todo va bien, nos vamos a conectarnos al servidor para realizar el envío del email, esto lo vamos a realizar con la función marcada en verde en la figura. Donde se pasa como parámetro el email que hemos obtenido a partir del objeto 'Alumno'. Esta función se explica con más detalle a continuación.

```

async EnviarEmail(nombreAlumno: string) {
  this.peticionesAPI.DameAlumnoEmail (nombreAlumno)
  .subscribe (async (res) => {
    if (res[0] !== undefined) {
      const email = res[0].Email;
      this.comServer.EnviarRecordatorio(email);
      const alert = await this.alertController.create({
        header: 'Recordatorio enviado',
        buttons: ['OK']
      });
      await alert.present();
    } else {
      const alert = await this.alertController.create({
        header: 'No hay ningun alumno con este email',
        buttons: ['OK']
      });
      await alert.present();
    }
  });
}

```

Fig.54. Función enviar email situada en el archivo .ts del juego de control de trabajo

La siguiente figura corresponde a la función que se ejecuta dentro de la función 'EnviarEmail()' (función marcada en azul en la figura 54), donde recuperamos el objeto 'Alumno' a partir de una petición a la API donde filtramos mediante el nombre de usuario.

```

public DameAlumnoEmail(nombreAlumno: string): Observable<Alumno> {
  return this.http.get<Alumno>(this.APIUrlAlumnos + '?filter[where][Username]=' + nombreAlumno );
}

```

Fig.55. Función de la petición a la API del alumno mediante el nombre de usuario

Y la siguiente figura corresponde también a la función que se ejecuta dentro de la función 'EnviarEmail ()' (función marcada en verde en la figura 54), donde nos conectamos al servidor mediante sockets para emitir el evento 'enviarRecordatorio' y como datos enviamos el email del alumno. Una vez enviado el email, nos desconectamos.

```
public EnviarRecordatorio(email: string) {  
  this.servidor.connect();  
  this.servidor.emit ('enviarRecordatorio' , {email: email});  
  this.servidor.disconnect();  
}
```

Fig.56. Conexión con el servidor

Una vez tenemos todo listo en la parte del cliente, pasamos al servidor para recibir el evento que acabamos de emitir. Como vemos en la siguiente imagen, el servidor recibe el evento mediante socket y realiza una petición a la API para llevar a cabo el envío de email al usuario correspondiente.

```
socket.on("enviarRecordatorio", (datos) => {  
  peticionesAPI.EnviaRecordatorio(datos.email);  
});
```

Fig.57. Función en la que el servidor recibe el evento enviado por el cliente

A continuación, se ejecuta la función que se ha realizado mediante la petición a la API. Esta función es la que realiza el envío del mail utilizando Nodemailer y Gmail como proveedor. Como vemos, el emisor siempre va a ser Classpip mediante su email de empresa donde comunicara este tipo de recordatorios. El destinatario será el alumno que se pasa su email como parámetro. Y la información que va dentro del correo electrónico, en este caso, es una plantilla en formato recordatorio a la que se le envía a cualquier alumno que decida el profesor.

Como vemos en la imagen, hay dos campos marcados, el campo marcado en azul es donde irá colocado el email de empresa de Classpip con el que se va a enviar todo este tipo de emails. Y el campo marcado en verde, es la contraseña de aplicación, que no es la contraseña con la que se inicia sesión normalmente. Para una correcta configuración y obtención de esta contraseña se explica en el capítulo de 'Retos destacados' en el apartado 5.1.3. Y finalmente, se envía el mail correctamente, y en el caso se algún problema saltaría un mensaje de error.


```

public EnviarRecordatorio(email: string) {
  console.log ('Estoy dentro de EnviarEmail sumaya');
  console.log ('creo las opciones');
  const mailOptions = {
    from: "Classpip",
    to: email, //Cambia esta parte por el destinatario
    subject: "TU VALORACIÓN ES IMPORTANTE: Control de trabajo",
    html: "Estimado/a estudiante <br><br>" +
      "Un elemento muy importante para analizar el progreso de los trabajos en grupo es  

      "¡Muchas gracias!",
  };

  const transporter = nodemailer.createTransport({
    auth: {
      user: "firefighteradventure@gmail.com", // Cambialo por el email de classpip
      pass: "guwupytirqcdetbk" //Cambialo por tu contraseña de aplicación
    },
    service: "gmail",
  });

  transporter.sendMail(mailOptions, function(err, info) {
    if (err) {
      console.log(err);
    } else {
      console.log(info);
    }
  });
}

```

Fig. 58. Función de envío de correos electrónicos en el servidor

Una vez programado esto, si clicamos el icono de envío de email a un cierto usuario, recibiremos un correo electrónico igual al que se muestra en la siguiente figura.

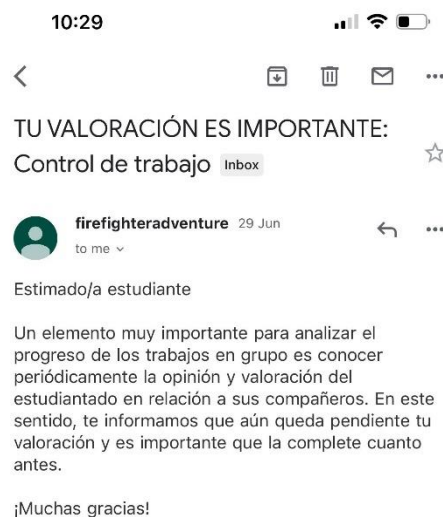


Fig. 69. Correo que se envía a los alumnos en forma de recordatorio

7.3.3 Retos destacados

La implementación de esta funcionalidad fue uno de los retos más destacados del trabajo en general. Por eso, se explica con mucho más detalle en el capítulo 10 denominado 'Retos más destacados'.

También uno de los retos, fue entender el funcionamiento de los sockets en esta funcionalidad. Una vez, teníamos implementada correctamente la funcionalidad de envío de emails y ya se realizaba correctamente el envío del mismo, tuvimos problemas para enviar de forma inmediata otro correo al acabar de enviar uno. Este problema nos dimos cuenta de que estaba relacionado con la conexión y desconexión del socket, y el evento que estábamos emitiendo. Como resultado, eliminamos la desconexión del socket que se realizaba inmediatamente después de emitir el evento, y esto soluciono el problema. Por lo tanto, finalmente la desconexión del socket se realizaba al salir de la página del juego de control de trabajo. Gracias a esta solución, era posible enviar un email al Usuario 1 e inmediatamente enviar otro email al Usuario 2.

7.4. Testing del correcto funcionamiento

En este punto, se explicará de forma detallada como se comprobó el correcto funcionamiento del juego, una vez implementado a nivel de código. Se realiza mediante una guía por pasos a seguir, que se ejecutó y realizó en su momento y que cualquier usuario con acceso al mismo puede realizar para su comprobación.

La comprobación del correcto funcionamiento del juego, comienza con la creación de un juego nuevo mediante el Dashboard, cosa que ya se ha mencionado y explicado en capítulos anteriores, por ello en este punto se comenzara la comprobación desde el punto de vista del docente, usando la aplicación del profesor.

1. Abrimos la aplicación del profesor.
2. Introduciendo un usuario y su respectiva contraseña en el formulario.
3. Accedemos a la propia aplicación, pulsamos el botón de "Log In".
4. Deslizamos hasta encontrar el juego seleccionado, para ello se dispone de botones para adelante y para atrás.
5. Accedemos al juego en cuestión pulsando en el botón "Ver más".



Fig.60. Página de inicio de sesión



Fig.70. Tarjeta del juego de control de trabajo

Una vez estamos dentro del juego de control de trabajo, en la figura 71, podemos visualizar los equipos que tenemos del grupo en el que estamos y los diferentes controles con los estados correspondientes. También tenemos un botón denominado 'Más información', donde si clicamos se despliega la información del significado de los iconos que se muestran en la tabla.

Haz click sobre el control que quieres examinar



Equipo	Control 1	Control 2	Control 3	Control 4
Equipo A	✓	🕒	🕒	🕒
Equipo B	✓	✓		

MÁS INFORMACIÓN

Fig.71. Tabla principal con la información de los controles de trabajo

Para examinar en detalle la información de cada uno de los controles, es necesario clicar encima del icono para desplegar la información. Como vemos en la figura 72, se despliega al detalle la puntuación que ha dado cada miembro del equipo a sus compañeros, y también podemos ver la suma total de puntos que ha obtenido cada miembro. Y en la parte de abajo, se despliegan los comentarios que han hecho cada miembro del equipo respecto al control que han evaluado.

