



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

TeamMaker: Plataforma para constituir equipos de trabajo
multidisciplinares

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Agulleiro Fuentes, Adrián

Tutor/a: Letelier Torres, Patricio Orlando

CURSO ACADÉMICO: 2022/2023

Resumen

El trabajo en equipos multidisciplinares es reconocido como el factor clave para el éxito en la mayoría de los proyectos, especialmente en aquellos asociados a nuevos desafíos e innovación. En el ámbito académico existe una clara tendencia en el interés por formar profesionales preparados y con experiencia en trabajo en equipos multidisciplinares. La formación de los estudiantes se organiza por especialidad y separada en diferentes titulaciones y escuelas, con lo cual, salvo contactos y colaboraciones puntuales, no hay mecanismos ni herramientas que faciliten la comunicación necesaria para la formación de equipos multidisciplinares para trabajar conjuntamente en proyectos. El objetivo de "TeamMaker" es ofrecer un espacio en el cual personas (profesores o estudiantes) interesadas en desarrollar un proyecto con un equipo multidisciplinar puedan publicar su iniciativa, y como contraparte otras personas que puedan estar interesadas puedan contactar para formar parte del equipo. Este TFG nace de la motivación asociada al desarrollo de TFG/TFM o proyectos de emprendimiento en el cual participen estudiantes de diferentes escuelas, lo cual es una necesidad actual de nuestra universidad, pero también aplicable a otras universidades, y en general podría ser útil en cualquier comunidad o institución de gran envergadura donde se quiera poner en contacto a personas para trabajar en un proyecto. Este TFG se desarrolla en el marco de una práctica de empresa en la UPV y asociada a la cátedra HP-UPV.

Palabras clave: equipos multidisciplinares, desarrollo de app, trabajo en equipo, estudiantes

Abstract

Working in cross-functional teams is recognized as a key factor for success in most projects, especially those associated with new challenges and innovation. In the academic field there is a clear trend in the interest in training professionals prepared and experienced in working in multidisciplinary teams. The training of students is organized by specialty and separated in different degrees and schools, so that, except for contacts and occasional collaborations, there are no mechanisms or tools to facilitate the necessary communication for the formation of multidisciplinary teams to work together on projects. The objective of "TeamMaker" is to offer a space in which people (teachers or students) interested in developing a project with a multidisciplinary team can publish their initiative, and as a counterpart other people who may be interested can contact to be part of the team. This TFG is born from the motivation associated with the development of TFG/TFM or entrepreneurship projects in which students from different schools participate, which is a current need of our university, but also applicable to other universities, and in general could be useful in any community or large institution where you want to put people in contact to work on a project. This TFG is developed within the framework of an internship at the UPV and associated with the HP-UPV chair.

Keywords: multidisciplinary teams, app development, teamwork, students

Índice de contenidos

1.	Introducción	7
1.1.	Motivación	7
1.2.	Objetivos	8
1.3.	Impacto esperado	8
1.4.	Estructura de la memoria	9
2.	Equipos multidisciplinares.....	10
3.	Estado del arte	13
3.1.	Organizadores de eventos	13
3.2.	Plataformas de comunicación.....	15
3.3.	Publicador de proyectos.....	15
3.4.	Conclusiones	16
4.	Tecnologías utilizadas	18
4.1.	Capa de presentación.....	18
4.1.1.	React Native.....	18
4.1.2.	Virtual DOM	19
4.1.3.	Paquetes de React Native	20
4.1.4.	Expo Dev.....	21
4.2.	Capa de lógica de negocio	22
4.2.1.	Node.JS.....	22
4.2.2.	Express.js.....	22
4.2.3.	Paquetes de Node.JS	23
4.3.	Capa de persistencia	24
5.	Desarrollo de la solución	25
5.1.	Planteamiento general.....	25
5.2.	Especificación de requisitos.....	26
5.2.1.	Requisitos funcionales.....	26
5.2.2.	Requisitos no funcionales.....	39
5.3.	Diseño	40
5.3.1.	Arquitectura del sistema	40
5.3.2.	Estructura detallada	44
5.4.	Programación.....	51

5.4.1.	Seguridad en el sistema	51
5.4.2.	Patrones de diseño y refactoring	55
5.4.3.	Análisis energético o de eficiencia algorítmica.....	59
5.5.	Pruebas	61
5.6.	Metodología y cronología	65
5.6.1.	Línea de trabajo	65
5.6.2.	Estructura y flujo de trabajo.....	65
5.6.3.	Planificación y seguimiento del trabajo	66
5.6.4.	Plan de trabajo.....	67
5.6.5.	Cronología del trabajo	69
6.	Conclusiones y trabajo futuro	72
	Referencias	74
	Apéndice	76
A.	Objetivos de Desarrollo Sostenible.....	76
B.	Manual de usuario	78



Índice de figuras

Figura 1. Habilidades con mayor interés en las organizaciones.....	10
Figura 2. Interfaz de la aplicación MeetUp [10].....	14
Figura 3. Estructura en memoria de los componentes en React Native	19
Figura 4. Diagrama de casos de uso de la aplicación	26
Figura 5. Prototipo de la cuenta de usuario	28
Figura 6. Prototipo de la vista principal de la interfaz	32
Figura 7. Diagrama de estado de solicitudes a perfiles	34
Figura 8. Prototipo de gestión de solicitudes	35
Figura 9. Prototipo de listado de postulaciones	36
Figura 10. Diagrama de estados de un proyecto	38
Figura 11. Arquitectura software del sistema	43
Figura 12. Tecnologías empleadas en cada una de las capas.....	44
Figura 13. Flujo de navegación de la aplicación	46
Figura 14. Diagrama de componentes del sistema.....	48
Figura 15. Diagrama de clases de la aplicación	49
Figura 16. Diagrama Entidad Relación de la aplicación	50
Figura 17. Cifrado de valores de entrada del usuario	52
Figura 18. Servicio de verificación de tokens	54
Figura 19. Verificador de campos de entrada.....	55
Figura 20. Fragmentos de la estructura del patrón Strategy	56
Figura 21. Fragmento de implementación del Patrón Proxy	57
Figura 22. Servicio de transporte de correos electrónicos	58
Figura 23. Configuración de una lista plana en React Native	59
Figura 24. Técnica de memoización en React Native.....	60
Figura 25. Estructura de código de una prueba unitaria	62
Figura 26. Estructura base de una prueba de integración	63
Figura 27. Resultado de las pruebas en la lógica de negocio.....	63
Figura 28. Resultados de la prueba de rendimiento de la IGU.....	64
Figura 29. Resultado de la prueba Flipper Performance Monitor	64
Figura 30. Workflow de unidad de trabajo	66
Figura 31. Planificación y seguimiento del trabajo	66
Figura 32. Modificaciones en un Sprint	67
Figura 33. Cronología del proyecto	69
Figura 34. Vistas iniciales de TeamMaker	78
Figura 35. Herramienta de búsqueda y ficha personal de TeamMaker	79
Figura 36. Navegación y gestión de mis proyectos.....	80
Figura 37. Manejo de las solicitudes a perfiles.....	81

1. Introducción

1.1. Motivación

Debido a una tendencia global en la especialización de conocimientos en el entorno empresarial y docente [1], actualmente hay una fuerte demanda de líneas de trabajo con equipos multidisciplinares [2]. Esto es, la unión de varios miembros con conocimiento de diferentes disciplinas para llevar a cabo un fin común. Se puede observar que hay un marco común en la búsqueda de profesionales para proyectos de media o gran escala en el ámbito corporativo y del sector tecnológico [3]. Cada vez son más las iniciativas que buscan formar un equipo con personas de diferentes disciplinas para poder desarrollar un producto o servicio. En el entorno académico también existe un proceso creciente debido a la demanda del mercado actual. Sin embargo, no existen herramientas que faciliten o automaticen la gestión y la formación de estas líneas de trabajo en grupos multidisciplinares.

La motivación principal es generar una herramienta útil, fácil de usar y muy accesible para promover y mejorar la realización de proyectos con equipos multidisciplinares dentro del ámbito académico. Para lograr esto se opta por la producción de una aplicación para smartphones válida para los sistemas operativos iOS y Android.

Contar con una herramienta que en unos pocos pasos permita compartir una propuesta de trabajo con miles de estudiantes y profesores de diferentes disciplinas, o por el contrario solicitar la unión a una propuesta de proyecto, es un gran incentivo. Los estudiantes se benefician mejorando sus aptitudes y competencias colaborando y trabajando conjuntamente con otras disciplinas y los docentes tienen una herramienta potente para captar al alumnado.

Actualmente es muy relevante la importancia de contar con mecanismos que mejoren la experiencia para encontrar perfiles adecuados para formar equipos multidisciplinares en el ámbito académico [4]. Tener al alcance una utilidad que permite visualizar las iniciativas de otros estudiantes y del profesorado para aprender más tecnologías y metodologías. En la contraparte, tener la capacidad de poder compartir una iniciativa con miles de personas cuando sea necesario. La evidencia se encuentra en la necesidad y proliferación de proyectos de emprendimiento en la universidad [5] y su transferencia a otras organizaciones que tengan unas necesidades similares.

La iniciativa surge de una práctica en empresa asociada a la cátedra HP-UPV que entre otras cosas busca producir actividades y fomentar el desarrollo de proyectos que se realicen en el ámbito de TFG/TFM, con una posible orientación hacia el emprendimiento y los equipos multidisciplinares. De ahí que TeamMaker sea una necesidad para este ámbito y también una justificación para otras organizaciones que busquen algo similar.

1.2. Objetivos

El principal objetivo es desarrollar un sistema software permita gestionar el proceso de publicación de proyectos con equipos multidisciplinares. Este sistema será accesible por el usuario final a través de una aplicación para móviles con el nombre “TeamMaker”. Para ello se proponen los siguientes requisitos u objetivos secundarios:

- Publicar un proyecto a través de un formulario con sus respectivos campos válidos y una lista de perfiles o roles que serán definidos por la administración de la aplicación y los usuarios. Este proyecto debe ser verificado por algún miembro de la administración antes de ser público en el listado.
- Gestionar las solicitudes a los perfiles del proyecto que se ha publicado, incentivando a través de un sistema de notificaciones la respuesta ágil. El creador del proyecto puede ver los usuarios que se han inscrito a los diferentes perfiles de su proyecto así como descargar un documento proporcionado por el solicitante a modo de Curriculum Vitae o carta de presentación. Tras la revisión de la candidatura puede aceptar o rechazar la posición y ponerse en contacto con dicho usuario.

Otro objetivo es la inscripción a perfiles en los proyectos multidisciplinares que hay publicados en la lista de la aplicación. Para llevar a cabo todo el flujo de actividades necesarias para el proceso de inscripción, se definen los siguientes requisitos u objetivos secundarios:

- Sistema de búsqueda preciso que permite filtrar los proyectos que el usuario desea encontrar por texto incluido en el título o descripción, por perfil o disciplina y por validez para TFG/TFM.
- Utilidad para inscribirse a los perfiles de los proyectos y tener un listado del estado de las inscripciones realizadas. Cualquier cambio en el estado de la inscripción será notificado al usuario.
- El usuario en todo momento puede cambiar su información de contacto, así como su documento de presentación y su imagen de modo que sea flexible durante el tiempo para los diferentes proyectos a los que se inscribe.

1.3. Impacto esperado

El producto software va a suponer una mejora en la calidad de formación de equipos multidisciplinares. Tener una herramienta robusta que permita conectar el entorno docente para aprender las competencias necesarias. Ofrece ventajas tanto para aquel usuario que desea llevar a cabo una idea como para aquel que quiere aprender cosas nuevas uniéndose a algún proyecto. Esta iniciativa espera generar un impacto positivo en la meta 4.4 del objetivo 4 de los ODS de las Naciones Unidas que pretende aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento.

1.4. Estructura de la memoria

La memoria se divide en los siguientes seis capítulos:

- El capítulo 1 es introductorio y en él se presenta el contexto en el que se desarrolla el proyecto, se describe la problemática que se pretende abordar y se plantea el objetivo general del proyecto.
- El segundo capítulo consta de una descripción detallada y contexto de la importancia del trabajo en equipos multidisciplinares y del impacto que tiene en la sociedad actual.
- A continuación, en el capítulo 3 se aborda el análisis del estado del arte donde se realiza una revisión de las aplicaciones y herramientas existentes en el mercado que tienen una finalidad similar a la de "TeamMaker". Se analizan las características, ventajas y limitaciones de cada una de estas aplicaciones.
- La descripción de la tecnología utilizada para el desarrollo de la aplicación, la explicación del proceso de selección de las tecnologías y el detalle de las herramientas utilizadas se especifican en el cuarto capítulo.
- En el capítulo 5 se explica el proceso de desarrollo de la aplicación, se detallan las características y funcionalidades de la aplicación, se muestra su diseño y se describe el proceso de implementación y pruebas.
- Por último, en el sexto capítulo se realiza una valoración crítica del proyecto, se destacan los logros alcanzados y las limitaciones encontradas durante el desarrollo. Finalmente, se proponen líneas de trabajo futuro para la mejora y el perfeccionamiento de la aplicación.

2. Equipos multidisciplinares

Los equipos multidisciplinares, también conocidos como equipos *cross-functional*, son grupos de personas con diferentes habilidades y conocimientos que trabajan juntas hacia un objetivo común. Estos equipos están compuestos por individuos de diferentes áreas o disciplinas, tales como marketing, diseño, tecnología, finanzas, recursos humanos, entre otros, que trabajan juntos para abordar un problema o desarrollar un producto.

Los equipos multidisciplinares se han convertido en una herramienta fundamental tanto en el ámbito académico como en el corporativo y su impacto ha sido significativo. Según el informe "The Future of Jobs Report 2020" [6] del World Economic Forum, se espera que la capacidad de trabajar en equipos multidisciplinares sea una de las habilidades clave que los trabajadores deberán desarrollar en los próximos años. Esto se debe a que estos equipos han demostrado ser altamente efectivos en la solución de problemas complejos y la generación de nuevas ideas y soluciones innovadoras.

La Figura 1, que corresponde al gráfico del capítulo 2.3 del mencionado informe [6] muestra como la importancia relativa de la habilidad de trabajar en equipos *cross-functional* ha aumentado en alrededor del 50% de las organizaciones en el año 2020. Fueron menos del 10% las organizaciones que disminuyeron el interés en tener empleados con esta habilidad. Un 40% mantuvieron el interés.

Un ejemplo práctico, de 1000 organizaciones, alrededor de 900 decidieron que la mejor opción era aumentar o mantener el interés en los equipos multidisciplinares.

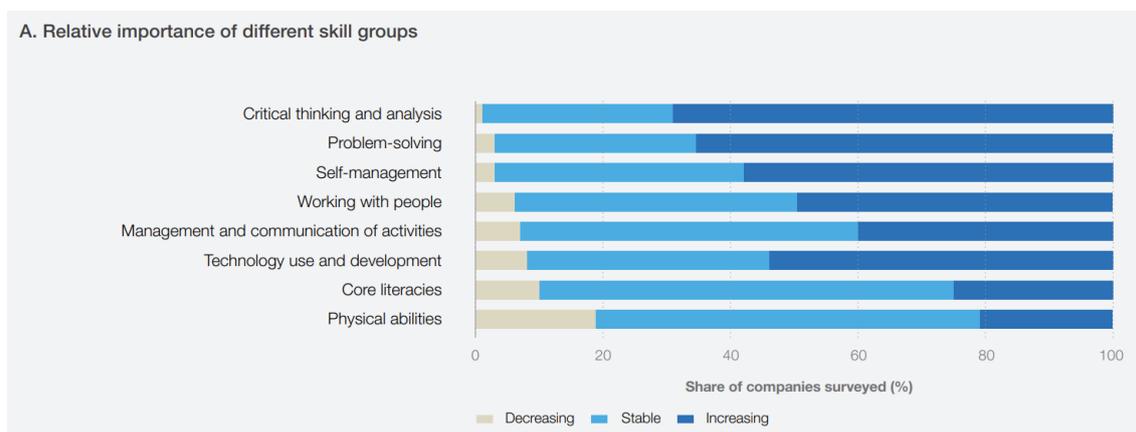


Figura 1. Habilidades con mayor interés en las organizaciones

En el ámbito académico, los equipos multidisciplinares han demostrado ser una herramienta muy útil para abordar cuestiones complejas y fomentar la innovación. Un estudio publicado en la revista "Annual Review of Organizational Psychology and Organizational Behavior" [7] destaca que los equipos multidisciplinares pueden ser especialmente efectivos en entornos de investigación, donde la generación de ideas

innovadoras y la solución de problemas complejos son cruciales para avanzar en el conocimiento científico. Además, los equipos multidisciplinares pueden ayudar a fomentar la colaboración entre diferentes disciplinas y mejorar la calidad de la investigación al reunir diferentes perspectivas.

En el ámbito corporativo, los equipos multidisciplinares también han demostrado ser muy efectivos. Un informe de McKinsey & Company [8] señala que los equipos que están compuestos por individuos con diferentes habilidades, conocimientos y experiencias pueden mejorar la creatividad y la toma de decisiones, lo que puede llevar a una mayor innovación y productividad. Además, los equipos multidisciplinares también pueden ayudar a mejorar la colaboración y la comunicación dentro de una organización, ya que los miembros de estos equipos a menudo provienen de diferentes áreas funcionales y pueden tener diferentes perspectivas y prioridades.

La importancia de los equipos multidisciplinares radica en que permiten abordar problemas complejos desde diferentes perspectivas, lo que aumenta la creatividad y la calidad de las soluciones [9]. Además, los equipos *cross-functional* pueden llevar a cabo tareas de manera más rápida y eficiente, ya que cada miembro aporta su propia experiencia y conocimientos específicos. Esto reduce el riesgo de errores y aumenta la efectividad del equipo. Un ejemplo de la importancia de los equipos multidisciplinares se puede observar en el desarrollo de tecnologías innovadoras. Los productos tecnológicos modernos son cada vez más complejos y requieren la colaboración de diferentes disciplinas para su desarrollo. Por ejemplo, un equipo de desarrollo de software puede estar compuesto por ingenieros, diseñadores, expertos en experiencia de usuario, entre otros. Esta colaboración permite que se genere un producto final que es de alta calidad y que se adapta a las necesidades de los usuarios.

En cuanto a los beneficios para las personas, trabajar en equipos multidisciplinares permite el desarrollo de habilidades sociales y emocionales, como la empatía, la comunicación efectiva, el liderazgo y la resolución de conflictos. Además, trabajar en un entorno colaborativo puede aumentar la motivación y el sentido de pertenencia, lo que a su vez puede aumentar la satisfacción laboral.

En conclusión, los equipos multidisciplinares son esenciales tanto en el ámbito académico como en el corporativo. Permiten abordar problemas complejos desde diferentes perspectivas, mejoran la eficiencia y la productividad, y son esenciales para la innovación y el éxito empresarial. Las personas también pueden beneficiarse de trabajar en equipos *cross-functional*, ya que les permite desarrollar habilidades sociales y emocionales, aumentar la motivación y la satisfacción laboral.