Control 1 del equipo **Equipo A**

	e1	e2	e3	e4
e1 e1  	4	2	1	3
e2 e2  	3	3	2	2
e3 e3  	3	3	3	1
e4 e4  	7	1	1	1
Total	17	9	7	7

Comentarios

e1
Todo Ok, de part...

e2
Olee

Fig.72. Información detallada de uno de los controles

Otra de las funcionalidades que se pueden realizar en este juego, es el envío de correos electrónicos a los alumnos que falten por votar, donde el control se muestra que aún está pendiente. Al clicar sobre el icono en forma de sobre, automáticamente se envía el mail al alumno seleccionado, y aparecerá en pantalla la notificación que se muestra en la figura 73.

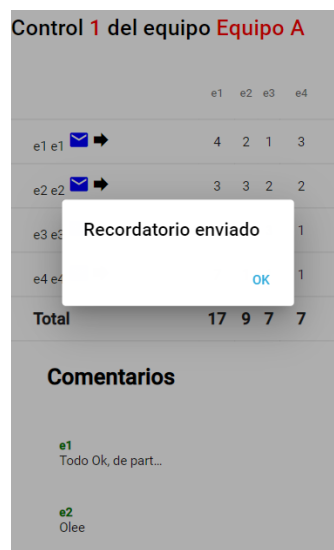


Fig.73. Envío del correo electrónico de forma exitosa

CAPÍTULO 8. JUEGO DE CUESTIONARIO DE SATISFACCIÓN

En este capítulo se explicará en qué consiste el juego de cuestionario de satisfacción, las diferentes funcionalidades implementadas en el juego, que estrategia o planificación se utilizó para su implementación, los retos más destacados a lo que me he enfrentado y finalmente las soluciones que se aplicaron según el reto presentado.

Este capítulo es totalmente diferente al de mi compañero de proyecto, puesto que en este capítulo cada uno explica el juego que ha implementado de forma independiente.

8.1. ¿En qué consiste el juego de cuestionario de satisfacción?

El juego de cuestionario de satisfacción consiste en una herramienta del profesor para conocer la opinión de los alumnos respecto a un tema que el docente decida. En este caso, va enfocado a una opinión relacionada con la satisfacción que sienten los alumnos respecto a un tema relacionado con las clases, metodología del docente, interés sobre una asignatura o tema en concreto...

En la aplicación, el docente podrá saber el grado de satisfacción de sus alumnos mediante la creación de un número de afirmaciones, donde los alumnos responderán con una nota del 1 al 5 (5 es el grado de satisfacción máximo) indicando cuanto de satisfechos están con las situaciones/afirmaciones que el profesor pregunta. El profesor también tiene la opción de hacer preguntas abiertas, y en ese caso los estudiantes deberán responder con un comentario en escrito.

Una vez respondidas las preguntas, se mostrará los resultados en formato tabla, indicando que media ha obtenido cada una de las afirmaciones escritas por el profesor. También, tiene la opción de visualizarlo en formato gráfico, donde el eje y serán las afirmaciones y el eje x la media de votos. Respecto a las preguntas abiertas, se mostrarán en forma de desplegable. Teniendo la opción de desplegar todas las respuestas de cada una de las preguntas abiertas realizadas.

El profesor también podrá visualizar el número de participantes de la encuesta y el número de respuestas en forma de porcentaje, que eso indicará el grado de participación de los alumnos. Y en el caso del Dashboard, también existe la opción de generar y descargarse un pdf con todos los datos anteriormente mencionados en forma de resumen.

Gracias a la información proporcionada por esta herramienta, el profesor será capaz de saber cuál es el grado de satisfacción de sus alumnos y podrá mejorar en las cosas que considere. Si el docente recibe una calificación muy baja en una de sus afirmaciones o preguntas abiertas, podrá hacer autocrítica y mejorar en ese aspecto. Esto puede ser muy beneficioso para el rendimiento de sus alumnos, ya que se sentirán escuchados por el docente y verán cambios en las áreas que consideran que no les motivan tanto como otras.

8.2. Funcionalidad 1: Visualización de la información

8.2.1 Planificación para la implementación

Antes de proceder con la implementación de la funcionalidad, se ha hecho un análisis previo a la migración.

El primer paso fue decidir qué elementos, que se muestran en el Dashboard se mantendrían para el móvil del profesor también. Tenemos 5 elementos: el botón de desactivación del juego, la generación del informe en pdf, la tabla y la gráfica que muestran los resultados de las afirmaciones, por último, los desplegados que muestran las respuestas de las preguntas abiertas. A continuación, se explicará en detalle las razones por las que hemos decidido prescindir de algunos elementos.

El botón desactivación, que está marcado en verde en la figura 74, se decidió mantenerlo, puesto que es un elemento esencial para pasar el juego de activo a inactivo cuando el profesor lo considere. Por lo tanto, es necesario tenerlo en la aplicación del móvil del profesor para poder desactivar el juego desde cualquier parte y no esperar a desactivarlo una vez pueda acceder al Dashboard desde el ordenador.

El segundo elemento, que está marcado en azul en la figura 74, que es el botón para generar y descargar el informe de las respuestas en PDF, decidimos no mantenerlo para el móvil del profesor. Ya que no era necesario tenerlo disponible y descargarlo en el móvil, con descargarlo en su despacho en el ordenador desde el Dashboard, consideramos que es suficiente.

El tercer elemento, que está marcado en rojo en la figura 74, que es la información que nos indica la cantidad de participantes y números de respuestas de esa encuesta. Se consideró que era un elemento interesante de mantener, para que el profesor tuviera la posibilidad de acceder a esa información en cualquier lugar. Es interesante poder tener esta información a mano, por si el docente lo quiere revisar en clase de forma rápida y poder recordar a los alumnos que completen la encuesta para subir el grado de participación.

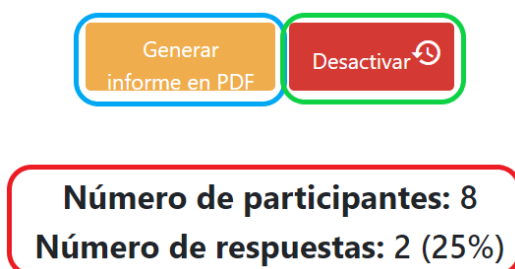


Fig.74. Primera sección del juego de cuestionario de satisfacción en el Dashboard

El cuarto elemento, que corresponde a la tabla donde se encuentra la información relevante e importante de este juego. Este elemento era esencial tenerlo en el móvil del profesor por su importancia, por lo tanto, no hubo duda en cuanto a su migración. Pero en este caso se decidió mantener solamente la tabla, debido a su fácil lectura y visualización. Se descartó mantener la gráfica porque esa información realmente ya se mostraba en la tabla, y era un elemento innecesario.

Respuestas a las afirmaciones



Fig. 75. Segunda sección del juego de cuestionario de satisfacción en el Dashboard

Y finalmente, el quinto elemento, que son los desplegados, que muestran las respuestas desarrolladas de los alumnos respecto a las preguntas abiertas que realizó el profesor. Es un elemento tan esencial como el anterior, ya que es donde se muestra la información relevante e interesante para el docente. Respecto al diseño, se decidió mantenerlo en formato desplegable, puesto que era cómodo para el formato móvil por si las respuestas de los alumnos fueran demasiado extensas. Así, el profesor tenía la posibilidad de desplegar las respuestas de las preguntas que le interesaran en ese momento, y evitar la saturación de texto en la página de la aplicación.

Respuestas a las preguntas abiertas

La figura muestra un desplegable con una pregunta abierta y sus respuestas. La pregunta es '¿Cuál ha sido la incidencia crítica más positiva?' y el desplegable está abierto, mostrando 'Respuestas' y un 'x'.

¿Cuál ha sido la incidencia crítica más positiva? ^

Respuestas

x

¿Cuál ha sido la incidencia crítica más negativa? v

Fig. 76. Tercera sección del juego de cuestionario de satisfacción en el Dashboard

8.2.2 Implementación del juego

Para la implementación del juego hemos tenido que crear una página en exclusiva para visualizar la información del juego de cuestionario de satisfacción, la página se ha denominado “juego-cuestionario-satisfacción”. Para establecer todo lo mencionado anteriormente hemos tenido que modificarlo todo, tanto en el archivo html como en el archivo ts, como en el module.ts.

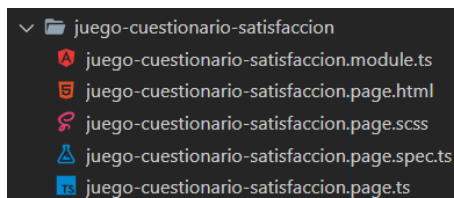


Fig. 77. Estructura y archivos del juego de cuestionario de satisfacción

El primer paso fue visualizar la información del número de participantes y el número de respuestas. Esto se realizó modificando el archivo .html del juego. Una vez, se mostraba la información de forma estática, se pasó a hacer las peticiones a la API correspondientes para obtener la información necesaria. Esto fue algo que se añadió en el archivo .ts, que se explicará a continuación.

```
<ion-list *ngIf = "informacionPreparada" lines = "none" style="text-align: center;">
  <label style="text-align: center;"><h2 style="color: rgb(109, 11, 19)"><strong>Datos generales</strong></h2></label>
  <div>
    <h4><strong>Número de participantes</strong></h4>
    <h3>{{inscripciones.length}}</h3>
  </div>
  <div>
    <h4><strong>Número de respuestas actuales </strong></h4>
    <h3>{{numeroRespuestas}} ({{{(numeroRespuestas/inscripciones.length)*100}.toFixed(0)}}%)</h3>
  </div>
</ion-list>
```

Fig. 78. Parte del código del archivo html del juego de cuestionario de satisfacción

A continuación, el objetivo era visualizar la tabla con la información relevante de la encuesta. Esa información eran las afirmaciones que el profesor había creado previamente y la nota media de puntuación del 1 al 5 de los alumnos que habían votado en ese momento. Para hacerlo, añadimos el código que se muestra a continuación en el archivo .html. Al principio se creó con información inventada estática para comprobar la correcta visualización de la tabla y ya posteriormente, se creó la función en el archivo .ts para hacer las peticiones necesarias a la API para obtener esa información y rellenar los campos de la tabla.


```

<ion-list style="text-align: center;">
  <label astyle="text-align: center;"><h1 style="color: rgb(109, 11, 19)"><strong>Afirmaciones</strong></h1></label>
  <br>
  <ion-item *ngFor="let afirmacion of afirmaciones;">
    <ion-item lines = "none">
      <ion-label class="ion-text-wrap" style = "font-style: italic;">{{afirmacion.Texto}} </ion-label>
    </ion-item>
    <ion-item lines = "none" style="text-align: right;">
      <ion-label style = " color: rgb(11, 105, 27); text-align: right;">{{afirmacion.Media.toFixed(2)}} </ion-label>
    </ion-item>
  </ion-item>
</ion-list>

```

Fig. 79. Parte del código del archivo html del juego de control de trabajo

Y como paso final en lo relacionado con el archivo .html, se añaden los desplegados con las preguntas abiertas y con sus correspondientes respuestas. Como hemos mencionado anteriormente, se hacen pruebas primero con información estática y ya después se crea la función en el archivo .ts para recuperar la información dinámica necesaria.

```

<ion-list style="text-align: center;">
  <label style="text-align: center;"><h1 style="color: rgb(109, 11, 19)"><strong>Preguntas abiertas</strong></h1></label>
  <br>
  <ion-item><h3><strong> {{cuestionario.PreguntasAbiertas[i]}}</strong></h3></ion-item>
  <ion-item *ngFor="let respuesta of respuestas">
    <ion-label class="ion-text-wrap" style = "font-style: italic;"> {{respuesta}} </ion-label>
  </ion-item>
</ion-list>

```

Fig. 80. Parte del código del archivo html del juego de control de trabajo

Para obtener la información que introducimos en cada uno de los campos mencionados anteriormente, programamos las funcionalidades en el archivo .ts del juego de cuestionario de satisfacción.

El primer paso fue decidir qué información queríamos obtener nada más abrir la página del juego y, por lo tanto que, se ejecutaran esa funciones, y esto se programa en el 'ngOnInit()' del archivo. Las funciones más destacables que se implementaron, fueron 'this.sesion.DameJuego()', donde se llama a la funcionalidad que se encuentra en el Service nombrado "sesión", llamada 'DameJuego()'. Esta funcionalidad realiza una petición directamente a la API, para obtener el juego seleccionado en el que estamos. A partir de allí, mediante la función 'this.peticionesAPI.DameCuestionarioSatisfaccion(this.juegoSeleccionado.cuestionarioSatisfaccionId)' obtenemos el cuestionario de satisfacción que necesitamos mostrar, a partir Id del cuestionario de satisfacción que obtuvimos previamente al obtener el juego seleccionado. Gracias a estas dos funciones, podemos mostrar la información necesaria que vayamos necesitando sobre ese cuestionario.

A continuación, implementamos una función llamada 'RecuperarInscripcionesAlumnoJuego()', donde se obtienen las respuestas de cada uno de los alumnos. Gracias a esta información que recibimos de la API podemos rellenar la mayor parte de los campos de las tablas mencionados previamente.

A continuación, preparamos la información para poder visualizarla en cada uno de los sitios, eso lo hacemos con la función llamada 'PrepararInformacion()'.

Finalmente, para llevar a cabo el uso de las anteriores herramientas ha sido necesario importar en el `module.ts` ciertas librerías para la correcta visualización de todos los elementos descritos anteriormente:

```
import { FormsModule } from '@angular/forms';
import { MatStepperModule } from '@angular/material/stepper';
import { MatInputModule, MatRadioModule } from '@angular/material';
import { MatTableModule } from '@angular/material/table'
```

El código de las funcionalidades mencionadas se encuentra en el repositorio de GitHub llamado 'classpip-movil-profesor-dev'.

8.2.3 Retos destacados

Para la implementación de este juego no hubo retos demasiado destacables como para comentarlos en esta sección. Gracias a todos los retos que se presentaron en los juegos anteriores, pudimos resolver la mayoría de complicaciones que surgieron en este juego sin bastantes problemas. Por eso, cabe destacar que, gracias a todas las complicaciones previas a la implementación de este juego, nuestras habilidades como programadores aumentaron y teníamos experiencia en resolver errores que previamente ya se habían resultado.

8.3. Testing del correcto funcionamiento

En este punto, se explicará de forma detallada como se comprobó el correcto funcionamiento del juego, una vez implementado a nivel de código. Se realiza mediante una guía por pasos a seguir, que se ejecutó y realizó en su momento y que cualquier usuario con acceso al mismo puede realizar para su comprobación.

La comprobación del correcto funcionamiento del juego, comienza con la creación de un juego nuevo mediante el Dashboard, cosa que ya se ha mencionado y explicado en capítulos anteriores, por ello en este punto se comenzara la comprobación desde el punto de vista del docente, usando la aplicación del profesor.

1. Abrimos la aplicación del profesor.
2. Introduciendo un usuario y su respectiva contraseña en el formulario.
3. Accedemos a la propia aplicación, pulsamos el botón de "Log In".
4. Deslizamos hasta encontrar el juego seleccionado, para ello se dispone de botones para adelante y para atrás.
5. Accedemos al juego en cuestión pulsando en el botón "Ver más".

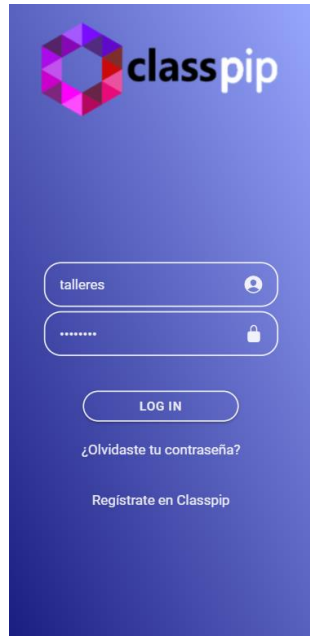


Fig. 81. Página de inicio de sesión



Fig. 82. Tarjeta del juego de cuestionario de satisfacción

Una vez estamos dentro del juego de cuestionario de satisfacción, en la figura 83, podemos visualizar los datos y estadísticas de dicho cuestionario. Los datos que se muestra son, el número de participantes, el número de respuestas actuales en formato número y en porcentaje, las respuestas a las afirmaciones que el docente ha escrito, y finalmente, se muestran las preguntas abiertas que ha propuesto el docente. Como se puede observar, las respuestas a las preguntas abiertas, se mostrarán una vez se haga clic sobre el desplegable de la pregunta que queramos abrir.