En la actualidad, a pesar de los avances tecnológicos y la creciente popularidad de las redes sociales, no existen mecanismos o herramientas digitales que faciliten la conexión de personas de diferentes perfiles en el ámbito académico para formar equipos y desarrollar proyectos de emprendimiento y/o que sean válidos para TFG/TFM. Esta falta de herramientas digitales se convierte en un problema para los estudiantes y profesionales que buscan colaboradores con habilidades complementarias para llevar a cabo proyectos de investigación, emprendimiento y trabajos finales de grado o máster. Sin un medio eficiente para encontrar a estas personas, el proceso de búsqueda y selección puede ser lento y tedioso, lo que puede limitar el potencial creativo y el éxito del proyecto.

Aunque existen algunas plataformas en línea para conectar a estudiantes y profesionales en general, estas no se adaptan específicamente a las necesidades del ámbito académico.

En conclusión, la falta de herramientas digitales para la conexión de personas de diferentes perfiles en el ámbito académico se convierte en una barrera para la colaboración creativa y el desarrollo de proyectos exitosos. Es importante que se fomente el desarrollo de plataformas específicas para este fin, que sean gratuitas y que garanticen la calidad de los perfiles de los usuarios para así mejorar el éxito y eficiencia de los proyectos.

3. Estado del arte

Debido a la creciente demanda y aparición emergente de proyectos de pronta creación, formado por un grupo de emprendedores de diferentes disciplinas que unen una idea o motivo común y sobre una base tecnológica, hoy en día se cuenta con diversas plataformas donde este perfil de usuario puede compartir y adquirir información. Además de tener utilidades para transformar esa idea innovadora en una realidad.

Estas actividades comprenden un rango que empieza desde ofrecer facilidades para que una persona pueda obtener una base y referencias para formular un proyecto, crear espacios donde las personas puedan intercambiar conocimiento hasta ofrecer actividades y eventos para reunir personas con un perfil u objetivo común y de forma guiada realizar un proceso o desafío para crear un nexo entre los participantes.

Estos no son los únicos elementos que se encuentran presentes en el ámbito tecnológico para fomentar la creación de proyectos. Muchas organizaciones como bancos o universidad también cuentan con departamentos que abren convocatorias para fomentar el pensamiento y la innovación.

En la red están presentes plataformas que permiten publicar ideas de diversos proyectos y otras que permiten abrir hilos de comunicación y espacios para interactuar con otros usuarios para facilitar la tarea de encontrar un grupo que permita llevar a cabo una idea con éxito. No obstante, son muy pocas o inexistentes las que tienen varias herramientas o aspectos unificados en un mismo lugar. No hay una aplicación para móviles a nivel académico que permita en muy pocos pasos y al alcance de la mano poder encontrar esos proyectos gracias a herramientas de búsqueda combinables y un sistema automatizado de notificaciones que esté pensado desde cero para agilizar el proceso de búsqueda y selección de talento y pasión para formar el equipo deseado para esa idea que quiere convertirse en algo más.

A continuación, se van a mostrar las herramientas que ofrecen funcionalidades similares a las del planteamiento de TeamMaker. Dividiendo en categorías y explicando que aspecto es competidor de la aplicación.

3.1. Organizadores de eventos

Existen organizaciones que abren un apartado para crear eventos donde estudiantes o profesionales formen un equipo multidisciplinar. Dentro de esta categoría existen dos vertientes. La primera y la más clásica es determinar como regla que la plataforma que se encarga de llevar a cabo la realización del evento es sostenida por la misma organización promotora de dicho evento. La segunda es que la organización ejerce el rol de ente externo al evento y únicamente se encarga de ser el nexo.



Santander X

Santander X¹ es una iniciativa por parte del Banco Santander² con el objetivo de crear una red de emprendimiento universitario. La idea es que las universidades y emprendedores puedan conectar, colaborar aportando sus ideas, conocimientos y ganas para alzar el espíritu del emprendimiento.

Ofrece un espacio digital donde los estudiantes y emprendedores tienen acceso a recursos de la comunidad. En este ecosistema los miembros pueden ponerse en contacto con otros miembros y comentar sus ideas. Además, pueden encontrar mentores y solicitar asesoramiento al departamento de emprendimiento de su universidad.

Comparar directamente esta plataforma con TeamMaker es algo complejo, ya que ambas tienen objetivos y alcances que divergen. Sin embargo, se pueden señalar algunas diferencias y ventajas de TeamMaker en relación con la plataforma de Santander X.

MeetUp

MeetUp³ es una plataforma en línea que permite a los usuarios crear y unirse a grupos de interés en su área local. La plataforma es conocida por su amplia gama de grupos, que abarcan desde actividades al aire libre hasta eventos de negocios y tecnología. Los usuarios pueden unirse a grupos ya existentes o crear sus propios grupos, lo que les permite conectarse con personas que comparten sus intereses.

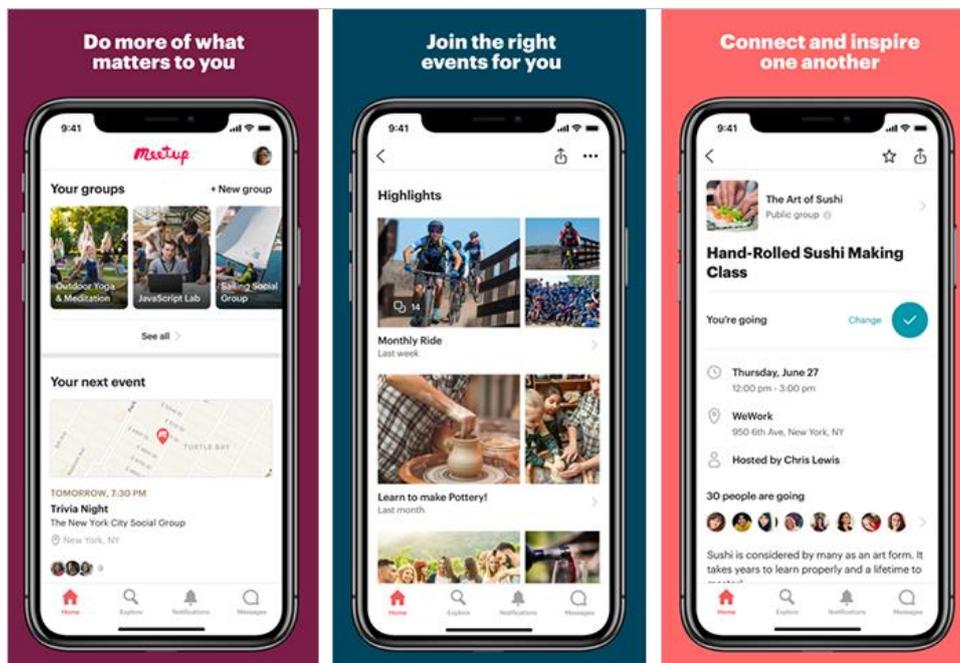


Figura 2. Interfaz de la aplicación MeetUp [10]

¹ Sitio web oficial de Santander X: <https://www.santanderx.com/es>

² Sitio web oficial de Banco Santander: <https://www.bancosantander.es>

³ Sitio web oficial de MeetUp: <https://www.meetup.com>

Apreciando las ilustraciones de la Figura 2, se puede observar cómo los grupos pueden organizar eventos en línea o presenciales, y los usuarios pueden asistir a eventos y recibir actualizaciones sobre los próximos eventos en su área. Además, la plataforma ofrece herramientas de comunicación interna para los miembros del grupo, lo que les permite interactuar y planificar actividades en línea.

3.2. Plataformas de comunicación

Existen diversas plataformas de comunicación en línea actualmente. Este apartado podría ser de gran extensión pues son multitud las plataformas de comunicación que existen actualmente y que posiblemente se formarán en el futuro.

Una plataforma muy utilizada en el sector más cercano al que TeamMaker va enfocado es LinkedIn ⁴, pues ofrece un lugar donde personas con ganas de emprender pueden encontrar a otras personas con el mismo objetivo y formar un equipo. LinkedIn es una red social orientada a la búsqueda de empleo, establecimiento de contactos profesionales y creación de redes de colaboración. Dado que TeamMaker también se enfoca en la colaboración multidisciplinar, es posible que haya alguna similitud en la forma en que ambas plataformas conectan a personas con intereses comunes y habilidades complementarias. Sin embargo, hay algunas diferencias importantes entre ambas plataformas que hacen que TeamMaker sea una mejor opción para la formación de equipos multidisciplinarios en proyectos.

3.3. Publicador de proyectos

En esta categoría se encuentran los competidores directos, pues TeamMaker se puede categorizar como un publicador de proyectos. Por ende, las plataformas aquí mencionadas serán las que tienen más similitudes y están más alineadas. La diferencia residirá en el enfoque y el contexto principalmente.

TeamerUP

TeamerUP ⁵ es una plataforma digital creada por la Universidad Politécnica de Valencia que tiene como objetivo conectar a emprendedores que tienen ideas de negocio o startups con personas que poseen habilidades y talentos específicos que puedan ayudar a desarrollar y hacer crecer esos proyectos. Los emprendedores pueden subir sus proyectos a la plataforma y buscar equipo para formar su "*dream team*", mientras que los interesados en unirse a una startup pueden buscar oportunidades para invertir su talento en proyectos que les resulten interesantes. Además, la plataforma ofrece herramientas para contactar con otros emprendedores y buscar nuevas ideas de negocio. En resumen,

⁴ Sitio web oficial de LinkedIn: <https://www.linkedin.com>

⁵ Sitio web oficial de TeamerUP: <https://teamerup.upv.es>

TeamerUP es una red de emprendedores que buscan formar equipos y llevar sus ideas de negocio a la realidad.

Las dos tienen un enfoque similar al permitir a los usuarios publicar proyectos y encontrar personas con las habilidades y conocimientos necesarios para llevarlos a cabo. La principal ventaja es que TeamMaker cuenta con una aplicación móvil donde los usuarios pueden acceder a la plataforma desde cualquier lugar y en cualquier momento, lo que brinda una mayor flexibilidad y movilidad. Además, como aplicación móvil, es más fácil de usar e intuitiva que una página web, lo que puede atraer a un público más amplio. Otro elemento clave en la diferenciación es que puede ser fácilmente migrada y personalizada para ser utilizada en cualquier organización lo que la convierte en una herramienta muy útil para la gestión de proyectos y la formación de equipos en diferentes contextos. En unas pocas horas se puede adaptar el sistema de TeamMaker para desplegarlo en cualquier otra organización que necesite este servicio para su comunidad.

Sendas plataformas ofrecen mecanismos de búsqueda para los proyectos que son bastante parecidos (búsqueda por texto, filtro por perfiles, etc). Sin embargo, una mejora respecto a TeamerUP es la posibilidad de poder combinar estas utilidades de búsqueda al mismo tiempo.

3.4. Conclusiones

Comparando TeamMaker con las plataformas descritas más arriba se puede concluir ciertos parámetros y argumentos para llevar a cabo el proyecto.

TeamMaker está diseñado específicamente para ayudar a formar equipos multidisciplinarios para trabajar juntos en proyectos, mientras que Santander X es una plataforma más extensa que ofrece una amplia gama de servicios para emprendedores y startups. TeamMaker se ha diseñado específicamente para facilitar la formación de equipos multidisciplinarios para proyectos de investigación y emprendimiento en un entorno académico, lo que lo hace especialmente útil para estudiantes universitarios y docentes. Santander X, por otro lado, se enfoca en ofrecer herramientas para emprendedores y startups para acelerar su crecimiento y desarrollo. Además, TeamMaker puede extenderse a cualquier gran organización fuera del ámbito académico que quiera promover este tipo de iniciativas.

Otro aspecto fundamental es el coste de incorporación de la plataforma. Santander X requiere de una colaboración con una universidad y de una serie de pasos y acuerdos intermedios para poder llegar a brindar la oportunidad y el acceso a esos miembros universitarios. Por el contrario, TeamMaker ofrece una aplicación para móviles lista para su uso e instalación y que tiene una estructura altamente liviana para que cualquier organización pueda migrar la plataforma a su infraestructura. Incluso varias organizaciones pueden colaborar entre ellas y compartir la infraestructura y los recursos para sus miembros.

MeetUp está orientado a planificar actividades entre grupos de personas que compartan algún objetivo. A pesar de que se mueve en el mismo contexto que TeamMaker, dista mucho en el enfoque, ya que TeamMaker está orientado a proyectos y no a actividades

puntuales (eventos con una fecha determinada). TeamMaker a diferencia de MeetUp se centra en agrupar personas con diferentes conocimientos y disciplinas, pero con un objetivo común en un proyecto de más larga duración en comparación con MeetUp. Como bien define en su sitio web, MeetUp busca en un área local, esto está alineado con TeamMaker. Sin embargo, TeamMaker ofrece más facilidades y herramientas para encontrar un proyecto en específico en esa misma cota. Por tanto, MeetUp está puramente centrado en las personas mientras que TeamMaker en los proyectos y el talento de esas personas.

En cuanto a la plataforma de comunicación, LinkedIn se centra en la búsqueda de empleo y en la promoción de las habilidades individuales de los usuarios. Mientras que, TeamMaker se centra en la formación de equipos de trabajo multidisciplinarios para abordar proyectos específicos. LinkedIn no ofrece una manera fácil de encontrar personas con habilidades específicas para colaborar en un proyecto, y aunque existen grupos de LinkedIn, no son específicos para la formación de equipos multidisciplinarios para trabajar en proyectos específicos. LinkedIn está diseñado para conectar personas con personas, no para conectar personas con proyectos específicos. TeamMaker está diseñado específicamente para ayudar a los usuarios a encontrar proyectos y a unirse a equipos multidisciplinarios, lo que hace que sea mucho más fácil para los usuarios encontrar proyectos que se ajusten a sus habilidades y experiencias.



4. Tecnologías utilizadas

Para desarrollar la aplicación TeamMaker, se plantea una arquitectura basada en tres capas [11] (explicado en el siguiente capítulo). Esta arquitectura divide la aplicación en tres capas lógicas diferentes, donde cada capa se encarga de diferentes funciones y procesos. A continuación, se muestran las tecnologías empleadas en cada capa.

4.1. Capa de presentación

La capa de presentación es la capa más cercana al usuario, es la que presenta la información de manera visual y recoge las acciones del usuario. Para TeamMaker, la capa de presentación es una aplicación móvil desarrollada en React Native, que se encarga de mostrar la información de los proyectos publicados, permitir la búsqueda, mostrar la información al usuario, etc.

4.1.1. React Native

React Native ⁶ es una tecnología de desarrollo de aplicaciones móviles multiplataforma que permite a los desarrolladores crear aplicaciones móviles para iOS y Android utilizando JavaScript y React. Esta tecnología fue desarrollada por Facebook y es una extensión de React, una biblioteca de JavaScript utilizada para construir interfaces de usuario en la web.

React Native utiliza una arquitectura basada en componentes, lo que significa que la aplicación está compuesta por una serie de componentes reutilizables. Estos componentes se pueden combinar para crear interfaces de usuario complejas y ricas en funcionalidades. Además, React Native proporciona una amplia variedad de componentes preconstruidos que se pueden utilizar para acelerar el proceso de desarrollo. Permite a los desarrolladores crear aplicaciones móviles para varias plataformas al mismo tiempo, esto significa que un solo conjunto de código puede ser utilizado para crear aplicaciones tanto para iOS como para Android, lo que reduce significativamente el tiempo y los costes de desarrollo.

React Native tiene capacidad para crear aplicaciones móviles altamente optimizadas y con un rendimiento similar al de las aplicaciones nativas. React Native utiliza un sistema de renderizado en tiempo real que permite a la aplicación actualizar la interfaz de usuario en tiempo real sin tener que esperar a que se complete una carga de página. Esto se traduce en una experiencia de usuario más rápida y fluida. También es compatible con una amplia variedad de bibliotecas y complementos de terceros, lo que facilita la integración de características adicionales en la aplicación.

⁶ Sitio web oficial de React Native: <https://reactnative.dev>

Teniendo en cuenta las ventajas de React Native mencionadas anteriormente, es una excelente opción para desarrollar la capa de presentación de la aplicación TeamMaker. Con React Native, se puede crear una interfaz de usuario personalizada y altamente optimizada que se adapte perfectamente a las necesidades de la aplicación y de los usuarios. Permite ahorrar tiempo y costes de desarrollo al permitir crear una aplicación móvil para múltiples plataformas al mismo tiempo. Esto es especialmente importante para una aplicación como TeamMaker, que debe estar disponible tanto para dispositivos iOS como para dispositivos Android.

4.1.2. Virtual DOM

El DOM (Document Object Model) es una representación estructurada en forma de árbol de todos los elementos HTML y XML en una página web. En React Native, el DOM no existe tal como lo conocemos en el desarrollo web tradicional, ya que React Native utiliza componentes nativos en lugar de HTML para crear interfaces de usuario. En lugar del DOM, React Native utiliza una jerarquía de componentes nativos para renderizar la interfaz de usuario. Cada componente nativo en React Native tiene su propio modelo de objetos, que es utilizado por React Native para actualizar la interfaz de usuario de manera eficiente y sin necesidad de recargar toda la aplicación. Este concepto ha sido asignado con el nombre de Virtual DOM.

El virtual DOM es una de las características más importantes de React Native y proporciona una gran ventaja para la aplicación móvil de TeamMaker. El virtual DOM es una representación ligera de la estructura de la interfaz de usuario en memoria, que es independiente de la plataforma subyacente. Esto significa que no importa si el usuario está utilizando un iPhone o un teléfono Android, la interfaz de usuario se renderizará de manera consistente y sin errores. La representación del virtual DOM se actualiza de forma asincrónica y solo se aplica los cambios necesarios a la interfaz de usuario real en la pantalla, lo que mejora significativamente el rendimiento de la aplicación. Al no tener que volver a renderizar toda la interfaz cada vez que hay un cambio, se ahorra tiempo y se mejora la experiencia. Además, permite trabajar con una abstracción más alta de la interfaz, en lugar de tener que preocuparse por detalles de bajo nivel de la plataforma subyacente. Esto simplifica significativamente el proceso de desarrollo, ya que no es necesario escribir código específico de la plataforma para cada tipo de dispositivo móvil.