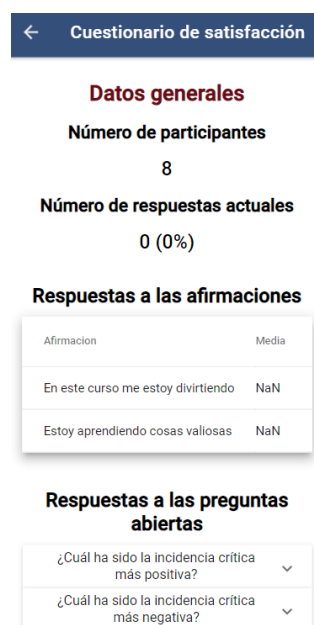


Fig. 83. Juego cuestionario de satisfacción desde la aplicación del profesor

CAPÍTULO 9. DISEÑO GENERAL DE LA APLICACIÓN

9.1. Cards juegos: Cambio de la estética para mejorar la funcionalidad

Una vez el docente inicia sesión, la aplicación le redirige a la pantalla de inicio, donde deslizando puede ir viendo los diferentes juegos activos que tiene como opción. Pero como se observa en la siguiente imagen, el diseño no está del todo conseguido, ya que la tarjeta es demasiado larga, haciendo que los botones de 'Anterior' y 'Siguiete' queden tapados. También sucedía, en ocasiones, que los botones se salieran de la pantalla dependiendo del tipo de móvil, y que la opción de moverse a otro juego resultara complicada.



Fig. 84. Pantalla de inicio previa al arreglo

Por eso, como solución se propuso fijar el tamaño de la tarjeta para evitar el problema con los botones. Pero podríamos tener un posible futuro problema con que la tarjeta si se llegara a cortar según el tipo de móvil con el que se visualizara. Por eso, para configurar a altura de la tarjeta no se puso una medida fija en píxeles, por ejemplo, 140px de altura como estilo. Si no que se configuró la altura para ocupar exactamente un cierto porcentaje de la pantalla y así sería un valor variable según el tipo de móvil que se utilizara. La estrategia era usar algo parecido a lo que se muestra en la siguiente imagen, donde se puede dejar un padding-top de un 20% y luego colocar la tarjeta con una altura del 70%, dejando un 10% como margen respecto al límite de abajo.

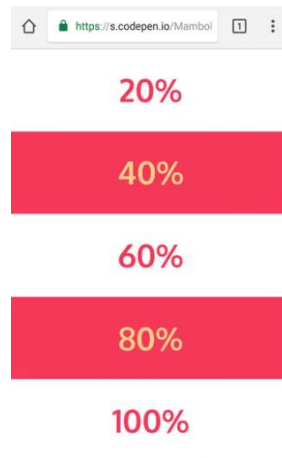


Fig. 85. Distribución de porcentajes en una pantalla móvil

Obviamente, se ha tenido en cuenta que, si la altura del móvil es muy pequeña, el contenido dentro de la tarjeta puede verse afectado. Para arreglar ese problema, se pueden utilizar estilos en cada uno de los contenedores que engloban los elementos que están dentro de la tarjeta, para crear un diseño 'Responsive'. Este tipo de diseño utiliza diseños flexibles, imágenes flexibles y consultas de medios de hojas de estilo en cascada. El objetivo del diseño 'Responsive' es crear aplicaciones que detecten el tamaño y la orientación de la pantalla del visitante y cambien el diseño en consecuencia. Por lo tanto, según el tamaño de la pantalla, el diseño cambiará de tamaño, evitando posibles problemas de superposición o recorte de elementos.

También se decidió mover los botones de 'Anterior' y 'Siguiente' arriba, para mejorar su visualización y evitar problemas de funcionalidad de los mismos.

9.2. Pantalla de inicio de juego de competición

Para el juego de competición desde un principio queríamos añadir una pantalla de inicio donde se encuentren las diferentes funciones que puede realizar el profesor. Como se muestra en la siguiente imagen, lo dividimos en dos partes:

- En la primera sección, se encuentran las funciones que puede realizar el profesor en relación con las jornadas.
- En la segunda sección, se encuentran las funciones que están relacionadas con la asignación de ganadores y la desactivación del juego.



Fig. 86. Pantalla de inicio del juego de competición

Los botones nos redirigen a cada una de las páginas de las funcionalidades que corresponden al juego de competición en el que estemos.

9.3. Botones: Estética según el tipo de función del botón

Los colores de los botones dentro de los juegos siguen un mismo patrón de colores según la funcionalidad que tengan.

1. Botón de redirección: Los botones que nos redirigen a otras páginas decidimos que fueran grises. Este color es el denominado 'light' al añadir un ion-button.

```
<button ion-button color="light">Ver jornadas</button>
```



2. Botón de desactivación: Los botones que desactivan los juegos decidimos que fueran rojos. Este color es el denominado 'danger' al añadir un ion-button.

```
<button ion-button color="danger">Desactivar juego</button>
```



3. **Botón de información:** Los botones que despliegan información adicional decidimos que fueran azules. Este color es el denominado 'primary' al añadir un ion-button.

```
<button ion-button color="primary">Más información</button>
```



9.4. Tablas: Mejora estética para correcta visualización

Las tablas es un elemento muy repetitivo en la implementación de cada uno de los juegos. Y en algunos de los casos, cuando la tabla era muy ancha a causa de la gran cantidad de columnas, se cortaba y no se veía toda la información necesaria.

Por eso necesitábamos una solución para evitar esta problemática, y vimos que con el uso de estilos podíamos solucionarlo. Este estilo era el 'overflow-x', donde esta propiedad específica si se recorta el contenido, se agrega una barra de desplazamiento o se muestra el contenido de desbordamiento de un elemento a nivel de bloque, cuando se desborda en los bordes izquierdo y derecho. Al especificar que queríamos que la propiedad fuera 'auto', en ese momento se debe proporcionar un mecanismo de desplazamiento para las celdas que se desbordan.

```
<div *ngIf = "tablaPreparada" style="width: 100%; overflow-x: auto;">
```

Fig. 87. Style añadido para solucionar la problemática con las tablas

Como vemos en la siguiente imagen existen varios valores a especificar para la propiedad de 'overflow-x', en este caso como hemos mencionado anteriormente, utilizamos el valor auto que es el que nos interesa.

Value	Description
visible	The content is not clipped, and it may be rendered outside the left and right edges. This is default
hidden	The content is clipped - and no scrolling mechanism is provided
scroll	The content is clipped and a scrolling mechanism is provided
auto	Should cause a scrolling mechanism to be provided for overflowing boxes
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Fig. 88. Style añadido para solucionar la problemática con las tablas

Por lo tanto, finalmente obtenemos un 'scrolling' horizontal que nos permite visualizar la parte de la tabla que previamente se cortaba por el tamaño del dispositivo.

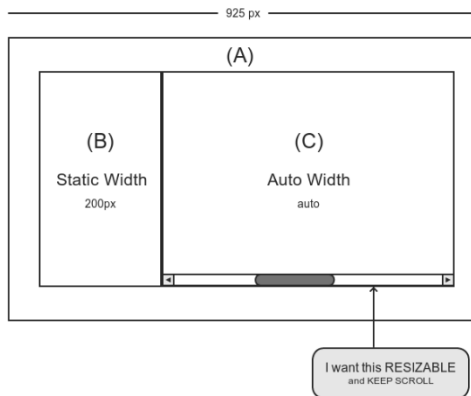


Fig. 89. Ejemplo del scrolling horizontal que obtenemos mediante overflow-x: auto

CAPÍTULO 10. RETOS MÁS DESTACADOS

En este capítulo se explicarán varios de los retos que se han ido presentando durante el proceso de implementación de los juegos o funcionalidades. En cada reto se explicará un resumen de lo que trata, la problemática a la que nos enfrentamos y las propuestas o soluciones que se han aplicado en cada caso.

Este capítulo es completamente diferente en relación con la memoria de mi compañero de proyecto, puesto que cada uno tuvo unos retos durante la implementación de los juegos.

10.1. Nodemailer: Herramienta de envío de correos electrónicos

A la hora de implementar algunos de los juegos en la aplicación del profesor, vimos que era interesante añadir la opción de enviar mails a los alumnos como funcionalidad en algunos de los juegos. Desde el momento del análisis de cada uno de los juegos, llegamos a la conclusión que era una funcionalidad que queríamos añadir en el juego de evaluación, control de trabajo y cuestionario de satisfacción.

Los primeros pasos del proyecto, fue analizar lo que ya teníamos en la aplicación del profesor, por lo tanto, ya teníamos en consideración la existencia de una funcionalidad que mandaba mail en el caso de que un profesor se haya olvidado de su contraseña. Como nuestro objetivo era hacer algo similar y queríamos mantener la coherencia del uso de herramientas en este proyecto, investigamos qué herramienta se usaba en ese momento para hacer el envío de correos.

La herramienta que estaba siendo utilizada era 'Nodemailer'. Nodemailer es un módulo que nos permite añadir la funcionalidad de envío de correos electrónicos en nuestro servidor NodeJS. En nuestro caso, ya se estaba utilizando una cuenta de gmail previamente como proveedor de envío de correos, por eso nos pusimos a investigar esa opción. Pero nodemailer puede ser usado con diversos proveedores y servicios, tal y como se indica en su documentación.

10.1.1. Gmail: Problemas de autenticación

Gmail es la forma más rápida de comenzar a enviar correos electrónicos con Nodemailer. Aunque es un servicio que espera que el usuario sea un usuario real, no un robot, por lo que ejecuta muchas heurísticas para cada intento de inicio de sesión y bloquea cualquier cosa que parezca sospechosa para defender al usuario de intentos de robo de cuenta. Y esta seguridad fue el primer de los problemas a los que nos tuvimos que enfrentar.

Anteriormente, Gmail tenía la opción de aplicaciones "menos seguras". Si esta opción estaba habilitada, era posible iniciar sesión y, por lo tanto, también mandar el mail utilizando nodemailer desde el servidor de nuestra aplicación. Pero Google decidió que, a partir del 30 de mayo del 2022, esta opción ya no está disponible por temas de seguridad. Entonces, si una aplicación no cumple con los estándares de seguridad de Google, bloquean cualquier intento de inicio de sesión a nuestra cuenta desde esa aplicación.

Ya que las aplicaciones menos seguras pueden facilitar el acceso de los hackers a la cuenta, por lo que la inhabilitación de esta opción contribuye a la protección de cuentas de gmail. En la siguiente imagen se muestra el comunicado que hizo Google respecto a la inhabilitación de la opción de aplicaciones "menos seguras".

Aplicaciones menos seguras y la cuenta de Google

Para proteger tu cuenta, a partir del 30 de mayo del 2022, Google dejará de admitir aplicaciones y dispositivos de terceros que te pidan que inicies sesión en tu cuenta de Google usando solo tu nombre de usuario y contraseña.

Importante: Este plazo no se aplica a los clientes de Google Workspace o de Google Cloud Identity. La fecha en que el cambio se implementará para estos clientes se anunciará en el blog de Workspace más adelante.

Para obtener más información, sigue leyendo.

Fig.90. Comunicado google respecto a las aplicaciones menos seguras

Para comprobar si esto afectaba a la implementación de envío de correos en la aplicación del profesor, usamos como prueba la funcionalidad de '¿Olvidaste tu contraseña?' que se encuentra en la pantalla de inicio de sesión. En la siguiente imagen se puede ver a que funcionalidad nos referimos.

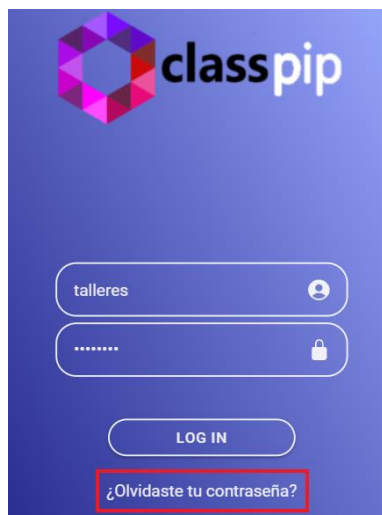


Fig.91. Pantalla de inicio de sesión de la aplicación del profesor

La estructura de la funcionalidad de envío de mails que ya estaba implementada previamente, para la funcionalidad de la restauración de la contraseña, es como la que se muestra a continuación en la siguiente imagen.

```
export class EnviarEmailService {
  public EnviarEmail() {
    const transporter = nodemailer.createTransport({
      host: 'smtp.gmail.com',
      port: 465,
      secure: true,
      auth: {
        user: 'firefighteradventure@gmail.com', // Cambialo por tu email
        pass: 'Mazinger72' // Cambialo por tu password
      },
      service: "gmail",
    });
    const mailOptions = {
      from: "firefighteradventure@gmail.com",
      to: "classpipupc@gmail.com", // Cambia esta parte por el destinatario
      subject: "tu contraseña",
      html: " Tu contraseña en classpip es esta XXXX",
    };
    // tslint:disable-next-line:only-arrow-functions
    transporter.sendMail(mailOptions, function(err, info) {
      if (err) {
        console.log(err);
      } else {
        console.log(info);
      }
    });
  }
}
```

Fig.92. Captura de código de la funcionalidad de envío de mails

Se puede ver que la autenticación se hacía introduciendo directamente la dirección de correo electrónico del emisor como su contraseña. Esto previamente era funcional, si la configuración de la cuenta de ese correo electrónico tenía habilitada la opción de aplicaciones "menos seguras". Pero como se ha explicado anteriormente, esta opción ya no existe, por lo tanto, al probar la funcionalidad de '¿Olvidaste tu contraseña?' desde la aplicación del profesor, nos mostraba el siguiente error.

```
{ Error: Invalid login: 535-5.7.8 Username and Password not accepted. Learn more at
535 5.7.8 https://support.google.com/mail/?p=BadCredentials b16-20020a5d4b9000000b0020c5253d8c2sm5092968wrt.14 - gsmt
at SMTPConnection._formatError (C:\Users\Sumaya\Desktop\TFG\classpip-server-dev\node_modules\nodemailer\lib\smtp-connection\index.js:784:19)
at SMTPConnection.actionAUTHComplete (C:\Users\Sumaya\Desktop\TFG\classpip-server-dev\node_modules\nodemailer\lib\smtp-connection\index.js:1536:34)
at SMTPConnection._responseActions.push.str (C:\Users\Sumaya\Desktop\TFG\classpip-server-dev\node_modules\nodemailer\lib\smtp-connection\index.js:540:26)
at SMTPConnection._processResponse (C:\Users\Sumaya\Desktop\TFG\classpip-server-dev\node_modules\nodemailer\lib\smtp-connection\index.js:947:20)
at SMTPConnection._onData (C:\Users\Sumaya\Desktop\TFG\classpip-server-dev\node_modules\nodemailer\lib\smtp-connection\index.js:749:14)
at TLSSocket.SMTPConnection._onSocketData.chunk (C:\Users\Sumaya\Desktop\TFG\classpip-server-dev\node_modules\nodemailer\lib\smtp-connection\index.js:189:44)
at TLSSocket.emit (events.js:182:13)
at addChunk (stream_readable.js:283:12)
at readableAddChunk (stream_readable.js:264:11)
at TLSSocket.Readable.push (stream_readable.js:219:10)
code: 'EAUTH',
response:
  '535-5.7.8 Username and Password not accepted. Learn more at\n535 5.7.8 https://support.google.com/mail/?p=BadCredentials b16-20020a5d4b9000000b0020c5253d8c2sm5092968wrt.14 - gsmt',
responseCode: 535,
command: 'AUTH PLAIN' }
```

Fig.93. Error de autenticación en el servidor al ejecutar la funcionalidad de envío de mails

El error indica que el inicio de sesión es inválido, que el nombre de usuario y la contraseña no han sido aceptados. Al buscar este error en internet, encontramos varias soluciones, donde la mayoría indican que la solución del problema era la habilitación de la opción que Google ha borrado hace muy poco tiempo (Opción de aplicaciones "menos seguras").