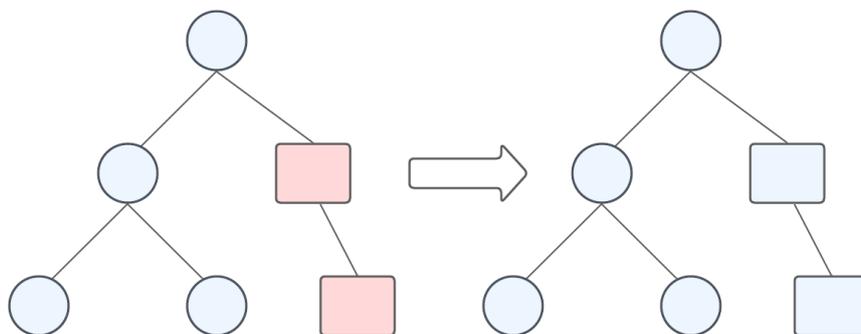


Figura 3. Estructura en memoria de los componentes en React Native

Como se puede ver en la Figura 3, cuando se produce un cambio en alguno de los componentes que React Native almacena en memoria, únicamente se modifica y vuelve a renderizar ese componente y los que están contenidos en él (marcado en color rojo y con forma rectangular). El resto de la estructura se mantiene intacta. Esto mejora significativamente el rendimiento de la aplicación y se traduce en una mayor fluidez y experiencia de usuario en el manejo.

4.1.3. Paquetes de React Native

Cuando se trabaja en el desarrollo de una aplicación en React Native, se hace necesario la utilización de diferentes paquetes o librerías para implementar diversas funcionalidades y características en la aplicación. Para empezar, React Native viene con algunos paquetes preinstalados porque son esenciales para el funcionamiento de la aplicación. Pero además hay que hacer uso de los paquetes de terceros disponibles en el ecosistema de React Native, que se pueden instalar a través de administradores de paquetes. Muchos de estos paquetes están respaldados por grandes organizaciones y suponen un gran aumento de productividad en el desarrollo de la aplicación. A continuación, se exponen los paquetes que usa TeamMaker con una concisa explicación. Estos paquetes se encuentran listados en el archivo de configuración del proyecto y son los siguientes:

- **@gorhom/bottom-sheet:** Es un paquete que permite agregar una hoja de fondo para mostrar contenido adicional.
- **@react-native-community/datetimepicker:** Sirve para implementar un componente gráfico de selector de fecha y hora.
- **@react-navigation/drawer:** Proporciona una navegación de menú deslizante para aplicaciones React Native.
- **@react-navigation/native:** Ofrece herramientas de navegación para aplicaciones React Native, como Stack, Drawer y Bottom Tabs.
- **@react-navigation/stack:** Produce una navegación de pila para aplicaciones React Native.
- **expo:** Es un paquete que proporciona una plataforma todo en uno para desarrollar, compilar y publicar aplicaciones React Native.
- **expo-checkbox:** Inserta de forma fácil un componente de casilla de verificación en formularios.
- **expo-font:** Permite cargar y utilizar fuentes personalizadas.
- **expo-image-picker:** Proporciona una forma fácil de seleccionar imágenes de la galería o la cámara en un dispositivo.
- **expo-secure-store:** Es un paquete que proporciona una forma segura de almacenar y recuperar datos confidenciales tanto en Android como iOS.
- **expo-splash-screen:** Muestra una pantalla de presentación personalizada.
- **expo-status-bar:** Herramientas para personalizar la barra de estado en una aplicación React Native.
- **moment:** Facilita la gestión de fechas y tiempos, así como los formatos.
- **react:** Es la biblioteca principal de React que permite a los desarrolladores construir interfaces de usuario en aplicaciones web y móviles.

- **react-hook-form:** Es un paquete que proporciona una forma fácil de manejar formularios en aplicaciones React Native.
- **react-native:** Es el marco de trabajo principal para la construcción de aplicaciones móviles en React Native.
- **react-native-gesture-handler:** Es un paquete que proporciona una forma fácil de manejar gestos táctiles en aplicaciones React Native.
- **react-native-reanimated:** Proporciona herramientas para animar componentes en aplicaciones React Native.
- **react-native-safe-area-context:** Es un paquete que proporciona herramientas para manejar el espacio seguro en la pantalla en aplicaciones de forma nativa.
- **react-native-screens:** Es un paquete que proporciona herramientas para manejar la navegación en aplicaciones React Native.
- **expo-document-picker:** Es un paquete que proporciona una forma fácil de seleccionar documentos en una aplicación tanto en Android como en iOS.

4.1.4. Expo Dev

Para el despliegue de la aplicación móvil se utiliza la herramienta Expo ⁷, una plataforma gratuita y de código abierto que permite a los desarrolladores crear, probar y desplegar aplicaciones móviles para iOS y Android utilizando React Native. Expo proporciona herramientas y servicios para simplificar el proceso de desarrollo y despliegue de aplicaciones móviles, permitiendo a los desarrolladores enfocarse en la creación de características y funcionalidades para sus aplicaciones. Una de las principales ventajas de utilizar Expo es la facilidad de uso y la rápida configuración inicial. Se puede crear una nueva aplicación base utilizando sólo un comando y comenzar a desarrollar la aplicación inmediatamente, sin tener que preocuparse por la configuración y el manejo de dependencias. Tiene soporte para la función de "Live Reload", que permite a los desarrolladores ver los cambios realizados en el código de su aplicación en tiempo real en sus dispositivos móviles. Esto acelera significativamente el proceso de desarrollo y reduce el tiempo que se necesita para probar y depurar la aplicación.

Estas funcionalidades se pueden encontrar también en una aplicación genuina de React con las configuraciones adecuadas. El motivo principal de utilizar Expo es su servicio de compilación en la nube que permite compilar y construir la aplicación para iOS y Android sin la necesidad de configurar y mantener un sistema de compilación local. Por ejemplo, para compilar la aplicación para iOS se necesita un dispositivo de Apple con una versión del sistema operativo Mac OS ⁸ adecuada. Sin embargo, con este servicio se puede construir la aplicación en la nube sin necesidad de disponer de estos dispositivos para realizar el proceso de compilación. Esto reduce el tiempo necesario para desarrollar y desplegar una aplicación móvil.

Otro elemento muy importante es que Expo tiene la capacidad de proporcionar actualizaciones de la aplicación de manera rápida y sencilla. Permite actualizar la

⁷ Sitio web oficial de Expo: <https://expo.dev>

⁸ Sitio web oficial de Mac OS: <https://www.apple.com/macOS>



aplicación sin necesidad de volver a enviar el *bundle*⁹ a las tiendas de aplicaciones, lo que puede ser un proceso largo y tedioso. En su lugar, Expo utiliza su propia plataforma de actualización Over-The-Air (OTA) [12] para proporcionar actualizaciones a los usuarios de manera rápida y eficiente. Las actualizaciones se pueden enviar directamente al dispositivo del usuario, lo que permite corregir errores, añadir nuevas funcionalidades y mejorar el rendimiento de la aplicación en tiempo real. Las actualizaciones OTA en Expo se basan en un sistema de versiones y etiquetas. Cuando un desarrollador envía una actualización, se crea una nueva versión y se etiqueta con un número de versión único. Los usuarios que tienen la aplicación instalada en sus dispositivos reciben una notificación de actualización y pueden elegir actualizar la aplicación cuando les resulte conveniente.

4.2. Capa de lógica de negocio

4.2.1. Node.JS

Node.JS¹⁰ es una tecnología de servidor que permite ejecutar código JavaScript en el lado del servidor. Fue creado en 2009 y desde entonces ha ido ganando popularidad debido a su facilidad de uso y escalabilidad. Se basa en el motor de JavaScript V8¹¹, que también se utiliza en el navegador Chrome¹².

Una de las principales ventajas es su capacidad para manejar una gran cantidad de solicitudes concurrentes sin bloquear el hilo de ejecución. Esto se logra a través del modelo de entrada/salida (E/S) no bloqueante, lo que significa que las solicitudes no esperan a que se completen las operaciones antes de pasar a la siguiente solicitud. En su lugar, las solicitudes se procesan en una cola de eventos (término *event queue* en inglés) y se entregan a un hilo de ejecución de forma asíncrona. Esto permite que el servidor maneje múltiples solicitudes en paralelo, lo que reduce el tiempo de espera del cliente y mejora la capacidad de respuesta del servidor.

Node.JS también cuenta con una amplia gama de módulos y paquetes de código abierto disponibles a través de su gestor de paquetes llamado NPM (Node Package Manager)¹³. Estos paquetes pueden ser fácilmente integrados para agregar funcionalidades adicionales. En el caso de las herramientas implicadas en esta capa el uso de este gestor de paquetes será crucial para tener el servicio completo de TeamMaker.

4.2.2. Express.js

⁹ Archivo que contiene todos los elementos necesarios para instalar y ejecutar una aplicación

¹⁰ Sitio web oficial de Node.JS: <https://nodejs.org>

¹¹ Sitio web oficial de JavaScript V8 Engine: <https://v8.dev>

¹² Chrome Browser by Google: <https://www.google.com/chrome>

¹³ Sitio web oficial de NPM: <https://www.npmjs.com>

Express.js ¹⁴ es un marco de trabajo web minimalista que permite crear APIs (Interfaz de Programación de Aplicaciones) de manera sencilla y eficiente. Express.js se integra con Node.JS a través de su sistema de *middleware*, que son funciones que se encuentran en medio del procesamiento de las peticiones que maneja Node.JS y que tienen acceso a los objetos de solicitud y respuesta HTTP. Permite definir rutas, configurar otros *middlewares* para la aplicación y definir parámetros para las rutas. Además, Express.js es altamente personalizable y permite integrar fácilmente otras herramientas y librerías para la creación de aplicaciones web complejas.

4.2.3. Paquetes de Node.JS

Como se ha mencionado anteriormente, es crucial hacer uso de los paquetes y librerías de código abierto disponibles a través del gestor NPM para generar de forma eficiente la capa de lógica de negocio. La índole y el objetivo que desempeñan cada uno de los paquetes es muy diferente en algunos casos, pero juntos crean un bloque en el sistema que permite gestionar las solicitudes y las respuestas HTTP de forma eficiente. También se aumenta el desacople entre la aplicación y los paquetes o dependencias, por lo que la mantenibilidad de la aplicación aumenta en ciertos aspectos.

A continuación, se puede apreciar el listado de paquetes que se encuentran dentro de la capa de lógica de negocio y una breve explicación de la implicación que tienen respecto al sistema como conjunto.

- **bcryptjs:** Un paquete para cifrar contraseñas de manera segura y cifrar datos confidenciales.
- **cors:** Middleware para Express.js que permite el acceso cruzado de recursos para sitios web y aplicaciones web de diferentes dominios.
- **dotenv:** Un paquete que carga variables de entorno desde un archivo `.env` a `process.env` para su uso en la aplicación.
- **express-validator:** Un middleware para Express.js que proporciona una forma fácil de validar datos de entrada y validar las solicitudes de API.
- **helmet:** Middleware para Express.js que ayuda a proteger las aplicaciones web configurando encabezados HTTP para mejorar la seguridad.
- **jsonwebtoken:** Implementa la creación y verificación de *tokens* JSON Web ¹⁵ para la autenticación de usuario.
- **multer:** Un middleware para Express.js que permite la carga de archivos.
- **mysql:** Proporciona una API para conectarse a una base de datos MySQL ¹⁶ y ejecutar consultas SQL.
- **nodemailer:** Enviar correos electrónicos desde la aplicación.
- **randomatic:** Un paquete para generar cadenas aleatorias de caracteres y números usando algoritmos criptográficos no predecibles.

¹⁴ Sitio web oficial de Express.js: <https://expressjs.com>

¹⁵ Sitio web oficial de JSON Web Token: <https://jwt.io>

¹⁶ Sitio web oficial de MySQL: <https://www.mysql.com>



- **sharp:** Proporciona una API para manipular imágenes, como cambiar su tamaño, recortarlas y convertirlas a otros formatos de imagen.

Además, para mejorar el rendimiento en el desarrollo de este servicio y poder transformar código fuente para que sea interpretado con JavaScript nativo se cuenta con unos paquetes que no se instalan en la aplicación final (la versión de producción). Estos paquetes se llaman “Dev dependencies” y como el nombre indica son dependencias que solo se tienen en cuenta en el ámbito de desarrollo. Cuando se despliega este servicio en la infraestructura de producción no se instalan.

- **@babel/core:** Es el núcleo de Babel ¹⁷, el cual permite transformar el código JavaScript a través de plugins.
- **@babel/cli:** Conjunto de herramientas para utilizar Babel en la línea de comandos.
- **@babel/node:** Herramienta para utilizar Babel en Node.JS para transpilar código sobre la marcha mientras se ejecuta.
- **@babel/preset-env:** Conjunto de plugins de Babel que permiten transformar el código JavaScript a una versión más antigua que sea compatible con navegadores o entornos específicos.
- **chai:** Es una librería de aserciones (*assertions*) para JavaScript que permite hacer afirmaciones sobre el comportamiento de un programa.
- **jasmine:** Herramienta de pruebas para JavaScript que proporciona una sintaxis limpia y fácil de usar para escribir pruebas unitarias y de integración
- **mocha:** Ofrece una gran flexibilidad en la organización y ejecución de pruebas, y soporta múltiples estilos de aserciones y reportes de resultados.
- **morgan:** *Middleware* de registro de solicitudes HTTP que registra información sobre las solicitudes entrantes, como la dirección IP del cliente, la fecha y hora de la solicitud y la ruta de la solicitud.
- **nodemon:** Ayuda a reiniciar automáticamente la aplicación Node.JS cada vez que se realiza un cambio en el código fuente.
- **supertest:** Es una librería de pruebas de integración para Node.JS que permite realizar peticiones HTTP a una aplicación y comprobar la respuesta obtenida mediante una sintaxis fácil de usar y extensible.

4.3. Capa de persistencia

La capa de persistencia en TeamMaker está implementada utilizando MySQL, un sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Permite almacenar, organizar y recuperar grandes cantidades de datos de forma eficiente y escalable. En la capa de persistencia, se utilizan consultas SQL para interactuar con la base de datos y manipular los datos según las necesidades de la aplicación.

¹⁷ Sitio web oficial de BabelJS: <https://babeljs.io>

5. Desarrollo de la solución

5.1. Planteamiento general

El propósito principal del sistema es generar una herramienta que sea muy amigable para el usuario y de fácil acceso para incentivar y mejorar la calidad en el proceso de formación de un equipo multidisciplinar en el ámbito académico. Aportar una infraestructura para la acogida de publicaciones de proyectos académicos que necesitan miembros de otras disciplinas para continuar con su trayectoria. Para abordar dicha problemática se puede hacer uso de la tecnología (software) para ofrecer soluciones que permitan mejorar el acercamiento de los usuarios interesados en formar equipos multidisciplinarios.

Actualmente hay varias plataformas predominantes en cuanto al uso diario en el ámbito académico y de las organizaciones. Las más empleadas o reconocidas son las plataformas web, móviles y de escritorio. Por tanto, es conveniente analizar cuál de las plataformas puede ser la más acertada para realizar el producto software.

Empezando por la web, es una plataforma altamente extendida y admitida por la mayoría de los dispositivos. Sin embargo, no ofrece el mismo nivel de interacción que una aplicación para dispositivos móviles puede aportar [13]. Una aplicación móvil puede proporcionar notificaciones *push* (de sistema) para recordar y notificar al usuario de algún cambio de estado de forma mucho más rápida y visual. Es un medio que el usuario tiende a consultar con mayor frecuencia [14]. También permite un acceso más directo a otros componentes del dispositivo como puede ser la cámara o los botones físicos del *smartphone* y permiten una interacción más fluida y conocida por el usuario.

En cuanto a una aplicación de escritorio hay un factor de descarte determinante que es el aspecto de la movilidad. Con una aplicación de escritorio sería necesaria un dispositivo más voluminoso como puede ser una computadora y eso reduciría la conexión entre el usuario y el sistema. Sobre todo, en el ámbito de la interacción y las notificaciones.

Por tanto, la decisión ha sido desarrollar una aplicación para dispositivos móviles como componente que interactúa directamente con el usuario.

5.2. Especificación de requisitos

5.2.1. Requisitos funcionales

Los requisitos funcionales del sistema se pueden definir en un conjunto de cuatro requisitos cuyos casos de uso asociados son los siguientes:

Característica	Casos de uso
Cuenta de usuario	CU1. Iniciar sesión, CU2. Registrarse, CU3. Modificar cuenta.
Publicación de proyecto	CU6. Crear proyecto, CU9. Ver proyectos.
Manejo de proyecto	CU4. Inscribirse a un perfil, CU7. Gestionar proyectos, CU8. Buscar un proyecto, CU11. Ver inscripciones.
Comunicación	CU5. Enviar notificación, CU10. Aceptar o rechazar inscripción.
Administración	CU12. Validar proyectos, CU13. Crear perfil.

Con el fin de obtener una mayor abstracción de la solución propuesta y entender cada uno de los aspectos, visualizar si existe alguna precondition o procedimiento determinado, así como asignar el ámbito al que pertenece cada caso de uso, a continuación, se muestra en formato de diagrama (Figura 4) los diferentes casos de uso con los actores del sistema y posteriormente se describe cada caso de uso en el formato estándar correspondiente (especificación semi formal).

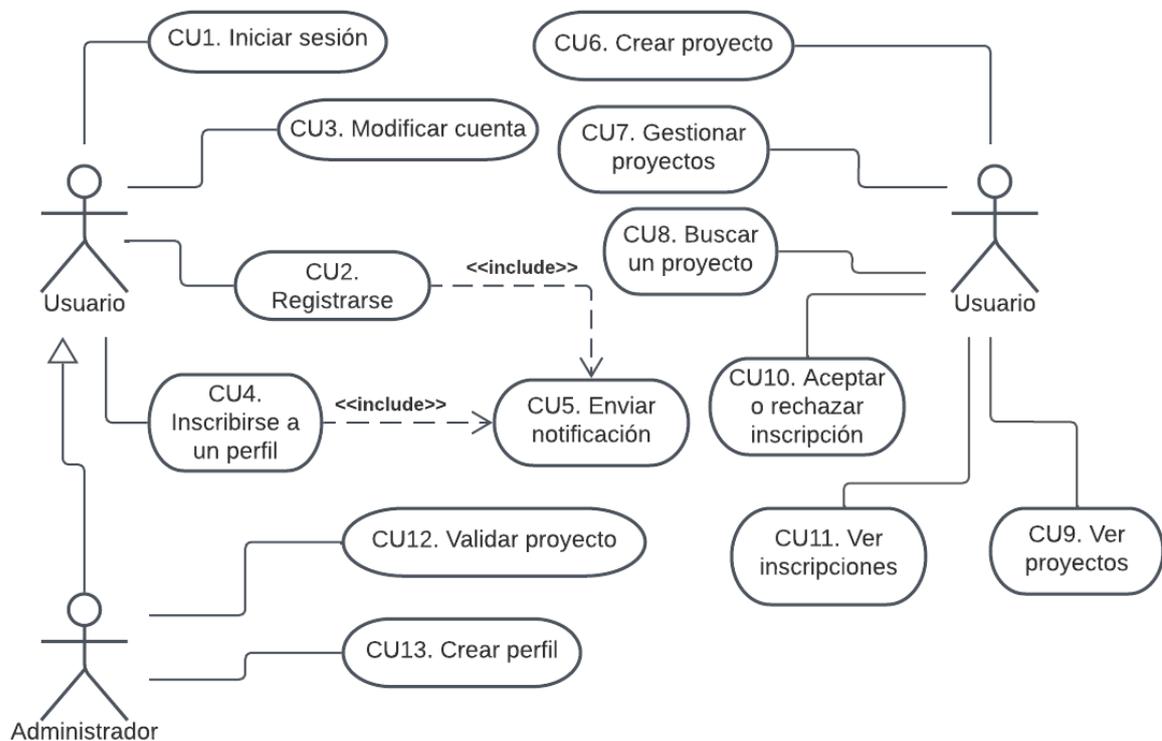


Figura 4. Diagrama de casos de uso de la aplicación

Id y Nombre	CU1. Iniciar sesión
Descripción	El usuario inicia sesión en la aplicación.
Precondición	El usuario debe estar registrado en la aplicación.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de sesión. 2. El usuario introduce su correo electrónico y su contraseña. 3. El usuario pulsa el botón de "Iniciar sesión". 4. El sistema verifica las credenciales del usuario.
Alternativas / Errores	<ol style="list-style-type: none"> 4a. Si el correo electrónico introducido por el usuario no está registrado en la aplicación, el sistema muestra un mensaje de error indicando que el correo electrónico no está registrado. 4b. Si la contraseña introducida por el usuario no coincide con la registrada en la aplicación, el sistema muestra un mensaje de error indicando que la contraseña es incorrecta.
Postcondición	El usuario es redirigido a la vista principal de la aplicación.