En ese momento, decidimos utilizar otro servicio para nodemailer y prescindir del servicio que ofrece Gmail por el momento, ya que el incremento de seguridad que justo habían aplicado nos estaba impidiendo avanzar en nuestro trabajo.

10.1.2. Gmail: Problemas de autenticación

Al comprobar que las opciones anteriores no cubren nuestras necesidades, decidimos investigar otros programas de correo, que usan conexiones seguras basadas en API para enviar correos electrónicos desde WordPress. Las opciones más recomendadas son las siguientes:

1. SendLayer
2. SMTP.com
3. Sendinblue

Nos decantamos en un principio por la opción de Sendinblue, ya que ofrece servicios confiables de entrega de correo electrónico, así como un plan gratuito bastante bueno. Este correo utiliza una API segura y, como beneficio adicional, a la hora de registrarse no hay que añadir una tarjeta de crédito para completar el proceso de configuración (a menos que se desee un plan pago).

Este servicio ofrece las siguientes características:

- **Coste:** Gratuito para hasta 300 correos por día. También ofrece una amplia variedad de planes de pago con límites de envío de correos más altos.
- **Requisitos del remitente:** Funciona con cualquier tipo de correo electrónico, pero se recomienda utilizar un correo electrónico específico del dominio para una mayor confiabilidad.
- **Conclusión:** Es un correo con una fácil configuración y tiene una sólida opción gratuita sin necesidad de introducir la tarjeta de crédito. Además, sus servicios son seguros y confiables.

Creación de la cuenta

El primer paso, es tener una cuenta en Sendinblue. En el caso de no estar registrado es posible crear una cuenta como se muestra en la figura 94. La ventaja de Sendinblue, como hemos comentado anteriormente, es que no requiere introducir una tarjeta de crédito para el registro. Una vez que ingresemos nuestra dirección de correo electrónico, es necesario revisar la bandeja de entrada para verificar nuestro correo.

SMTP Expert Deliverability

Rely on our powerful SMTP service for all your transactional emails

Immediate access - No credit card required

Fig.94. Registro en la página Sendinblue

A continuación, para asegurar de que los correos electrónicos que enviemos se entreguen correctamente, es necesario autorizar nuestro dominio en Sendinblue. Para hacerlo, iniciamos sesión, clicamos sobre el nombre de nuestra compañía en la esquina superior derecha de la pantalla, y luego seleccionamos 'Senders & IP' en el menú (Como vemos en la siguiente figura).

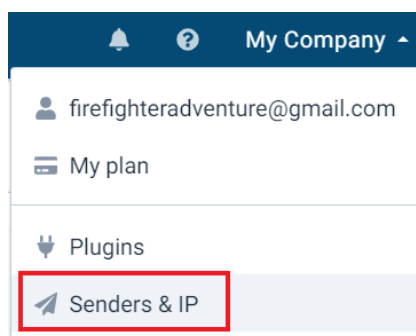


Fig.95. Funcionalidades que ofrece sendinblue

Como en este caso, Classpip es aún un proyecto en desarrollo, no tiene un dominio asignado, fue una opción con la que no pudimos seguir. Por eso se buscó otra alternativa que se explica a continuación.

10.1.3. Gmail: Uso de una contraseña de aplicación

Después de un par de días desde la inhabilitación de la opción de Google que hemos descrito en los apartados anteriores, investigando un poco más, encontramos una posible solución que podía arreglar el problema.

Era posible configurar una contraseña de aplicación en la cuenta de Google, esta contraseña funciona como una contraseña alternativa para la cuenta con la que estaremos intentando iniciar sesión desde el móvil del profesor. Solo puede ser utilizada por las aplicaciones con las que se comparte, por lo que es más seguro que compartir su contraseña principal como se hacía desde un principio.

Los pasos para llevar a cabo esta configuración son una cuenta de Gmail, es la siguiente:

1. **Activar la verificación en dos pasos** para la cuenta de Google que queramos usar. Para ello, clicamos en el icono de nuestra cuenta situado en la parte superior derecha y clicamos sobre la opción de “Gestionar tu cuenta de Google”. A continuación, nos dirigimos al apartado de seguridad y posteriormente, al apartado de “Iniciar sesión con Google” como vemos en la siguiente figura. Nos dirigimos a la opción de “Verificación en dos pasos” y seguimos los pasos que nos piden para activarlo correctamente.

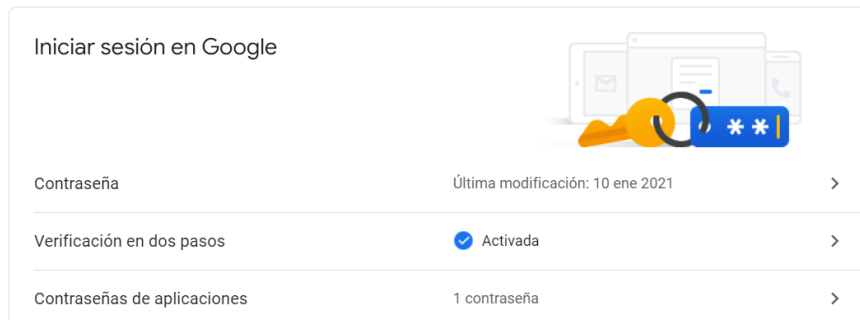


Fig.96. Apartado de configuración en la cuenta de Google para la verificación en dos pasos

2. **Crear una contraseña específica** de la aplicación para WP Mail SMTP en la configuración de seguridad de la cuenta de Google. En este caso, en el mismo apartado de la figura anterior, nos dirigimos a la opción de contraseñas de aplicaciones. A continuación, nos saldrá un formulario como el que se muestra en la siguiente imagen. Donde indica “Seleccionar aplicación” seleccionamos “Otro (Nombre personalizado)” e indicamos el nombre de la aplicación. En este caso, por ejemplo, ya hay uno creado llamado “nodemailer”. Indicamos que el nombre de la aplicación será “Classpip” y clicamos sobre el botón “Generar”.

Tus contraseñas de aplicación

Nombre	Fecha de creación	Último uso
nodemailer	15:42	15:43

Selecciona la aplicación y el dispositivo para los que quieres generar la contraseña de aplicación.

Fig.97. Formulario de creación de contraseñas de aplicaciones en la configuración del Gmail

En pocos segundos, aparecerá un popup indicando la contraseña que podemos utilizar para la aplicación que deseemos, como se ve en la siguiente imagen.

Contraseña de aplicación generada

Tu contraseña de aplicación para el dispositivo

rfga pehc kkdk gqpi

Cómo utilizarla

Accede a la sección de configuración de tu cuenta de Google en la aplicación o el dispositivo que estás intentando configurar. Sustituye tu contraseña por la contraseña de 16 caracteres que se muestra arriba. Al igual que la contraseña normal, esta contraseña de aplicación ofrece acceso completo a tu cuenta de Google. No tendrás que recordarla, así que no la escribas ni la compartas con nadie.

Email

Password

Fig.98. Popup de la clave que se acaba de generar para la aplicación

Como indican las instrucciones, hay que sustituir la contraseña principal que teníamos en la función de nuestro servidor, por esta contraseña de 16 caracteres, como se muestra en la siguiente figura. Es importante introducirla en el código antes de cerrar el popup, porque no saldrá reflejada en ningún sitio más. Si se pierde habrá que crear una nueva contraseña de nuevo, como hemos explicado anteriormente.

```

export class EnviarEmailService {
  public EnviarEmail() {
    const transporter = nodemailer.createTransport({
      host: 'smtp.gmail.com',
      port: 465,
      secure: true,
      auth: {
        user: 'firefighteradventure@gmail.com', // Cambialo por tu email
        pass: 'dsxrpgujauvgxfjd' // Cambialo por tu password
      },
      service: "gmail",
    });
    const mailOptions = {
      from: "firefighteradventure@gmail.com",
      to: "classpipupc@gmail.com", // Cambia esta parte por el destinatario
      subject: "tu contraseña",
      html: " Tu contraseña en classpip es esta XXXX",
    };
    // tslint:disable-next-line:only-arrow-functions
    transporter.sendMail(mailOptions, function(err, info) {
      if (err) {
        console.log(err);
      } else {
        console.log(info);
      }
    });
  }
}

```

Fig.99. Captura de código de la funcionalidad con la contraseña de aplicación

Es importante tener en cuenta, que cada vez que se cambie la contraseña de la cuenta principal de Google, también se desactivarán las contraseñas de la aplicación.

Para comprobar si esta solución funciona correctamente, se hace como siempre, ejecutando la función de '¿Olvidaste tu contraseña?' para verificar el envío del correo electrónico. Y como se puede comprobar en las siguientes capturas, el mail ha sido enviado correctamente, por lo tanto, se puede concluir que esta solución es válida y funciona.

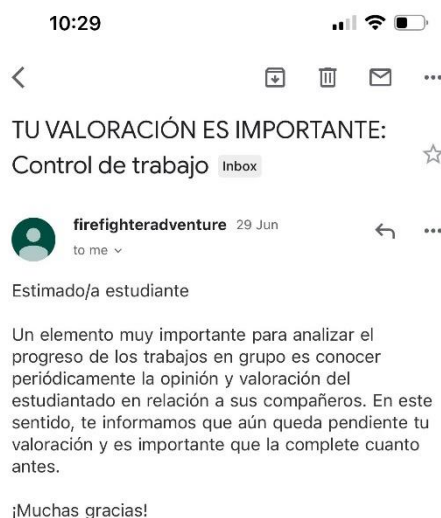


Fig. 100. Correo de recordatorio en el juego de control de trabajo

10.2. Problemática con las versiones

A continuación, se explicará una de las problemáticas que influyeron en la gestión u organización del tiempo del proyecto, ya que a causa de este problema algunos de los tiempos establecidos en los sprints fueron retrasados.

En un momento dado, durante el proyecto, tuve que eliminar los repositorios del Dashboard y el del móvil del profesor. Al volver a clonarlo todo y seguir exactamente los pasos que se explican en el repositorio de Classpip sobre la instalación, al hacer 'npm i' en ambos repositorios recién clonados daban un error como el que se muestra a continuación.

```

npm WARN deprecated uuid@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://github.com/uuidjs/uuid/blob/master/CHANGELOG.md for details.
npm WARN deprecated mkdirp@0.5.1: Legacy versions of mkdirp are no longer supported. Please update to mkdirp 1.x. (Note that the API surface has changed to use Promises in 1.x.)
npm ERR! code 1
npm ERR! path C:\Users\Sumaya\Desktop\TFG\classpip-dashboard-dev\node_modules\node-sass
npm ERR! command failed
npm ERR! command C:\WINDOWS\system32\cmd.exe /d /s /c node scripts/build.js
npm ERR! Building: C:\Program Files\nodejs\node.exe c:\Users\Sumaya\Desktop\TFG\classpip-dashboard-dev\node_modules\node-gyp\bin\node-gyp.js rebuild --verbose --libsass_ext= --libsass_
_ldflags= --libsass_library=
npm ERR! gyp info it worked if it ends with ok
npm ERR! gyp verb cli [
npm ERR! gyp verb cli   'C:\\Program Files\\nodejs\\node.exe',
npm ERR! gyp verb cli   'C:\\Users\\Sumaya\\Desktop\\TFG\\classpip-dashboard-dev\\node_modules\\node-gyp\\bin\\node-gyp.js',
npm ERR! gyp verb cli   'rebuild',
npm ERR! gyp verb cli   '--verbose',
npm ERR! gyp verb cli   '--libsass_ext=',
npm ERR! gyp verb cli   '--libsass_cflags=',
npm ERR! gyp verb cli   '--libsass_ldflags=',
npm ERR! gyp verb cli   '--libsass_library='
npm ERR! gyp verb cli ]
npm ERR! gyp info using node-gyp@3.8.0
npm ERR! gyp info using node@16.13.2 | win32 | x64
npm ERR! gyp verb command rebuild []
npm ERR! gyp verb command clean []
npm ERR! gyp verb clean removing "build" directory
npm ERR! gyp verb command configure []
npm ERR! gyp verb check python checking for python executable "/path/to/executable/python" in the PATH
npm ERR! gyp verb `which` failed Error: not found: /path/to/executable/python
npm ERR! gyp verb `which` failed   at getNotFoundError (C:\Users\Sumaya\Desktop\TFG\classpip-dashboard-dev\node_modules\which\which.js:13:12)
npm ERR! gyp verb `which` failed   at F (C:\Users\Sumaya\Desktop\TFG\classpip-dashboard-dev\node_modules\which\which.js:68:19)

```

Fig.101. Error al realizar 'npm i' en el Dashboard

Al observar que era un error que nunca había visto, decidí informarme sobre ello en internet. Donde después de una extensa búsqueda, todas las respuestas eran muy parecidas, la mayoría proponían eliminar la caché, eliminar la carpeta node_modules, eliminar el archivo package.lock.json, y volver a ejecutar el comando 'npm i'.

Después de intentar esta opción varias veces y otras muchas más en ambos proyectos, el error seguía saliendo y siendo el mismo. En ese momento, decidí preguntar la opinión de un profesor e intentar resolverlo, ya que podía ser que era un error común y que ellos podían ver cuál era realmente la solución.

A continuación, explicaré en detalle cada una de las soluciones que se probaron hasta llegar a la solución correcta y la que actualmente funciona.

10.2.1. Primera propuesta: Realizar los pasos de instalación de nuevo

La primera propuesta surgió durante la primera reunión con el profesor. El primer paso que me sugirió llevar a cabo fue ejecutar el comando "npm i node-sass", ya que el error indicaba algo relacionado con esa dependencia. Pero el comando no se terminaba de ejecutar correctamente, saltaba un error muy parecido que el del principio. Por lo tanto, el siguiente paso que me recomendó, fue borrar los repositorios que daban problemas y seguir de nuevo los pasos que se indicaban en la guía de classpip, ejecutando de nuevo también el comando "npm i". Una vez aplicada esta segunda posible solución, el resultado fue negativo también. Saltaba exactamente el mismo error, y en ese momento decidimos preguntar la opinión de otro de los profesores que podía ayudarme con mi problema.

10.2.2. Segunda propuesta: Desinstalar Node completamente

La segunda propuesta surgió durante la segunda reunión para solucionar esta problemática. Al empezar la reunión y explicarle al profesor todos los pasos que había seguido previamente para solucionar el problema sin éxito, me comentó que por el tipo de error que estaba saliendo, podía ser una problemática de versiones. Por eso, el primer paso que me recomendó fue desinstalar completamente Nodejs de mi ordenador para eliminar cualquier tipo de conflicto de versiones.

Para la desinstalación completa seguí los siguientes pasos:

1. Limpiar la caché con el comando "npm cache clean --force". Es importante ejecutar este comando, ya que siempre que instalamos un paquete con el comando "npm i", npm almacena el caché dentro del sistema de archivos del usuario.
2. Verificar la caché con el comando "npm cache verify".
3. Abrir el panel de control, elegir la opción de 'Desinstalar un programa' en la sección 'Programas', buscar Nodejs en la lista y desinstalarlo.
4. Reiniciar el ordenador.
5. Buscar las diferentes carpetas que se muestran en la siguiente imagen, y si están, eliminarlas. En el caso de no encontrarlas, es totalmente normal, puesto que pueden existir o no según la versión instalada o dependiendo de otros factores también.

- C: \ Archivos de programa (x86) \ Nodejs
- C: \ Archivos de programa \ Nodejs
- C: \ Users \ {User} \ AppData \ Roaming \ npm o abra ejecutar y escriba *appdata* y haga clic en Aceptar y abra roaming allí encontrará *npm* .
- C: \ Users \ {User} \ AppData \ Roaming \ npm-cache o abra ejecutar y escriba *appdata* y haga clic en Aceptar y abra roaming allí encontrará *npm-cache* .
- C: \ Users \ {User} \ .Npmrc
- C: \ Users \ {User} \ AppData \ Local \ Temp \ npm-*

Fig.102. Lista de carpetas a borrar

6. Verificar las variables de la ruta del entorno y mirar que las referencias a 'npm' o 'Nodejs' no existan.
7. Para verificar que node está 100% desinstalado, ejecutamos el comando "where node" en el terminal. En el caso de que no se haya desinstalado correctamente, el comando generara la ubicación donde se encuentra el programa en Windows. Por lo tanto, habría que ir a esa ubicación y desinstalarlo. En el caso de que todo haya ido bien en la desinstalación, al ejecutar el comando se mostrara un mensaje de 'No encontrado', que indica que node está totalmente desinstalado.
8. Como último paso, reiniciamos el ordenador para evitar cualquier problema.