Id y Nombre	CU2. Registrarse
Descripción	Proceso para que un usuario se registre en la aplicación con verificación de correo.
Precondición	<ul style="list-style-type: none"> + El usuario debe tener una dirección de correo electrónico de una cuenta universitaria. + El usuario no debe estar registrado previamente en la aplicación.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de registro en la aplicación. 2. El usuario introduce su nombre, apellidos, dirección de correo electrónico y una contraseña. 3. El usuario pulsa el botón "Registrarse". 4. El sistema verifica que la dirección de correo electrónico introducida pertenece a una cuenta universitaria. 5. El sistema envía un correo electrónico al usuario con un código de verificación. 6. El usuario introduce el código de verificación recibido en el correo electrónico en la aplicación. 7. El sistema verifica que el código de verificación sea correcto. 8. El sistema registra al usuario y le dirige a la vista principal.
Alternativas / Errores	<ol style="list-style-type: none"> 4a. Si el sistema detecta que la dirección de correo electrónico introducida no pertenece a una cuenta universitaria, muestra un mensaje de error informando al usuario del problema y le indica que utilice una dirección de correo electrónico de una cuenta universitaria. 5a. Si el sistema no puede enviar el correo electrónico al usuario, muestra un mensaje de error informando al usuario del problema y le indica que compruebe su conexión a internet o que intente registrarse más tarde. 7a. Si el usuario introduce un código de verificación incorrecto, muestra un mensaje de error informando al usuario del problema y le indica que introduzca un código de verificación válido.
Postcondición	El usuario ha sido registrado y puede acceder a su perfil.

Id y Nombre	CU3. Modificar cuenta
Descripción	Proceso para que un usuario registrado pueda modificar su cuenta en la aplicación.
Precondición	El usuario debe haber iniciado sesión en la aplicación.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede a su cuenta en la aplicación. 2. El usuario hace clic en el botón "Cuenta". 3. El sistema muestra un formulario con los campos de perfil del usuario, incluyendo su nombre, apellidos y dirección de correo electrónico. 4. El usuario modifica los campos que desea cambiar. 5. El usuario pulsa el botón "Guardar cambios". 6. El sistema verifica que los cambios sean válidos y actualiza la cuenta del usuario. 7. El sistema redirige al usuario a su cuenta actualizada.
Alternativas / Errores	6a. Si el sistema detecta que los cambios no son válidos (por ejemplo, si el usuario ha introducido un correo electrónico incorrecto), muestra un mensaje de error informando al usuario del problema.
Postcondición	La cuenta del usuario ha sido actualizada con la información modificada.



La Figura 5 muestra la vista de la cuenta del usuario. Se puede modificar la imagen de perfil o si procede eliminarla. Datos como el nombre, la dirección de correo electrónico se pueden cambiar gracias al formulario disponible. El apartado siguiente consta del componente para subir un CV o documento de presentación hacia los organizadores de los perfiles a los que se inscriba el usuario. En ese conjunto el usuario puede actualizar el documento o descargar el actual para estar al tanto de mantener la información actualizada.

El último aspecto consta del activador para abrir el formulario de cambio de contraseña y un botón para cerrar la sesión activa del usuario en ese momento.

Figura 5. Prototipo de la cuenta de usuario

Id y Nombre	CU4. Inscribirse a un perfil
Descripción	Un usuario se inscribe en un perfil de un proyecto.
Precondición	+ El usuario ha iniciado sesión en el sistema. + El proyecto y el perfil al que desea inscribirse existen y están validados por el administrador. + Todavía hay plazas disponibles en el perfil al que desea inscribirse.
Secuencia principal	1. El usuario accede a la lista de proyectos en la vista principal. 2. El usuario selecciona el proyecto al que desea inscribirse. 3. En la vista del proyecto, el usuario selecciona el perfil al que desea inscribirse. 4. El usuario pulsa el botón "Inscribirse" del perfil seleccionado. 5. El sistema muestra un mensaje de confirmación y envía una notificación al creador del proyecto. 6. El creador del proyecto puede aceptar o rechazar la inscripción del usuario en el perfil seleccionado. 7. Si el creador del proyecto acepta la inscripción del usuario, el sistema envía una notificación al usuario confirmando la inscripción.
Alternativas / Errores	4a. Si el perfil seleccionado ya está completo, el sistema bloquea el botón de inscripción. 4b. Si el usuario ya está inscrito en el perfil seleccionado, el sistema muestra el botón de "Retirar" la inscripción.
Postcondición	+ El usuario está inscrito en el perfil seleccionado y aparece en la lista de inscritos en la vista del perfil. + El sistema envía notificaciones al creador del proyecto y al usuario en función de la acción tomada por el creador del proyecto.

Id y Nombre	CU5. Enviar notificación
Descripción	Este caso de uso describe el envío de notificaciones por correo electrónico a los usuarios de la aplicación.
Precondición	El usuario debe tener una dirección de correo electrónico registrada en la aplicación.
Secuencia principal	1. El sistema determina que se necesita enviar una notificación por correo electrónico al usuario. 2. El sistema recopila la información necesaria para enviar la notificación, como la dirección de correo electrónico del usuario y el contenido de la notificación. 3. El sistema envía la notificación por correo electrónico al usuario.
Alternativas / Errores	1a. Si el sistema no puede determinar que se necesita enviar una notificación por correo electrónico al usuario, no se enviará ninguna notificación. 2a. Si la dirección de correo electrónico del usuario no está registrada en la aplicación, no se enviará ninguna notificación. 3a. Si el sistema no puede enviar la notificación por correo electrónico al usuario, se informará al usuario de que la notificación no pudo ser entregada.
Postcondición	El usuario recibirá una notificación por correo electrónico.

Id y Nombre	CU6. Crear proyecto
Descripción	Proceso que sigue un usuario para crear un nuevo proyecto dentro de la aplicación.
Precondición	El usuario debe estar registrado e identificado en la aplicación.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación y selecciona la opción "Crear proyecto". 2. La aplicación muestra un formulario en el que el usuario debe rellenar la información del proyecto, como el título, descripción, objetivo y perfiles necesarios. 3. El usuario completa el formulario y selecciona la opción "Crear proyecto". 4. La aplicación valida los datos introducidos y crea un nuevo proyecto en la base de datos con el estado "pendiente de validación". 5. La aplicación muestra un mensaje al usuario informando de que el proyecto se ha creado correctamente y se encuentra pendiente de validación.
Alternativas / Errores	<ol style="list-style-type: none"> 2a. Si el usuario no ha rellenado alguno de los campos obligatorios del formulario, la aplicación muestra un mensaje de error informando de que es necesario completar todos los campos obligatorios. 4a. Si los datos introducidos por el usuario no son válidos, la aplicación muestra un mensaje de error informando de que no se ha podido crear el proyecto.
Postcondición	El proyecto se ha creado en la base de datos con el estado "pendiente de validación". El usuario creador del proyecto es el propietario del proyecto y puede modificar su información o añadir nuevos perfiles necesarios. El administrador de la aplicación recibirá una notificación para validar el proyecto.

Id y Nombre	CU7. Gestionar proyectos
Descripción	Este caso de uso describe cómo el usuario puede gestionar sus proyectos. El usuario puede ver una lista de sus proyectos y acceder a cada uno de ellos para modificar su información.
Precondición	El usuario debe haber iniciado sesión y tener al menos un proyecto creado en su cuenta.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede a la sección "Mis proyectos". 2. El sistema muestra una lista con los proyectos creados por el usuario. 3. El usuario selecciona el proyecto que desea gestionar. 4. El sistema muestra la información del proyecto. 5. El usuario modifica la información que desea cambiar. 6. El usuario pulsa el botón "Guardar cambios". 7. El sistema guarda los cambios y muestra un mensaje de confirmación.

Alternativas / Errores	<p>1a. Si el usuario no ha iniciado sesión, el sistema muestra un mensaje de error y le redirige a la página de inicio de sesión.</p> <p>3a. Si el proyecto seleccionado no existe, el sistema muestra un mensaje de error y vuelve a la lista de proyectos del usuario.</p> <p>5a. Si el usuario introduce datos incorrectos o incompletos, el sistema muestra un mensaje de error y no permite guardar los cambios.</p> <p>6a. Si se produce un error al guardar los cambios, el sistema muestra un mensaje de error y no guarda los cambios.</p>
Postcondición	El sistema actualiza la información del proyecto modificado en la base de datos.

Id y Nombre	CU8. Buscar un proyecto
Descripción	Permite al usuario buscar proyectos en la aplicación utilizando un término de búsqueda o filtrado.
Precondición	El usuario debe haber iniciado sesión en la aplicación.
Secuencia principal	<p>1. El usuario accede a la vista principal de la aplicación.</p> <p>2. El usuario escribe un término de búsqueda en el campo de búsqueda, añade algún filtro pulsa el botón "Aplicar".</p> <p>4. La aplicación muestra una lista de proyectos que coinciden con el término de búsqueda.</p> <p>5. El usuario puede hacer clic en un proyecto para ver más detalles.</p>
Alternativas / Errores	<p>2a. Si el usuario no ha introducido un término de búsqueda válido, la aplicación muestra un mensaje de error y no realiza la búsqueda.</p> <p>3a. Si no hay proyectos que coincidan con el término de búsqueda, la aplicación muestra un mensaje indicando que no se han encontrado resultados.</p>
Postcondición	<p>+ El usuario puede ver una lista de proyectos que coinciden con el término de búsqueda.</p> <p>+ El usuario puede ver los detalles de un proyecto específico si hace pulsar sobre él.</p>

Id y Nombre	CU9. Ver proyectos
Descripción	Este caso de uso describe cómo un usuario puede ver los proyectos existentes en la plataforma.
Precondición	El usuario ha iniciado sesión.
Secuencia principal	<p>1. El usuario accede a la vista principal de la aplicación.</p> <p>2. En la vista principal, el usuario puede deslizar los proyectos hacia arriba o hacia abajo para ver más proyectos.</p> <p>3. El usuario puede filtrar los proyectos por texto, perfil y si es válido para TFG/TFM.</p> <p>4. El usuario puede pulsar sobre un proyecto para verlo.</p>
Alternativas / Errores	Si no hay proyectos en la plataforma, el usuario verá un mensaje indicando que no hay proyectos disponibles.
Postcondición	El usuario ha visto los proyectos disponibles en la plataforma.

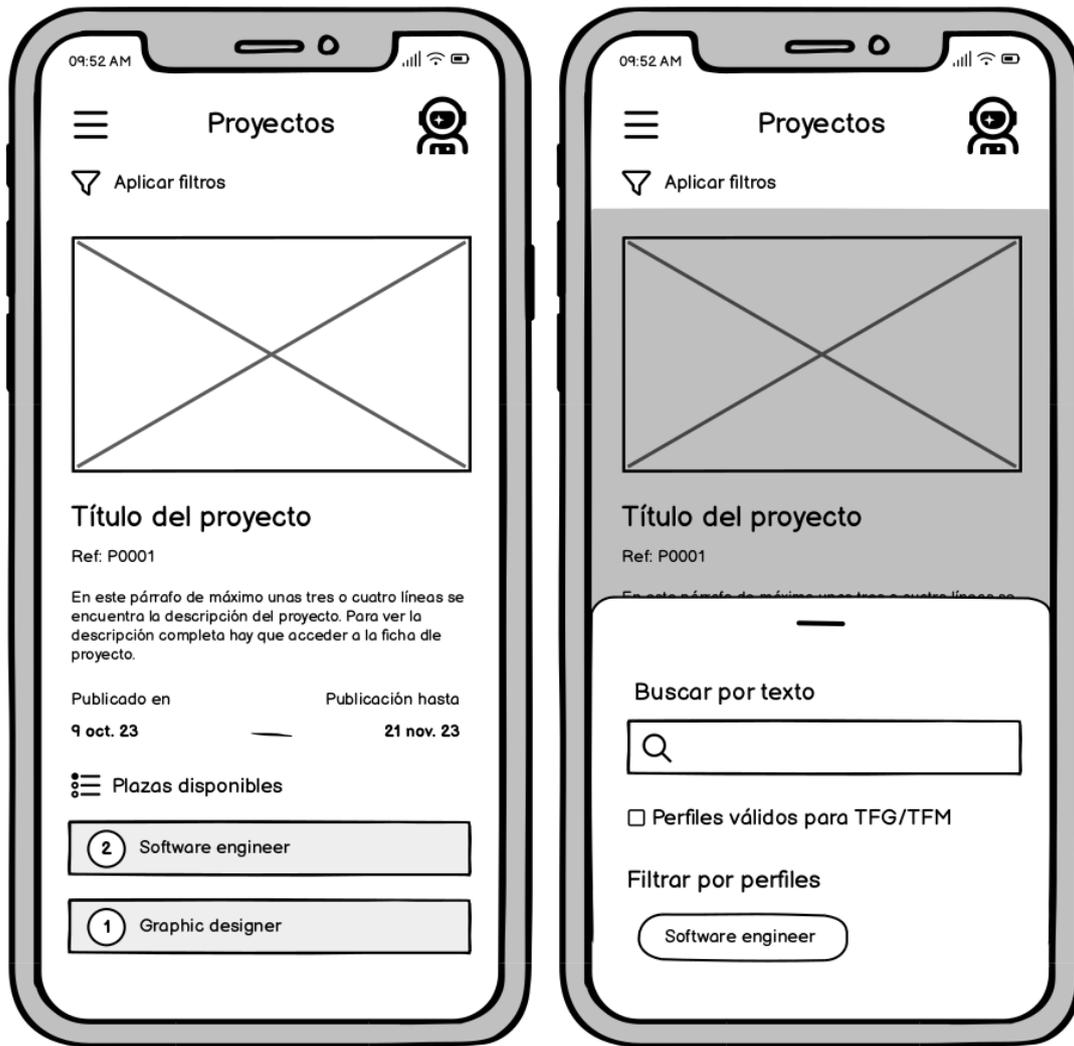


Figura 6. Prototipo de la vista principal de la interfaz

La vista principal de la aplicación cuando el usuario ha iniciado sesión tiene el siguiente aspecto (Figura 6). La interfaz cuenta con una barra superior con tres elementos; botón que activa la navegación a través de un componente de navegación de tipo *drawer*, el título de la ventana actual y la imagen de perfil del usuario que además es un disparador que lanza la vista de la cuenta del usuario. Justo debajo se encuentra el pulsador para activar el componente de búsqueda y filtros, tal y como se puede ver en el prototipo de la derecha. No es más que un componente de aspecto *bottom sheet*, donde el usuario puede realizar la búsqueda y aplicar los filtros que necesite. Por último, se encuentra la lista de proyectos optimizada para cargar en la memoria del dispositivo sólo lo que el usuario ve en la pantalla. Para visualizar el siguiente proyecto tiene que deslizar hacia arriba y para ver el anterior debe deslizar hacia abajo. En el aspecto de usabilidad es un gesto ampliamente extendido e integrado en los usuarios de dispositivos móviles.

Id y Nombre	CU10. Aceptar o rechazar una inscripción
Descripción	El creador del proyecto puede aceptar o rechazar las solicitudes de inscripción en los perfiles del proyecto.
Precondición	+ El usuario debe haber iniciado sesión. + El usuario debe ser el creador del proyecto. + Debe existir al menos una solicitud de inscripción en alguno de los perfiles del proyecto.
Secuencia principal	1. El usuario accede a su lista de proyectos en "Mis proyectos". 2. El usuario selecciona el proyecto que desea gestionar. 3. El usuario accede a la sección de "Solicitudes de inscripción". 4. El sistema muestra la lista de solicitudes de inscripción pendientes. 5. El usuario selecciona la solicitud que desea aceptar o rechazar. 6. El sistema muestra la información del solicitante y los perfiles a los que se ha inscrito. 7. El usuario puede aceptar o rechazar la solicitud de inscripción. 8. El sistema actualiza la base de datos para reflejar esa decisión. 9. El sistema envía una notificación al solicitante informando de la decisión del creador del proyecto.
Alternativas / Errores	4a. El sistema muestra un mensaje informando al usuario que no existen solicitudes de inscripción pendientes. 8a. Si se produce un error al actualizar la base de datos, el sistema muestra un mensaje de error informando al usuario que no se ha podido actualizar la acción solicitada.
Postcondición	+ Si se acepta la solicitud de inscripción, el usuario queda inscrito en el perfil del proyecto correspondiente. + Si se rechaza la solicitud de inscripción, el usuario no queda inscrito en el perfil del proyecto correspondiente.

Como se ha hecho especial hincapié en el ciclo de las inscripciones, es importante definir expresamente los nombres de cada uno de los estados en los que se puede encontrar la solicitud de participación de un usuario a un perfil de un proyecto. Para realizar tal hazaña se opta por una definición semi formal a través del diagrama de estados de la Figura 7. Quedando los nombres y transiciones de la siguiente manera:



Figura 7. Diagrama de estado de solicitudes a perfiles

Las acciones de aceptar o rechazar son realizadas por el usuario que ha creado el proyecto y está gestionando las solicitudes de los perfiles de su proyecto. Además, en todo momento puede deshacer una acción dejando de nuevo el estado en “solicitado”. Los eventos de solicitar o desapuntarse corresponden a la parte de usuario que se quiere inscribir a un perfil de un proyecto que es ajeno a los suyos.

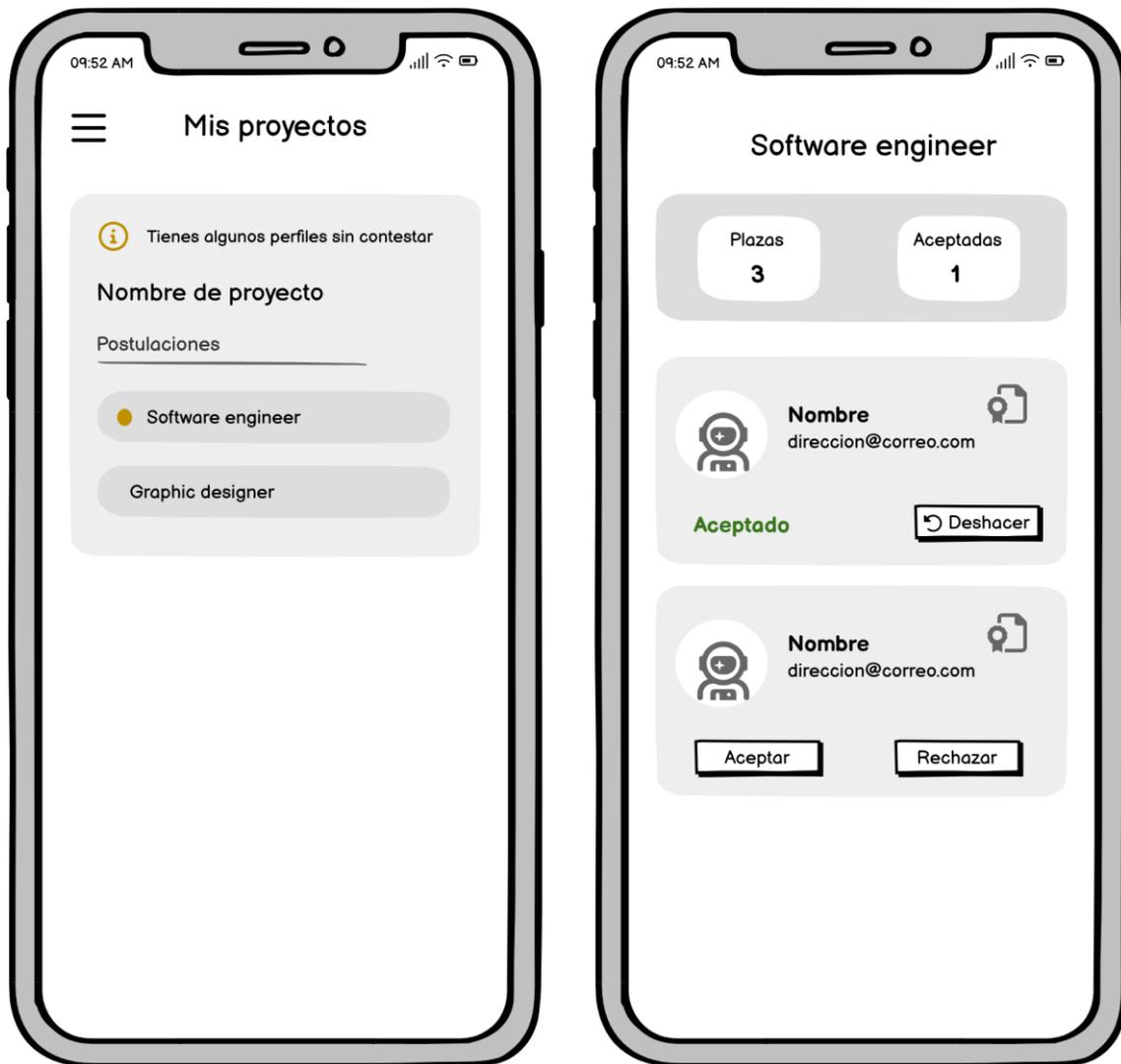
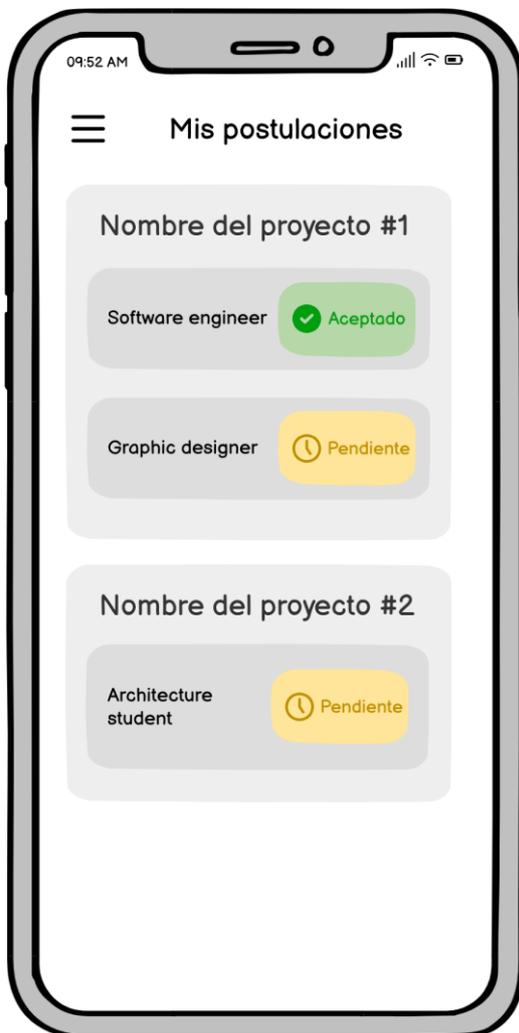


Figura 8. Prototipo de gestión de solicitudes

La Figura 8 es una representación de cómo gestionar las solicitudes, se accede desde la opción “Mis proyectos” de la navegación. El propio sistema debe advertir si hay alguna postulación pendiente de gestionar, tal como se muestra en el prototipo. A continuación, se pueden visualizar las plazas de ese perfil y las que han sido aceptadas. El usuario puede ver cada uno de los postulantes con su nombre, correo electrónico para ponerse en contacto y un documento que generalmente sería un CV. En este momento acepta o rechaza la inscripción y el sistema notificará la decisión al usuario. También se puede deshacer la operación.

Id y Nombre	CU11. Ver inscripciones
Descripción	Este caso de uso permite a los usuarios ver las inscripciones que han realizado en otros proyectos. Los usuarios pueden ver información sobre las inscripciones, como el nombre del proyecto, el perfil al que se han inscrito y su estado actual.
Precondición	+ El usuario debe haber iniciado sesión en el sistema. + El usuario se ha inscrito al menos a un perfil de un proyecto.
Secuencia principal	1. El usuario navega a la sección "Mis postulaciones". 2. Se muestran las inscripciones y el estado en el que se encuentran.
Alternativas / Errores	Si no hay inscripciones, se muestra un texto indicando que no hay inscripciones en ese momento.
Postcondición	No hay postcondiciones para este caso de uso.



La Figura 9 representa la vista de “Mis postulaciones”. Para llegar a ella, el usuario accede desde el menú principal a la sección de “Mis postulaciones”. Aquí aparecen en modo de cajas un conjunto representativo de cada una de las inscripciones. Este conjunto está formado por el título del proyecto y un listado de los perfiles a los que se ha inscrito el usuario y el estado actual en el que se encuentra esa inscripción al perfil. Si el usuario profesional sobre el título del proyecto puede visualizar la ficha específica en cualquier momento.

En caso de que haya una etiqueta marcada con el valor “Aceptado” o “No aceptado” se habrá lanzado una notificación al usuario y no debería demorarse la puesta en contacto más de cuatro días.

Figura 9. Prototipo de listado de postulaciones

Id y Nombre	CU12. Validar proyecto
Descripción	Este caso de uso describe el proceso de validación de un proyecto por parte del actor administrador del sistema. El objetivo de esta tarea es asegurarse de que el proyecto cumple con los estándares de calidad requeridos por el sistema y que cumple con los términos y condiciones del servicio. Si el proyecto no cumple con los estándares requeridos, el administrador debe rechazar el proyecto y notificar al propietario del proyecto sobre los motivos del rechazo.
Precondición	+ El administrador ha iniciado sesión en el sistema. + Existe un proyecto pendiente de validación.
Secuencia principal	<ol style="list-style-type: none"> 1. El administrador accede a la lista de proyectos pendientes de validación. 2. El administrador selecciona el proyecto a validar. 3. El administrador revisa los detalles del proyecto, como su descripción, objetivos y recursos requeridos. 4. El administrador verifica que el proyecto cumple con los estándares de calidad requeridos por el sistema. 5. Si el proyecto cumple con los estándares de calidad, el administrador aprueba el proyecto y notifica al propietario del proyecto sobre su aprobación. 6. Si el proyecto no cumple con los estándares de calidad, el administrador rechaza el proyecto y notifica al propietario del proyecto sobre los motivos del rechazo.
Alternativas / Errores	1a. No hay proyectos pendiente de validación. El sistema muestra al administrador un mensaje indicando que no hay proyectos.
Postcondición	+ El proyecto es aprobado y se convierte en un proyecto disponible en el sistema. + El proyecto es rechazado y no está disponible en el sistema.

Con el objetivo de clarificar la secuencia de principal del caso de uso y comprender bien el ciclo de vida de un proyecto en la aplicación, se pueden apreciar los diferentes estados y las transiciones entre éstos gracias al siguiente diagrama de estados de la Figura 10:

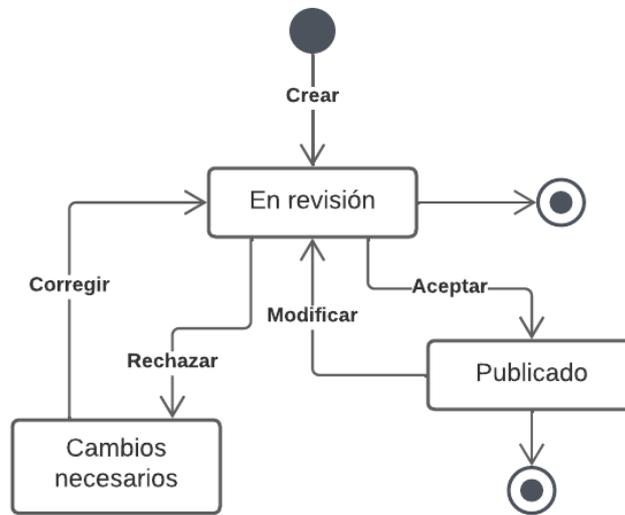


Figura 10. Diagrama de estados de un proyecto

Cuando un usuario crea un proyecto, éste queda en estado de revisión. Una vez el administrador revisa el proyecto puede aceptar el contenido y mover el proyecto al estado publicado o rechazar la publicación e indicar los motivos o los cambios necesarios para poder publicar correctamente el proyecto. Si el usuario realiza las correcciones oportunas y guarda los cambios, el proyecto volverá a estar en revisión.

Si el usuario ya tiene un proyecto publicado (ya fue aceptado por el administrador) y decide realizar algún cambio, el proyecto cambiará de nuevo a un estado de revisión. Cuando se encuentra en revisión o está publicado, el proyecto puede ser eliminado del sistema por el administrador o por el usuario.

Id y Nombre	CU13. Crear perfil
Descripción	Este caso de uso describe cómo el administrador crea un perfil en el sistema.
Precondición	El administrador ha iniciado sesión en el sistema.
Secuencia principal	<ol style="list-style-type: none"> 1. El administrador accede a la sección "Perfiles" del sistema. 2. El sistema muestra la lista de perfiles existentes en el sistema. 3. El administrador hace clic en el botón "Crear nuevo perfil". 4. El sistema muestra un formulario de creación de perfil. 5. El administrador completa el formulario con los datos del perfil. 6. El administrador hace clic en el botón "Crear". 7. El sistema crea el nuevo perfil y muestra un mensaje de confirmación al administrador.

Alternativas / Errores	<p>5a. Si el administrador no completa alguno de los campos requeridos del formulario, el sistema muestra un mensaje de error y resalta los campos que faltan.</p> <p>5b. Si el administrador completa el formulario con datos incorrectos o inválidos, el sistema muestra un mensaje de error y resalta los campos que tienen problemas.</p> <p>6a. Si el sistema detecta que ya existe un perfil con el mismo nombre, muestra un mensaje de error al administrador y le pide que elija otro nombre.</p>
Postcondición	El perfil se ha creado correctamente en el sistema. El perfil aparece en la lista de perfiles y los usuarios pueden inscribirse en él.

5.2.2. Requisitos no funcionales

Seguridad: La aplicación debe garantizar la seguridad de los datos personales de los usuarios y la confidencialidad de la información compartida en los proyectos.

- Autenticación y autorización: Debe contar con un sistema de autenticación y autorización robusto para garantizar la seguridad de los datos personales y la información compartida en los proyectos.
- Cifrado de datos: Cifrar los datos de usuario y las comunicaciones para asegurar la confidencialidad.

Usabilidad: La aplicación debe ser fácil de usar y permitir una experiencia de usuario agradable y satisfactoria.

- Diseño intuitivo y fácil de usar: Contar con un diseño intuitivo que permita a los usuarios navegar y acceder a la información de manera fácil y rápida.
- Aprendizaje: Fácil de aprender para los nuevos usuarios, sin requerir un entrenamiento extenso.

Rendimiento: La aplicación debe ser capaz de manejar grandes volúmenes de datos y proporcionar un tiempo de respuesta rápido.

- Tiempo de respuesta: Debe tener un tiempo de respuesta rápido, lo que significa que los usuarios deben ser capaces de realizar acciones y obtener resultados de forma eficiente.
- Escalabilidad: Ser capaz de manejar un gran número de usuarios y proyectos sin sufrir una pérdida significativa en el rendimiento.

Mantenibilidad: La aplicación debe ser fácil de mantener y evolucionar a medida que se agregan nuevas funcionalidades y mejoras.

- Modularidad: La aplicación debe ser modular para permitir la evolución independiente de las diferentes partes del sistema y reducir el acoplamiento.
- Legibilidad del código: El código fuente de la aplicación debe ser legible y fácil de entender para facilitar su mantenimiento.



Fiabilidad: La aplicación debe ser capaz de funcionar sin errores y proporcionar una experiencia de usuario estable y sin interrupciones.

- Tolerancia a fallos: La aplicación debe ser capaz de recuperarse de posibles fallos sin interrupciones en el servicio.
- Disponibilidad: Estar disponible para su uso en todo momento, con una tasa de disponibilidad alta.

5.3. Diseño

La arquitectura de sistema es un aspecto crucial en el desarrollo de cualquier aplicación, incluyendo las aplicaciones para móviles. En el caso de TeamMaker, una arquitectura de sistema sólida y bien pensada puede ser la clave para garantizar que la aplicación sea eficiente, escalable y fácil de mantener.

La arquitectura de sistema define la estructura general de la aplicación, incluyendo la forma en que los diferentes componentes interactúan entre sí y cómo se maneja la información en la aplicación. En el caso de una aplicación para móviles, esto incluye cómo se almacenan y acceden a los datos, cómo se gestiona la memoria y cómo se manejan las interacciones con el usuario. Una buena arquitectura de sistema también puede hacer que la aplicación sea más fácil de mantener y actualizar en el futuro. Por ejemplo, una estructura bien organizada puede facilitar la incorporación de nuevas funcionalidades o la corrección de errores en el código. Además, una arquitectura bien pensada puede permitir el uso de patrones de diseño conocidos y probados, lo que puede simplificar el desarrollo y hacer que el código sea más fácil de leer y entender.

La arquitectura de sistema es fundamental para el éxito de cualquier aplicación para móviles. En nuestro caso, un enfoque cuidadoso en el diseño y desarrollo de la arquitectura puede ayudarnos a crear una aplicación que sea eficiente, escalable y fácil de mantener, lo que a su vez puede mejorar la experiencia del usuario y el éxito comercial de la aplicación.

5.3.1. Arquitectura del sistema

Se plantea una arquitectura de tres capas lógicas [11] donde cada una de ellas desempeña una serie de funciones y procesos. Esta alternativa es una buena opción por diversos motivos. Mejora la gestión de código porque permite separar claramente las responsabilidades y funciones de la aplicación en diferentes capas lógicas, lo que hace que el código sea más modular, fácil de entender y mantener. Esto también facilita el mantenimiento cuando se realiza una actualización del sistema, ya que cada capa se puede modificar o actualizar de forma independiente sin afectar a las otras capas.

Una arquitectura de tres capas es escalable, lo que significa que se pueden agregar recursos para mejorar el rendimiento de cada capa de forma independiente. Esto permite que el sistema pueda crecer y adaptarse a las necesidades de los usuarios y del negocio.

Además, facilita las pruebas pues cada una de las partes se puede probar de forma independiente y después se puede probar la interacción entre las diferentes capas.

Capa de presentación

La capa de presentación es la capa más cercana al usuario, es la que presenta la información de manera visual y recoge las acciones del usuario. Para TeamMaker, la capa de presentación podría ser la aplicación móvil desarrollada en React Native, que se encargaría de mostrar la información de los proyectos publicados, permitir la búsqueda, mostrar la información al usuario, etc. En el caso de TeamMaker, el uso de React Native para la capa de presentación ofrece una serie de ventajas significativas. En primer lugar, ofrece una experiencia de usuario nativa. Esto significa que las aplicaciones móviles desarrolladas con React Native pueden ofrecer una experiencia de usuario fluida y altamente optimizada (cercano al rendimiento de una aplicación nativa). Las aplicaciones son capaces de aprovechar las características y funcionalidades nativas de los dispositivos móviles, como la cámara, el almacenamiento local y la conectividad, lo que mejora significativamente la experiencia de usuario. Permite el desarrollo de aplicaciones móviles para múltiples plataformas. Esto significa que una sola línea de trabajo puede crear una aplicación que funcione en múltiples plataformas móviles, como iOS y Android, lo que ahorra tiempo y recursos. Esto significa que se puede crear una sola aplicación que funcione en múltiples plataformas móviles, lo que aumenta la accesibilidad para los usuarios.

Capa de lógica de negocio

La capa de lógica de negocio es la capa que se encarga de la lógica de la aplicación, es decir, de las reglas que gobiernan el comportamiento de la aplicación. Para TeamMaker, esta capa podría ser implementada utilizando Node.JS y el framework Express.js, ya que permiten la creación de una API flexible y escalable que pueda manejar las funciones principales de la aplicación. Juntos, son la combinación perfecta para realizar la capa de lógica de negocio de una aplicación como TeamMaker.

Node.JS es conocido por su eficiencia y escalabilidad, lo que lo hace ideal para aplicaciones que manejan una gran cantidad de solicitudes y conexiones simultáneas. El modelo de programación asíncrono y no bloqueante con el que cuenta, permite que múltiples solicitudes se manejen al mismo tiempo, lo que acelera el tiempo de respuesta de la aplicación y garantiza que los usuarios no experimenten tiempos de espera prolongados. Además, la arquitectura modular permite personalizar la funcionalidad de la aplicación según las necesidades del usuario.