Una vez, teníamos Nodejs completamente desinstalado, buscamos la versión que se recomienda en la guía de classpip y la instalamos. En este caso, se recomendaba instalar la versión 10.13.0.

Al realizar la instalación descrita previamente, clonar los repositorios de nuevo y descargar las dependencias con "npm i", parecía que no daba ningún error y todo funcionaba correctamente respecto a classpip. Pero ahora el problema que me surgió fue que el proyecto en el que estaba trabajando en las prácticas de empresa, donde también utilizaba Nodejs como herramienta, dejó de funcionar. En ese momento, me di cuenta de que la solución que acababa de aplicar era válida solo para poder trabajar correctamente en el proyecto de classpip, pero que en otros proyectos donde también utilizaba Nodejs, la versión que acababa de instalar no era válida.

10.2.3. Tercera propuesta: NVM

La tercera propuesta y solución final, surgió a partir del momento en que se conoció la problemática exacta y se hizo la búsqueda concreta. Como los diferentes proyectos en los que estaba trabajando, necesitaban versiones diferentes, busqué alguna herramienta que me permitiera trabajar o cambiar la versión de node según en el proyecto en el que estuviera trabajando. Después de buscar y probar algunas opciones, encontré un software llamado NVM.

El software NVM (Node Version Manager) permite instalar y gestionar distintas versiones de NodeJS en un ordenador, lo que facilita el trabajo con proyectos que requieren versiones distintas. Permite intercambiar la versión de NodeJS que tenemos activa en un momento dado, entre todas las que se hayan instalado en el sistema. Además, nvm soluciona algunos problemas típicos que suelen ocurrir cuando se trabaja con npm y al instalar dependencias globales se requieren permisos de sudo. Por lo tanto, con nvm no los necesitarás y el software funcionará correctamente.

La instalación de este software fue una tarea bastante intuitiva que describiré a continuación.

La opción que elegí fue instalar este software usando nvm-windows, siguiendo unos pasos muy rápidos y fáciles de seguir:

1. Accedemos al link del repositorio que se adjunta a continuación:
<https://github.com/coreybutler/nvm-windows/releases>.
2. Descargamos el archivo “nvm-setup.zip” de la última release disponible.
3. Extraemos y ejecutamos el archivo “nvm-setup.exe”.
4. A continuación, seguimos los pasos de la instalación y elegimos la configuración que se desee.
5. Y listo, ya estaría instalado.

Si ya tenemos una versión de node instalada, si ejecutamos el comando “nvm list”, se listarán todas las versiones que tengamos instaladas en ese momento. Para cambiar de una a otra, simplemente ejecutamos el comando “nvm use <versión que queramos utilizar>”. Y ya se cambia y está lista para ser usada.

Gracias a esta solución fue posible trabajar con versiones de node diferentes, tanto en el proyecto de classpip, como en el proyecto que realizaba en las prácticas de empresa. Desde ese momento, no volvieron a suceder errores relacionados con versiones o con la instalación errónea de dependencias.

CAPÍTULO 11. CONCLUSIONES

En este capítulo se comentarán las conclusiones finales tanto técnicas, como personales, como la valoración de los objetivos del proyecto.

11.1. Conclusiones técnicas

Al comienzo del proyecto, era consciente de que me iba a enfrentar a varios retos técnicos durante el transcurso de cualquiera de las implementaciones que se hicieran. Una vez finalizado el proyecto, efectivamente, ha habido varios retos técnicos que se han tenido que superar. Tal y como se describe en el anterior capítulo, uno de ellos fue la problemática con las versiones o la problemática de la mecánica de envío de correos electrónico. Pero gracias a haberlos superado, he aprendido técnicas, herramientas o softwares nuevos que no sabía antes, y esto ha hecho crecer mi conocimiento técnico en gran parte. Este conocimiento, seguramente, me sea de utilidad en futuros proyectos en los que trabaje en el mundo laboral.

También, uno de los retos en la primera fase del proyecto fue entender todo el ecosistema de Classpip que ya estaba implementado por estudiantes de promociones previas. Fue un reto que con tiempo y con análisis, pudimos acabar de superarlo, aunque también parte de la ayuda nos la proporcionaron nuestros tutores. En nuestro caso, era necesario familiarizarse con el código y la lógica que se había seguido en los juegos, puesto que íbamos a migrar gran parte de algunas funcionalidades ya existentes. Por lo tanto, entender el código que habían hecho otras personas, adaptarlo y mejorarlo a nuestra manera, fue también una de las cosas que nos hizo aprender mucho.

Como tal, los lenguajes de programación no fueron un gran problema, puesto que ya eran familiares de asignaturas previas que se habían hecho durante algunos cursos en la universidad. Así que, se podría decir que hemos mejorado nuestras habilidades con estos lenguajes, gracias a haber trabajado con ellos mucho más durante estos meses.

11.2 Valoración de los objetivos y el plan de trabajo

Los objetivos iniciales se han cumplido, como se ha visto reflejado en la memoria. Aunque durante el transcurso del proyecto se han ido añadiendo cosas que veíamos interesantes para el proyecto y descartando otras que ya no encajaban con la forma que estaba tomando el proyecto.

La metodología de organización que hemos utilizado (Scrum), ha sido de gran ayuda para organizarnos con el tiempo y las tareas por hacer. Sobre todo, porque ambos estábamos realizando las prácticas de empresa durante todo este periodo. Por lo tanto, es una metodología que recomiendo a futuros estudiantes que quieran participar en el proyecto de Classpip.

A nivel personal, estoy contenta con el trabajo que he realizado, con todo lo que he aprendido durante estos meses y con haber cumplido con los objetivos iniciales que nos planteamos.

En relación con el trabajo en equipo, ha sido un placer trabajar con Suhail Chaib, mi compañero, puesto que ha sido una persona trabajadora, empática y que me ha apoyado en todo momento. Considero que hemos trabajado muy bien en equipo, nos hemos facilitado el trabajo el uno al otro y no hemos tenido ningún problema en ningún momento. Por eso, concluyo con que ha sido un placer para mí trabajar con él y realizar mi trabajo de fin de grado.

11.3. Valoración personal

Termino mi participación en este proyecto con buenas sensaciones, ha sido una experiencia muy buena y con la que he disfrutado. Me he sentido muy cómoda trabajando en un proyecto como este, donde la mayor parte del trabajo es algo que realmente disfruto hacer, que es programar. Gracias a haberlo experimentado, me siento con más ganas de dedicarme a algo parecido en mi futuro laboral, aunque soy consciente de lo mucho que queda por aprender, este es el campo tecnológico que me entusiasma.

Tengo la sensación de haber mejorado como desarrolladora, gracias a los retos que se han presentado y he solucionado en su momento. Empecé a programar con asignaturas que se realizan en la universidad, aprendiendo poco a poco de que va el mundo de la programación. Y termino sabiendo lo grande que es este campo y lo mucho que queda por aprender, para poder crecer como desarrolladora en un futuro. Pero Classpip ha sido el empujón para poder seguir aprendiendo y practicando en un futuro.

Quiero agradecer la ayuda que nos ha proporcionado nuestro tutor Miguel a lo largo del transcurso del proyecto. Haber trabajado en una parte de lo que es Classpip me enorgullece, y espero que siga creciendo y tenga un futuro magnífico.

REFERENCIAS

- [1] Miguel Valero García, Roc Meseguer Pallarés - Repositorios de Classpip en Github:
<https://github.com/classpip>
- [2] Business news daily – Información sobre la metodología SCRUM:
<https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>
- [3] Trello: <https://trello.com/es>
- [4] Stackoverflow - Información relevante para solucionar el problema del envío de correos electrónico con NodeMailer:
<https://stackoverflow.com/questions/45478293/username-and-password-not-accepted-when-using-nodemailer>
- [5] Información sobre la problemática entre Gmail y Nodemailer en el año 2022:
<https://wpmailsmtp.com/gmail-less-secure-apps/>
- [6] Información sobre desinstalar Nodejs de Windows completamente:
<https://es.acervolima.com/como-eliminar-por-completo-node-js-de-windows/>
- [7] Información sobre la instalación de NVM:
<https://midu.dev/como-instalar-node-en-mac-y-windows/>