Express.js permite crear aplicaciones web de manera sencilla y rápida. Es fácil de aprender y utilizar, lo que permite crear aplicaciones complejas en menos tiempo y con menos esfuerzo. Esta combinación es muy versátil y combina muy bien con otras tecnologías. Se pueden utilizar diferentes herramientas y tecnologías para construir diferentes partes de la aplicación, lo que aumenta la flexibilidad y la capacidad de la aplicación para manejar diferentes tipos de solicitudes y conexiones. En otras palabras, se favorece el desacoplamiento y aumenta la mantenibilidad. Juntos, estas tecnologías



ofrecen eficiencia, escalabilidad, flexibilidad y facilidad de uso, aspectos que son necesarios para crear aplicaciones móviles de alta calidad y rendimiento.

Capa de persistencia

La capa de persistencia es la capa que se encarga de la gestión de los datos de la aplicación. Esta capa se encargaría de almacenar y recuperar los datos necesarios para la aplicación, como la información de los proyectos, la información de los usuarios y los registros de las interacciones. En TeamMaker, esta capa podría ser implementada utilizando bases de datos relacionales como MySQL. Ya que una de sus principales ventajas es su amplia adopción y madurez en la industria [15]. Esto significa que hay una gran cantidad de documentación, herramientas y recursos disponibles para trabajar con MySQL, lo que facilita la tarea a la hora de integrar la base de datos en la aplicación. Además, es la más extendida a nivel académico por lo que reduce la curva de aprendizaje y acelera el proceso de desarrollo.

Un aspecto clave en la elección frente a esta comparativa, ha sido la infraestructura para el despliegue. En el contexto donde se realiza este proyecto, una práctica en la cátedra HP-UPV, la infraestructura de despliegue no cuenta con soporte para la tecnología de bases de datos NoSQL. Por tanto, en este caso es más conveniente utilizar una base de datos relacional como MySQL.

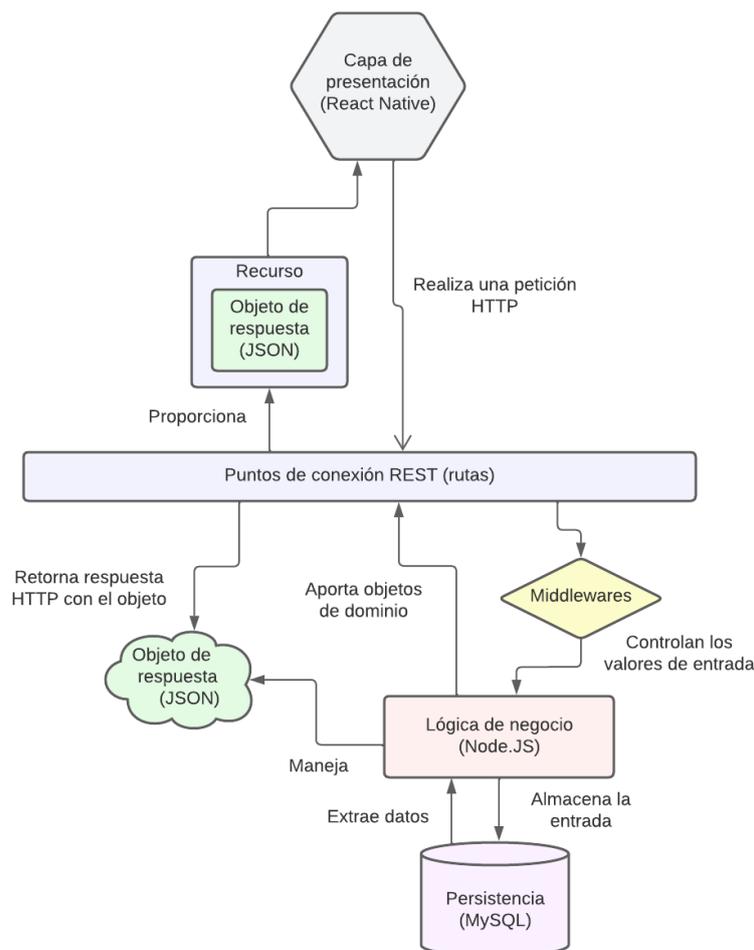


Figura 11. Arquitectura software del sistema

Tal como se muestra en el esquema de la Figura 11, la capa de presentación (aplicación para Android y iOS creada con React native) realiza una petición HTTP a alguno de los puntos de conexión de la API REST. Dependiendo de si en el cuerpo de la petición se encuentran valores de entrada o de si la petición requiere un *token* de autenticación, entrarán en el ciclo una serie de componentes software intermedios llamados *middlewares*. Estos componentes verifican ciertos parámetros de entrada que se detallarán posteriormente. A continuación, la lógica de negocio realiza el comportamiento necesario para atender la petición. Generalmente ese procesamiento o cómputo implicará una comunicación con la capa de persistencia, que proporciona la información necesaria sobre el recurso solicitado o bien almacenará nueva información dependiendo de la naturaleza del procesamiento. Finalmente, se genera el objeto respuesta, que es devuelto a la capa de presentación. Este objeto contiene siempre el código de respuesta (siguiendo el estándar HTTP [16] al tratarse de una API Web) y el cuerpo si lo hubiere.

La organización de las tecnologías usadas para el sistema se puede entender de manera gráfica (Figura 12) de la siguiente forma:



Figura 12. Tecnologías empleadas en cada una de las capas

5.3.2. Estructura detallada

A continuación, se describe con más detalle la estructura y responsabilidades de cada una de estas capas y cómo se comunican entre sí para llevar a cabo las funcionalidades de la aplicación. Además, se explica cómo se han implementado los diferentes componentes que conforman cada capa, como los enrutadores, controladores y middlewares.

Capa de presentación

En esta capa, se encuentra la interfaz de usuario de la aplicación, la cual se encarga de presentar la información al usuario de una manera amigable y fácil de usar. Como se ha comentado, esto se consigue con una aplicación para móviles escrita en React Native junto con el asistente de consola de Expo. En cuanto a la organización de los directorios y ficheros de la aplicación, el directorio principal de la capa de presentación se encuentra en la raíz del proyecto y se denomina "src". Dentro de este directorio, se encuentran los siguientes directorios:

- **assets:** Almacena todos los recursos gráficos utilizados en la aplicación, como imágenes y fuentes.
- **components:** Agrupa todos los componentes React utilizados en la interfaz de usuario. Estos componentes se utilizan para crear las diferentes pantallas de la aplicación.
- **navigation:** Gestiona todos los archivos de configuración de navegación utilizados por la aplicación.
- **screens:** Este directorio contiene todas las pantallas de la aplicación. Cada pantalla es un componente React que se encuentra en este directorio.
- **services:** Maneja todos los servicios utilizados por la aplicación para comunicarse con el backend.

La estructura de directorios de la aplicación se organiza de tal manera que los componentes y módulos estén agrupados según su función y responsabilidad. La comunicación entre los componentes se realiza mediante la técnica "*prop drilling*" o el "levantamiento de estado" (*lifting state up*), que consiste en pasar datos y funciones

desde un componente padre a sus componentes hijos como propiedades. Esto permite que los componentes hijos puedan acceder y modificar los datos y funciones del componente padre. Por ejemplo, si se busca que la pantalla de la ficha del proyecto y la de edición de ese mismo proyecto contengan el mismo objeto respuesta obtenido de la API, se puede realizar la técnica del “prop drilling” y crear un componente que agrupe estas dos vistas (que a la vez son componentes siguiendo las reglas de React). De esa forma se puede compartir la información que no sea sensible entre varias ventanas.

Por otro lado, la comunicación entre los módulos se realiza mediante la importación y uso de las funciones y clases definidas en los módulos. Los módulos pueden exportar una o varias funciones o clases, las cuales pueden ser importadas en otros módulos o componentes para ser utilizadas.

Por tanto, la comunicación entre los directorios de la aplicación se basa en la definición clara de las responsabilidades de cada componente y módulo, y en la utilización de técnicas como la “*prop drilling*” y la importación de módulos para permitir el intercambio de datos y funciones entre ellos de forma efectiva y organizada.

Flujo de navegación

El directorio “*navigation*” dentro de la estructura de directorios de la capa de presentación, es uno de los directorios más importantes en la estructura de directorios de TeamMaker en React Native, ya que es el encargado de gestionar la navegación de la aplicación. En una aplicación móvil, la navegación es esencial para permitir que los usuarios se muevan por la aplicación y accedan a las diferentes funcionalidades y pantallas. Por lo tanto, es importante que la navegación sea intuitiva y fácil de usar. En el caso de TeamMaker, el directorio *navigation* contiene los archivos que definen la navegación de la aplicación, como por ejemplo las diferentes pantallas o vistas que se pueden acceder desde el menú principal, la navegación de pila que permite retroceder y avanzar entre las pantallas, y la configuración de la barra de navegación. En el caso de la organización de los elementos de navegación de la aplicación es algo sofisticada. Se ha utilizado el potente framework React Navigation¹⁸ desarrollado para React Native que ofrece utilidades para manejar los flujos de navegación en aplicaciones. Siguiendo las buenas prácticas y recomendaciones de la documentación, se ha logrado generar una estructura que combina el uso de *reducers* [17], almacenadores de contexto y el concepto de memoización [18].

Para generar el *reducer* (reductor), se emplea la herramienta proporcionada de forma nativa *useReducer* de React a través de un *hook*. Un *hook* es una utilidad poderosa para añadir funcionalidad a un componente de React. Al utilizar hooks, se pueden agregar características como el estado, el ciclo de vida y los efectos secundarios a un componente de función (los utilizados en el proyecto). El hook *useReducer* es una función proporcionada por React que nos permite manejar el estado de nuestra aplicación de manera más efectiva y organizada. En lugar de manipular el estado directamente, *useReducer* nos permite actualizarlo mediante acciones que se describen con objetos.

¹⁸ Sitio web oficial de React Navigation: <https://reactnavigation.org>



Estos objetos contienen información sobre el tipo de acción que se debe realizar y cualquier dato adicional que se deba pasar.

En el caso de TeamMaker, el estado de autenticación del usuario es importante para determinar qué contenido se debe mostrar. En lugar de manejar esto de forma aislada en cada componente, se utiliza *useReducer* para mantener un estado centralizado y actualizarlo de manera uniforme en toda la aplicación. Para esto, se puede crear un contexto de autenticación utilizando el componente *AuthContext.Provider* [19]. Este contexto puede contener el estado de autenticación, el objeto de usuario y las acciones que permiten actualizar el estado. Las acciones pueden incluir iniciar la sesión, terminarla, registrarse, entre otras. Es decir, existe un componente que almacena el estado general de la sesión del usuario. Por ejemplo, conocer si existe una sesión activa, un *token* válido, etc. Por otro lado, se encuentra el *reducer*, que permite modificar el estado de este contexto global de la aplicación, de manera uniforme en todo el sistema. La función reductora se encarga de actualizar el estado en función de la acción recibida y devuelve el nuevo estado actualizado.

Finalmente, para mostrar el contenido en función del estado de autenticación, se puede utilizar la memoización en combinación con *AuthContext.Provider*. Al memoizar el componente que muestra el contenido, podemos evitar que se vuelva a renderizar cada vez que se actualiza el estado de autenticación. En cambio, el contenido se mostrará automáticamente en función del estado actualizado, lo que mejora el rendimiento y la eficiencia. El uso de *useReducer* junto con *AuthContext.Provider* y la memoización permite manejar el estado de autenticación de manera centralizada y eficiente en toda la aplicación, lo que permite mostrar el contenido de manera más efectiva y organizada.

Se puede interiorizar toda esta estructura entendiendo la siguiente representación gráfica que describe estas herramientas para la navegación y los diferentes componentes de navegación:

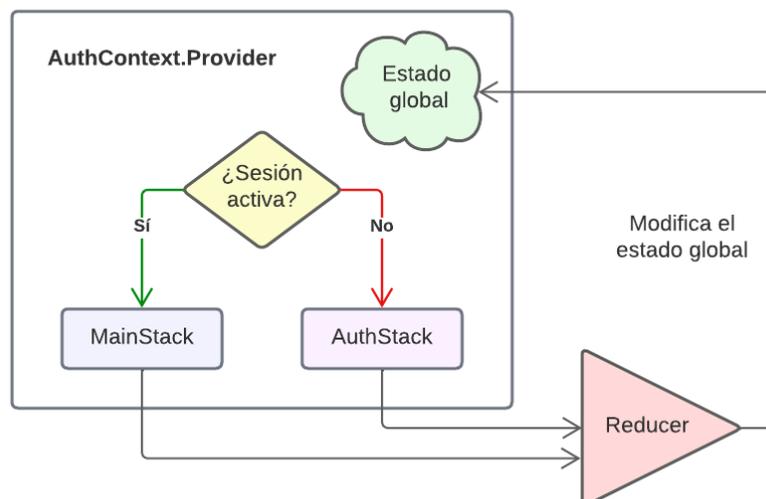


Figura 13. Flujo de navegación de la aplicación

Como se puede ver en la Figura 13, al acceder a la aplicación se comprueba si existe una sesión activa y gracias al proveedor de contexto se puede almacenar el estado global. Los dos componentes de navegación que se pueden cargar en función del estado son:

- **MainStack:** Contiene el navegador de la vista principal del usuario, cuando tiene una sesión activa. En esta pila de vistas, gracias al estado global se puede ver si el usuario tiene rol de administrador para mostrar las ventanas adicionales para este tipo de usuario.
- **AuthStack:** Esta pila contiene las vistas de un usuario que no tiene una sesión activa en la aplicación. Estas vistas son: iniciar sesión, registro, recuperar la contraseña, validar una cuenta.

Gracias al reductor, se puede modificar el estado global. Por ejemplo, cuando un usuario con una sesión activa decide terminar su sesión, se hará una llamada al reductor que se encuentra memoizado con el contexto y éste cambiará el estado eliminando la sesión del usuario.

Capa de lógica de negocio

Para estructurar de manera adecuada la capa de lógica de negocio en TeamMaker, se hace necesario definir una organización adecuada de los directorios y archivos que permitan una fácil lectura y mantenimiento del código. En la raíz de la capa de lógica de negocio de TeamMaker, se pueden encontrar varios directorios, entre los cuales se destacan:

- **controllers:** Contiene los controladores de las diferentes entidades que maneja la aplicación. Cada controlador es responsable de gestionar las operaciones de lectura, escritura, actualización y eliminación (CRUD) de una entidad específica. Por ejemplo, podría existir un controlador para los usuarios, otro para los proyectos, otro para los perfiles, entre otros.
- **middlewares:** Almacena los *middlewares* que se utilizan en la aplicación. Los *middlewares* en este contexto son funciones que se ejecutan antes o después de una petición HTTP y que pueden realizar tareas como la validación de datos, la autenticación de usuarios, la gestión de errores, entre otras.
- **models:** en este directorio se definen los modelos de las diferentes entidades que maneja la aplicación. Cada modelo representa una tabla en la base de datos y define los campos y relaciones de cada entidad.
- **routes:** en este directorio se encuentran los archivos de enrutamiento de la aplicación. Cada archivo de enrutamiento define las rutas HTTP que se pueden utilizar para acceder a los diferentes recursos de la aplicación y especifica el controlador que se encargará de gestionar cada una de estas rutas.
- **services:** en este directorio se pueden encontrar diferentes servicios que se utilizan en la aplicación. Estos servicios pueden ser utilizados por los controladores o por los *middlewares* para realizar tareas específicas, como el envío de correos electrónicos, la generación de *tokens* de autenticación, la carga de archivos, entre otras.



Además, se dejan a disposición tres directorios con el nombre “*uploads*”, “*userPics*” y “*resumes*” que serán de tipo estático. Es decir, tendrán una ruta estática propia a través del servicio de la lógica de negocio y en estos directorios se almacenará el contenido multimedia de los proyectos, las imágenes de perfil de los usuarios y los documentos respectivamente. Para manejar este tipo de contenido en una aplicación Node.JS con Express.js, es recomendable utilizar rutas estáticas con directorios para almacenar y servir este tipo de contenido.

Una ruta estática es una URL que no cambia, es decir, siempre apunta al mismo archivo o directorio. Al utilizar rutas estáticas para los archivos multimedia, se puede especificar una ruta que apunte a un directorio en el servidor que contenga los archivos. De esta forma, cuando el navegador del usuario solicita el archivo, Express.js puede buscar el archivo en el directorio y devolverlo como respuesta sin necesidad de realizar ninguna operación adicional.

Esto es especialmente útil en aplicaciones que muestran muchas imágenes o archivos de multimedia, ya que al tener rutas estáticas con directorios para cada tipo de archivo, se pueden servir de manera más eficiente, reduciendo el tiempo de carga y mejorando el rendimiento de la aplicación. Además, al utilizar rutas estáticas se pueden aplicar medidas de seguridad, como restringir el acceso a ciertos directorios, evitar la ejecución de archivos que pudieran ser peligrosos para la aplicación o limitar el tamaño de los archivos que se pueden subir.

Generando un sumario con aspecto gráfico, el flujo de control de las partes de la lógica de negocio generalmente se produce siguiendo la siguiente dinámica representada en la Figura 14:

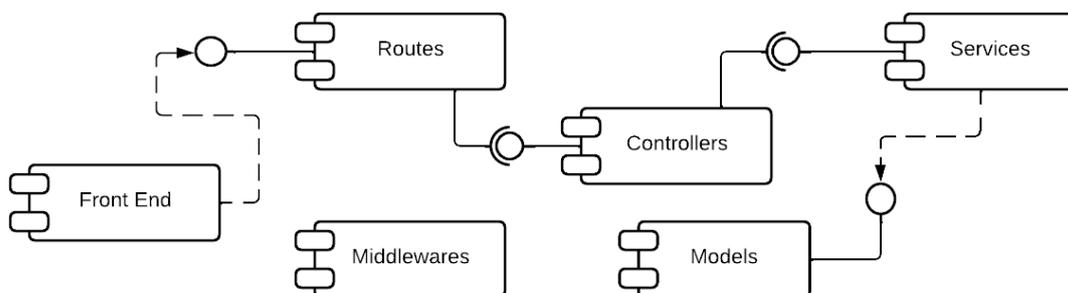


Figura 14. Diagrama de componentes del sistema

Un elemento de la capa de presentación realiza una petición a un punto de conexión de la lógica de negocio (una ruta de la API). El enrutador llama a la batería de *middlewares* si procede y se deja el contexto y los valores de entrada preparados para que el controlador haga lo propio. Este controlador realiza la operación correspondiente y hace uso de los servicios disponibles. Por ejemplo, puede accionar un servicio para un registro de usuario y otro servicio para enviar un mensaje de correo electrónico con un código de verificación.

Capa de persistencia

En MySQL, los datos se organizan en tablas, que contienen filas y columnas. Cada tabla se utiliza para almacenar información específica, como usuarios, proyectos, perfiles, etc. La estructura de las tablas se define mediante un esquema, que especifica el nombre de la tabla, el nombre y tipo de datos de cada columna, y las restricciones que se aplican a cada columna.

Para interactuar con MySQL desde la capa de lógica de negocio, se utilizan un módulo de conexión de Node.JS; mysql. Este módulo permite conectarse a la base de datos, ejecutar consultas y recibir los resultados de forma asíncrona.

La estructura de directorios de la capa de persistencia de la aplicación consiste en establecer un fichero de configuración exportable llamado “*db.js*” que se encuentra dentro del directorio “*database*”. Este fichero tiene los parámetros de conexión y configuración con la base de datos que se encuentra en otro servicio. Además de este fichero, el directorio podrá contener posteriormente otros ficheros de configuración o definiciones de esquemas.

Una aproximación para entender la estructura de la base de datos con un cierto nivel de abstracción puede ser el siguiente diagrama de clases de la Figura 15:

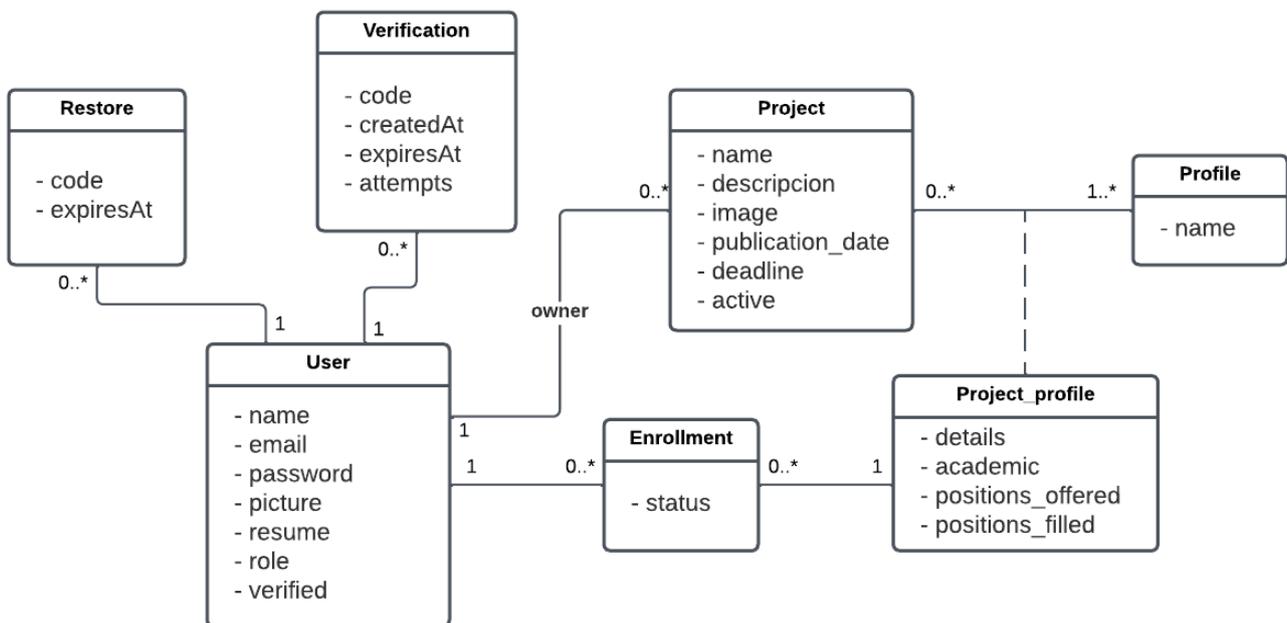


Figura 15. Diagrama de clases de la aplicación

Un usuario, en algún momento del ciclo de vida en la aplicación cuenta con una entrada en la tabla de verificaciones (pendientes). Si solicita restablecer el acceso a su cuenta tendría esa entrada en la tabla de restablecimientos. Cuando un usuario registra un proyecto, debe añadir al menos un perfil disponible del sistema. En ese momento el usuario señala la cantidad de posiciones ofrecidas, una descripción de la implicación en

ese proyecto, etc. Es por esto por lo que es necesaria esa clase asociación para que esa información sea única entre un perfil de la base de datos y el proyecto en cuestión.

Al descender un nivel en la definición del esquema de la base de datos, se puede apreciar como son necesarios algunos cambios de diseño para adaptar ese esquema a un modelo real compatible con una base de datos relacional. Aquí es donde aparece el diseño de la base de datos mediante un diagrama de Entidad Relación (ER).

En las diferentes clases que reciben una relación, se tiene que añadir un atributo (columna de la tabla) con el mismo tipo que la clave primaria de la tabla destino (clase relacionada). En la clase asociación al tratarse también como la solución de diseño para la asociación de “muchos a muchos” entre Proyecto y Perfil, se deben indicar estas dos clases en la nueva tabla y definir una clave primaria que engloba sendas tablas. Después tanto la tabla de proyectos como la de perfiles tendrán una clave ajena apuntando a esta nueva tabla que desempeña también la función de clase asociación (aportando información única en la relación). Estos cambios pueden verse reflejados de una forma más concisa con un diagrama extraído del script SQL del sistema con ingeniería inversa a través de una herramienta de automatización como diagrams.net¹⁹ (ver Figura 16).

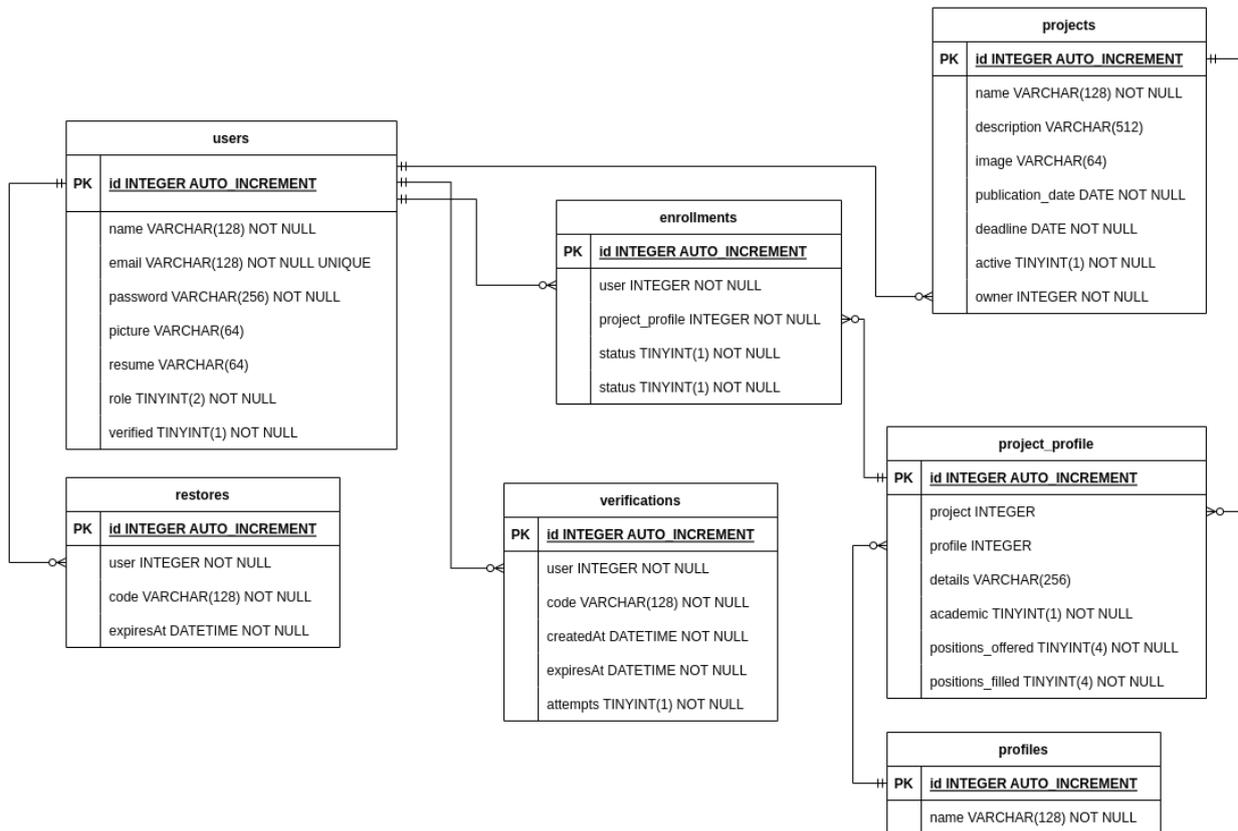


Figura 16. Diagrama Entidad Relación de la aplicación

¹⁹ Sitio web oficial de diagrams.net: <https://www.diagrams.net>

5.4. Programación

Ya se conoce tanto la arquitectura como tecnología base para el desarrollo e implementación de cada una de las capas del sistema software. El primer paso para poder realizar el desarrollo de la aplicación es entender los lugares donde se van a alojar o desplegar cada uno de los componentes. De esa forma la creación del producto estará asentada y optimizada desde el primer momento para las plataformas de despliegue necesarias. Debido a que este proyecto está amparado bajo una práctica de empresa, hay una parte de la infraestructura de despliegue que está preseleccionada por la organización. En este caso tanto la base de datos como la lógica de negocio se ubicarán en la plataforma Plesk ²⁰ que ofrece el sistema de webs avanzadas de la UPV ²¹.

Plesk es una plataforma de administración de servidores que permite a los usuarios gestionar fácilmente sus sitios web, aplicaciones y servicios en línea. Proporciona una interfaz gráfica de usuario intuitiva y herramientas automatizadas para simplificar la gestión de servidores web, bases de datos y correos electrónicos. La interfaz de usuario de Plesk hace que sea fácil y rápido realizar copias de seguridad de la base de datos y configurar medidas de seguridad adicionales. Esto es muy conveniente para el despliegue de la capa de persistencia en esta plataforma. Además, ofrece soporte por defecto para lanzar un proceso en ejecución de la lógica de negocio en Node.JS y automatiza muchos procesos a través de su interfaz gráfica como puede ser la instalación de los paquetes y las dependencias de la aplicación, atendiendo a las versiones correspondientes de cada paquete. Contar con una plataforma que integra el alojamiento de los directorios del proyecto, el despliegue de la lógica de negocio y la gestión de bases de datos en un mismo lugar es algo que agiliza mucho el proceso y la manipulación del sistema.

A continuación, se muestran algunos aspectos o procedimientos que se han tenido en cuenta en el proceso de implementación de la solución. Elementos destacables tales como la seguridad, rendimiento, aplicación de patrones, etc.

5.4.1. Seguridad en el sistema

Teniendo en cuenta que con esta propuesta es el dispositivo del usuario quien interactúa con el sistema, hay que considerar ciertos aspectos de seguridad. En este caso se hace especial hincapié en cómo se va a almacenar la información del usuario y en el tráfico de los datos a través de la red de la infraestructura del sistema. Para asegurar la calidad de este aspecto, se plantean las siguientes medidas de seguridad que hay que tener en cuenta para las siguientes fases de la creación del producto software:

- Almacenamiento seguro y cifrado de los datos sensibles
- Medidas de seguridad para la comunicación entre las diferentes capas del sistema en cuanto a confidencialidad, no repudio e integridad.

²⁰ Sitio web oficial de Plesk: <https://www.plesk.com>

²¹ Manual de Webs avanzadas de la UPV:
<https://wiki.upv.es/confluence/display/MANUALES/Servicio+de+webs+avanzadas>

- Verificación y tratamiento de los campos de entrada hacia los diferentes endpoints que tiene la parte servicio del sistema.

La seguridad de la información es de vital importancia en cualquier aplicación móvil. Uno de los datos más sensibles que se maneja en esta aplicación es la contraseña del usuario. Por esta razón, es importante tomar medidas para asegurar que esta información se almacene de manera segura y cifrada en la base de datos. Para lograr este objetivo, se ha utilizado el ya mencionado paquete de *bcryptjs*. Es una biblioteca que implementa el algoritmo de cifrado de contraseñas en JavaScript. Este algoritmo es conocido por ser uno de los más seguros para el almacenamiento de contraseñas [20].

La función hash utilizada por *bcryptjs* para cifrar la contraseña es irreversible, lo que significa que una vez que la contraseña se ha cifrado, es prácticamente imposible de revertir (al menos sin computación cuántica). La razón de esto es que la función hash toma los datos de entrada y los transforma en una cadena de caracteres fija (el hash). Este proceso no es reversible, porque para cualquier conjunto de datos de entrada, habrá un número infinito de resultados posibles de hash, y no hay forma de saber con certeza cuál es el conjunto original de datos de entrada. Además, *bcryptjs* agrega una capa adicional de seguridad mediante la adición de un *salt* aleatorio a la contraseña antes de que se cifre. El *salt* es un valor aleatorio que se agrega a la contraseña para hacer que el proceso de cifrado sea aún más complejo y difícil de descifrar. Esto significa que incluso si un atacante pudiera acceder a la contraseña cifrada, no podrían revertirla sin conocer el valor del *salt*.

Luego, *bcryptjs* aplica una función de hash de clave adaptativa conocida como *Blowfish* [21] para cifrar la cadena resultante. *Blowfish* es un algoritmo de cifrado simétrico que utiliza una clave secreta para cifrar y descifrar los datos. *Blowfish* utiliza una estructura de datos conocida como "caja de sustitución" (*S-box*) para cifrar los datos. La *S-box* es una tabla de 4 KB que contiene una serie de valores pseudoaleatorios que se utilizan para cifrar los datos.

```
const { name, email, password } = req.body;
const user = { name, email, password, role: 0, verified: false };

const salt = await bcrypt.genSalt(10);
user.password = await bcrypt.hash(user.password, salt);
```

Figura 17. Cifrado de valores de entrada del usuario

En la Figura 17 se muestra el sencillo paso de cifrar la contraseña del usuario gracias a esta tecnología. Se genera el mencionado *salt* con diez iteraciones de forma asíncrona. Después se llama a la función de generación del hash con el *salt* y se introduce en los argumentos la contraseña “en crudo” y el *salt* generado.

Para proporcionar diferentes capas de seguridad para el servicio de API RESTful (web), incluyendo protección contra ataques como cross-site scripting (XSS), clickjacking, y protección de cabeceras HTTP [22] se ha utilizado el paquete helmet. Esta herramienta es especialmente importante para proteger la comunicación entre el servicio web (lógica de negocio) y la aplicación para móviles, ya que puede evitar ataques a través de vulnerabilidades en las cabeceras HTTP. Además, permite configurar de manera sencilla el cifrado de las comunicaciones a través de HTTPS.

Por otro lado, para tener mecanismos que permitan de forma segura transmitir la información del usuario entre las diferentes capas del sistema sin tener que preocuparse por la interceptación o manipulación de esta información se hace uso del paquete “jsonwebtoken”. JSON Web Tokens (JWT) es un estándar abierto que define un formato compacto y autónomo para la transmisión de información segura entre diferentes partes, y este paquete proporciona una implementación fácil de usar en Node.JS. Para entender por qué estos paquetes son importantes para la seguridad del proyecto, es importante analizar cada uno de ellos en detalle.

En primer lugar, helmet proporciona una capa adicional de seguridad para la comunicación entre el servidor y el cliente a través de HTTPS. HTTPS es un protocolo de comunicación seguro que utiliza cifrado SSL/TLS para proteger la información que se transmite entre el servidor y el cliente. Cuando un usuario envía información a través de la aplicación, esa información se envía en texto plano, lo que significa que cualquier persona que tenga acceso a la red puede interceptar y leer la información. Sin embargo, se utiliza HTTPS, la información se cifra antes de ser enviada a través de la red, lo que significa que incluso si alguien intercepta la información, no podrá leerla.

En segundo lugar, gracias a jsonwebtoken, cuando un usuario inicia sesión en la aplicación, se crea un *token* (una cadena de caracteres) que sirve para tener la sesión de ese usuario. Contiene información sobre el usuario como su nombre y su ID de sesión. Para acceder a cualquier parte de la aplicación que requiera autenticación, el usuario debe proporcionar esta información de alguna manera. Por ejemplo, el punto de acceso de la API para crear un nuevo proyecto en el sistema requiere como valor de entrada indispensable en la cabecera HTTPS este JSON Web Token que identifique al usuario.

Cuando el usuario inicia sesión este *token* es almacenado en el espacio seguro del dispositivo a través de una API de almacenamiento seguro de claves para dispositivos móviles que utiliza el cifrado de hardware del dispositivo para proteger los datos almacenados. Esta herramienta funciona tanto en el sistema operativo Android como en iOS, pues sendos sistemas implementan esta utilidad ya que es la más extendida.

```
export const verifyToken = async (req, res, next) => {
  try {
    const auth = req.get('authorization');

    if (!auth) { throw {CODE: 401}; }

    const token = auth.substring(7);
    const decoded = jwt.verify(token, config.jwt_secret);

    try {
      const result = await service.getUserByTokenId(decoded.id);

      if (!result) { throw {CODE: 401}; }

      req.uid = decoded.id;
      next();
    }
  }
  . . .
}
```

Figura 18. Servicio de verificación de tokens

Lo que se puede apreciar en la Figura 18 es un fragmento de código que verifica que en la cabecera de la petición HTTP entrante existe el parámetro de autenticación. En caso de que exista extrae el valor y verifica si es un *token* válido. Para llevar a cabo esa tarea primero verifica la firma del *token*, si la firma es correcta (contiene la palabra secreta para el cifrado) comprueba que efectivamente ese usuario es el correspondiente en el registro de la base de datos gracias a una llamada al servicio que cuenta con esta funcionalidad. La sentencia “next” indica que esto es un *middleware*, es decir un servicio que se encuentra en el medio de un proceso y que solo continuará el procesamiento si se cumplen las condiciones mencionadas. En caso contrario se maneja la respuesta de error correspondiente de vuelta.

El tercer pilar del aspecto de seguridad es validar los datos de entrada de una solicitud HTTP antes de que se procese la solicitud. Esta función se puede usar con una librería del framework web popular Express.js. Antes de entrar en detalle se muestra un pequeño fragmento de código (Figura 19) que puede ser muy expresivo para entender el concepto.

```
return async (req, res, next) => {
  await Promise.all(validations.map(validation =>
    validation.run(req)));

  let err = validationResult(req);
  if (!err.isEmpty()) {
    return res.status(400)
      .json({ errors: err.errors });
  }

  next();
};
```

Figura 19. Verificador de campos de entrada

El algoritmo, que de nuevo es un *middleware*, recibe por el enrutador que tipo de petición es y que campos de entrada se esperan, así como las restricciones y comprobaciones que tienen los valores de entrada que provienen de esa ruta. Con los valores de entrada de la petición y sabiendo que parámetros debe comprobar, mapea las restricciones y ejecuta la orden “Promise.all(...)” que lanza un vector de promesas JavaScript. Esto permite ejecutar todas las validaciones de forma asíncrona y en paralelo, mejorando el rendimiento del servicio y el tiempo de procesamiento. Además, al utilizarlo se puede obtener un resultado combinado de todas las validaciones (otra promesa), lo que hace que sea más fácil manejar y procesar los resultados de validación de forma unificada. De hecho, así es como lo hace el servicio, ya que de esa forma puede mandar en el mensaje de respuesta de los campos erróneos en un mismo objeto respuesta a la capa de presentación e indicarlos en el formulario correspondiente (si procede). Si todos los campos de entrada son correctos y válidos, entonces el procesamiento sigue hacia el controlador correspondiente (o el siguiente *middleware*). En caso contrario, se manda el objeto respuesta mencionado con los campos erróneos y que acciones son necesarias para solucionarlo.

5.4.2. Patrones de diseño y refactoring

El desarrollo de software puede ser un proceso complejo, especialmente cuando se trata de una actividad de varios meses. Para manejar esta complejidad, se utilizan patrones de diseño y técnicas de refactorización para mejorar la calidad y la eficiencia del código.

Los patrones de diseño son soluciones probadas y comprobadas para problemas comunes de diseño de software. Estos patrones son útiles porque proporcionan una forma estandarizada de resolver problemas, lo que hace que el código sea más fácil de entender y mantener. Además, los patrones de diseño permiten la reutilización del código, lo que ahorra tiempo y reduce la posibilidad de errores. Suele ir en conjunto con la refactorización, un proceso que consiste en reestructurar el código existente sin cambiar su funcionalidad externa. El objetivo de la refactorización es mejorar la calidad del código, haciéndolo más fácil de entender, mantener y escalar. La refactorización también puede mejorar el rendimiento del código y reducir la complejidad.

En el caso de TeamMaker, el uso de patrones de diseño y la refactorización son importantes por varias razones. En primer lugar, estos enfoques permiten que el código sea más modular y escalable, lo que facilita el mantenimiento y la evolución del proyecto a medida que crece. En segundo lugar, el uso de patrones de diseño y refactorización puede reducir la cantidad de errores en el código, lo que a su vez reduce el tiempo y los costes de corrección de errores. A continuación, se exponen algunos patrones de diseño y medidas de refactoring a nivel de código fuente para explicar la implicación de este tema en el producto final.

Patrón Strategy

La aplicación TeamMaker permite a los usuarios buscar proyectos según diferentes criterios, como palabras, habilidades requeridas, validez para presentación como TFG, etc. En este caso, se propone utilizar el patrón de diseño Strategy [23] para encapsular diferentes algoritmos de búsqueda en espacios separados y permitir que el sistema elija el algoritmo que desea utilizar en tiempo de ejecución.

En el apartado del servicio se encuentran los siguientes algoritmos que todos ellos implementan la acción de buscar. Y lo importante para dar validez al patrón, todos implementan la misma acción, pero para fines distintos dentro del mismo contexto.

```
export const getAcademicProjects = async (req, res);
export const getProjectsByContent = async (req, res);
export const getProjectsByProfiles = async (req, res)
export const getUserProjects = async (req, res)

. . .

router.get('/academic', getAcademicProjects);
router.get('/mine', authJWT.verifyToken, getUserProjects);
router.get('/find/:content', getProjectsByContent);
router.get('/filter', getProjectsByProfiles);

. . .
```

Figura 20. Fragmentos de la estructura del patrón Strategy

La Figura 20 es un fragmento extraído de la parte donde se ha definido el espacio común para todas las estrategias de búsqueda. Al llamar a cada estrategia desde otro punto de la aplicación, se lanza por defecto la función “search()” que en base a la estrategia seleccionada y los parámetros de entrada (si hay) hará un tipo de búsqueda específico aplicando su estrategia.

Esto facilita mucho el mantenimiento por ejemplo en los enrutadores de la búsqueda, ya que de forma muy sencilla puedes indicar la ruta y la estrategia de búsqueda de esa ruta. Ganando una mejora en el entendimiento del código y en desacoplar los mecanismos de búsqueda con las rutas y los controladores.

Patrón Proxy

Como ya se ha explicado, la aplicación permite subir archivos y almacenarlos (imágenes, documentos como el CV, etc). Para subir los archivos, la capa de lógica de negocio utiliza el paquete Multer ²² de Node.JS, que se encarga de manejar la subida de archivos. Sin embargo, se decide agregar una capa de seguridad adicional para evitar que los usuarios suban archivos maliciosos o con un formato que no se espera manejar. En lugar de permitir que los usuarios suban archivos directamente a través de esta herramienta, se incorpora un Proxy [23] que se encarga de verificar cada archivo antes de permitir que se suba al sistema de ficheros. Este Proxy actúa como un intermediario entre el usuario y Multer.

El proxy genera el nuevo nombre del fichero (con el que se almacenará en el servidor) con un sistema de aleatorización criptográfico. Además, si es necesario modifica las imágenes (tamaño, calidad, etc) para optimizar el espacio y los tiempos de carga de este contenido en los dispositivos. Se puede apreciar un breve ejemplo en la Figura 21.

```
const { folder } = req.body;
const imgName = randomatic('Aa0', 32)
  + path.extname(req.file.originalname);

try {
  await sharp(req.file.buffer)
    .resize({ width: 250 })
    .toFile(`./${folder}/${imgName}`);
  . . .
}
```

Figura 21. Fragmento de implementación del Patrón Proxy

El Proxy utiliza la clase Multer original para realizar la subida de archivos, pero reemplaza la función *single* para agregar la funcionalidad adicional. Finalmente llama a

²² Repositorio de Multer: <https://github.com/expressjs/multer>



la función de devolución de llamada o *callback function* y el proceso de subida del archivo sigue su curso o se interrumpe con un error si el fichero de entrada no es válido.

El proyecto incorpora algunos patrones de diseño más, como puede ser el patrón **Singleton** [23] para tener una única instancia de conexión con la base de datos o el patrón **Decorator** [23] para modificar algunos recursos sin alterar la interfaz original de la herramienta, etc.

Refactoring

Se han realizado diversas tareas de refactoring durante el transcurso del proyecto, eso provoca una situación donde se podrían destacar varias actividades realizadas. Sin embargo, por claridad de la definición, se opta por mencionar y explicar uno de ellos que además ha sido recurrente durante el proceso de desarrollo.

El refactoring *Extract Method* es una técnica que consiste en extraer una sección de código de un método existente y convertirla en un nuevo método separado, con el objetivo de reutilizarlo en otros puntos de la aplicación. La idea detrás de este refactoring es hacer el código más modular, más fácil de leer y de mantener. Al extraer una sección de código en un método separado, se pueden evitar repeticiones de código. Además, puede ayudar a reducir la complejidad del código, ya que las funciones que realizan varias tareas se pueden dividir en funciones más pequeñas y especializadas.

```
export const sendMail = async (to, subject, html) => {
  const transporter = nodemailer.createTransport({
    host: config.email_host,
    port: config.email_port,
    . . .
  });

  await transporter.sendMail({
    from: config.email_from,
    to,
    . . .
  });
}
```

Figura 22. Servicio de transporte de correos electrónicos

Un ejemplo trivial y visible es la extracción de código repetitivo que se estaba produciendo en el código fuente de la aplicación para enviar mensajes a través de correo electrónico (ver Figura 22). En su lugar se extrae esta funcionalidad en un método exportable y genérico para cualquier tipo de mensajes. También se incorporó las variables de entorno seguras en este elemento para aumentar incluso más el desacoplamiento y la mantenibilidad en caso de migrar o cambiar los parámetros de configuración del servicio de correo electrónico. Esto se puede por ejemplo cuando se

accede a “config.email_host”. Desde la instancia “config” se accederá de manera segura a la variable de entorno configurada.

5.4.3. Análisis energético o de eficiencia algorítmica

Al querer diseñar un producto software para una plataforma móvil, es inevitable tener en cuenta que como todo dispositivo de esta índole, existen limitaciones en cuanto autonomía y recursos del sistema. Para ello se van a generar medidas que ofrecen un resultado considerable para un amplio espectro de dispositivos.

En el aspecto de eficiencia, se implementan medidas para reducir el uso de la memoria (recurso que es limitado en los sistemas operativos). El punto más crítico de este apartado es el listado de los proyectos en la pantalla principal. Visualizar los proyectos desde esta vista puede suponer la carga de algunos centenares de proyectos en memoria, con su correspondiente contenido multimedia, información, etc. Para solventar este estado, se hace uso de las herramientas extensibles del componente nativo de React llamado FlatList ²³.

Cuando se utiliza FlatList, sólo se renderizan los elementos que se encuentran actualmente en la ventana de visualización del usuario. Esto significa que, si hay una gran cantidad de elementos en la lista, sólo se renderizarán los que el usuario pueda ver en ese momento, en lugar de renderizar toda la lista al mismo tiempo. Esto se traduce en una mejora significativa del rendimiento de la aplicación, ya que se evita el uso excesivo de memoria y reduce el coste computacional. Gracias a esto, el sistema únicamente renderiza entre uno a tres proyectos en la interfaz (dependerá del tamaño de la pantalla del dispositivo y de la extensión de los proyectos mostrados). Se ha configurado la carga perezosa, lo que significa que sólo se cargan los elementos adicionales a medida que el usuario los va desplazando.

Además, si existe algún elemento que ya ha sido renderizado previamente en la pantalla actual se reutiliza evitando la redundancia. Por ejemplo, un botón o una fecha que es idéntica a la siguiente que se tiene que renderizar.

```
<FlatList
  data={projects}
  renderItem={renderItem}
  keyExtractor={keyExtractor}
  maxToRenderPerBatch={5}
  . . .
/>
```

Figura 23. Configuración de una lista plana en React Native

²³ Documentación oficial de FlatList: <https://reactnative.dev/docs/flatlist>



En la Figura 23 se pueden ver algunas propiedades configuradas en la implementación del componente. Por ejemplo, para aumentar en ciertos escenarios la eficiencia de este componente, también se añade un “*Key Extractor*” que ayuda a FlatList a identificar cada componente de la lista (es decir, cada proyecto publicado) y catalogarlo para añadirlo a una memoria cache interna y no tener que solicitar el recurso de nuevo cuando se vuelve a la ventana. Entre las propiedades de configuración de este componente también se marca el objeto a renderizar para cada elemento de la lista y otros aspectos como el número máximo de elementos a renderizar por bloque de carga.

En cuanto al coste computacional, gracias a la eficiencia en el uso de memoria de la aplicación, determinadas operaciones computacionales que pueden ser costosas y que son repetitivas a lo largo del uso de la aplicación, se pueden memorizar (guardar en memoria) para realizar el cómputo una única vez y de forma asíncrona a la carga de la aplicación. Aquí es donde entra de nuevo el concepto de memoización en React Native. La idea básica es almacenar en caché los resultados de una función y devolver el resultado almacenado cuando se llama a la misma función con los mismos argumentos. Si alguno de los argumentos cambia, entonces hay que recalcularse la operación.

Esta optimización se ha utilizado en algunos puntos de la aplicación cuyo peso computacional es considerable. Hay que tener en cuenta que se está almacenando recursos en la memoria del dispositivo, por lo que hay que ser cuidadoso y usar esta utilidad cuando de verdad aporte una mejora en rendimiento.

Los puntos de la aplicación donde se ha utilizado la técnica son los métodos de inicio de sesión y comprobación de *token* de sesión y la función que obtiene los datos de los proyectos en la carga perezosa del componente FlatList mencionado anteriormente.

Solicitar el recurso y la comprobación del *token* de sesión en cada ventana de la aplicación puede resultar ineficiente y redundante computacionalmente. Además, puede empeorar la experiencia de usuario pues el tiempo de respuesta y la reacción de la interfaz puede ser algo menos dinámica. Es por ello que cuando se entra en la aplicación se realiza este cálculo una única vez y ya se cuenta con ese resultado para enviarlo junto con las cabeceras en cada petición que se realice. Un fragmento del funcionamiento se puede apreciar en la Figura 24:

```
const authContext = React.useMemo(() => ({
  signIn: async (data) => {
    try {
      const api_call = await fetch(`${config.API}/auth/login`, {
        method: 'POST',
        headers: {
          Accept: 'application/json',
          . . .
        }
      })
    }
  }
})
```

Figura 24. Técnica de memoización en React Native

En cuanto a la lista de proyectos, como se ha expuesto anteriormente el componente que renderiza los elementos solicita los recursos de forma perezosa. Cuando el usuario desliza hacia abajo sobre la lista y llega al final del bloque de proyectos cargado, la lista solicita el siguiente bloque de proyectos para renderizarlos. La relación que tiene esto con la técnica de memoización es que a medida que ese bloque que se ha renderizado y que se está viendo en pantalla, se guarda en memoria cache. De tal forma que, si el usuario abandona esa vista y luego vuelve, no es necesario solicitar el recurso de nuevo. Este procedimiento se realiza de forma responsable y los bloques de proyectos no superan una extensión de cinco unidades almacenadas.

5.5. Pruebas

Las pruebas son una parte fundamental del proceso de desarrollo de software, ya que permiten garantizar la calidad y la fiabilidad de las aplicaciones. En este sentido, existen diferentes tipos de pruebas que se pueden aplicar durante el ciclo de vida del software para evaluar distintos aspectos de su funcionamiento. En el proyecto se han llevado a cabo los siguientes tipos de pruebas: unitarias, integración, rendimiento. Todas ellas han sido automatizadas a través de diversas herramientas.

Las pruebas unitarias son aquellas que se enfocan en evaluar el comportamiento de una unidad aislada de código, como una función o un método. Estas pruebas permitieron detectar errores y comportamientos inesperados en el código sobre todo en los controladores y los servicios en la lógica de negocio. Esto a su vez facilitó su corrección antes de integrar con otras partes del sistema.

Para llevar a cabo estas pruebas se instaló la herramienta Jasmine ²⁴ como paquete de desarrollo. Es decir, que en producción no será instalado cuando se ejecute el comando de instalación de dependencias a través del gestor de paquetes de Node.JS. Jasmine es un marco de pruebas de software utilizado para realizar pruebas de integración en aplicaciones. En términos simples, Jasmine es una herramienta que permite a los desarrolladores escribir y ejecutar pruebas automatizadas para asegurarse de que los distintos componentes de su aplicación funcionen correctamente juntos.

²⁴ Sitio web oficial de Jasmine: <https://jasmine.github.io>



```
it(_, async () => {
  const req = { . . . };
  const res = {
    status: jasmine.createSpy().and.returnValue({ . . . }),
    json: jasmine.createSpy(),
  };
  const next = jasmine.createSpy();

  await changePassword(req, res, next);

  expect(res.status).toHaveBeenCalledWith(200);
  expect(res.json).toHaveBeenCalledWith({ changed: true });
  expect(next).not.toHaveBeenCalled();
});
```

Figura 25. Estructura de código de una prueba unitaria

Para comprobar muchos de los parámetros con los que se trabaja en los controladores y servicios de la lógica de negocio se ha hecho uso de la utilidad de Jasmine llamada *Spy*. Esta genera una función simulada que se utiliza para hacer un seguimiento de cómo se comporta otra función en el código. En el caso de la Figura 25, se analiza el comportamiento de la función “changePassword” que se encarga de modificar la contraseña de un usuario con la premisa de que tiene una sesión activa y válida.

Las pruebas de integración se han enfocado desde el punto de vista de la comunicación entre la capa de presentación y lógica de negocio. El foco está en evaluar el comportamiento de diferentes unidades de código cuando se combinan y se comunican entre sí. Para que esta tarea sea posible hace falta algunas herramientas de desarrollo adicionales. En este proyecto se han utilizado (además de Jasmine) el conjunto tecnológico para pruebas para Node.JS formado por SuperTest ²⁵, Mocha ²⁶ y Chai ²⁷. En este escenario son necesarias pues hay que tratar como un conjunto a toda la lógica de negocio. Se trata de simular las mismas peticiones que se realizan desde la capa de presentación (la aplicación móvil) para probar que las respuestas que se generan son las adecuadas y no se produce ningún error inesperado. De esta forma se está realizando pruebas de una capa hacia otra.

²⁵ Repositorio oficial de SuperTest: <https://github.com/ladjs/supertest>

²⁶ Sitio web oficial de Mocha: <https://mochajs.org>

²⁷ Sitio web oficial de Chai: <https://www.chaijs.com>

```

it(_, () => {
  supertest(app)
    .post('/users/changePasswd')
    .expect(401)
    .send({ . . . })
    .end((err, res) => { . . . });
});

```

Figura 26. Estructura base de una prueba de integración

En la Figura 26 se puede ver como con la herramienta SuperTest se introduce como argumento “app” que es una instancia de toda la lógica de negocio. Es decir, como si fuera el servicio desplegado a modo de API RESTful en la web. A continuación se simula una petición HTTP a una ruta específica (la que se quiere probar) y la respuesta que debe generarse para que el caso de prueba sea correcto y pase la prueba.

Para facilitar la realización de las pruebas en el transcurso del desarrollo de la aplicación, se incorpora en la sección de *scripts* de la lógica de negocio la instrucción que permite con un comando en la terminal lanzar todas las pruebas. Esta instrucción se ubica en el paquete “package.json” que es el fichero de configuración de esta capa.

Fueron considerables las iteraciones que se produjeron realizando el lanzamiento del comando que dispara la ejecución de todos los casos de prueba (unitarias y de integración) debido a la detección de fallos alrededor del código fuente y la realización de cambios oportunos para solucionar los errores. En ocasiones el error era fácil de encontrar, por ejemplo en las pruebas unitarias, alguna sentencia SQL mal planteada. Otros defectos tuvieron una complejidad añadida pues el defecto se producía tras una cadena de sucesos entre módulos de la lógica de negocio. Finalmente, 126 casos de prueba fueron ejecutados correctamente, la salida de la herramienta se puede ver en la Figura 27 (resultado por consola):

```

> api@1.2.1 test
> jasmine

Randomized with seed 81167
Started

126 specs, 0 failures

```

Figura 27. Resultado de las pruebas en la lógica de negocio



Para medir el rendimiento de la aplicación se ha utilizado una de las funcionalidades que tiene la herramienta Flipper Performance Monitor ²⁸. Esta mide el rendimiento de la aplicación en React Native en tiempo real. El proyecto al tratarse de una aplicación bajo la capa de Expo, también es necesaria la instalación del paquete “expo-community-flipper” para que la ejecución en tiempo real sea compatible. La prueba dura diez segundos y mide los fotogramas por segundo de la interfaz gráfica. El último resultado arrojó una puntuación de 90% y un promedio de 59.3 fotogramas por segundo. La figura muestra la captura de los fotogramas por segundo durante la prueba de diez segundos, una captura por cada segundo transcurrido.

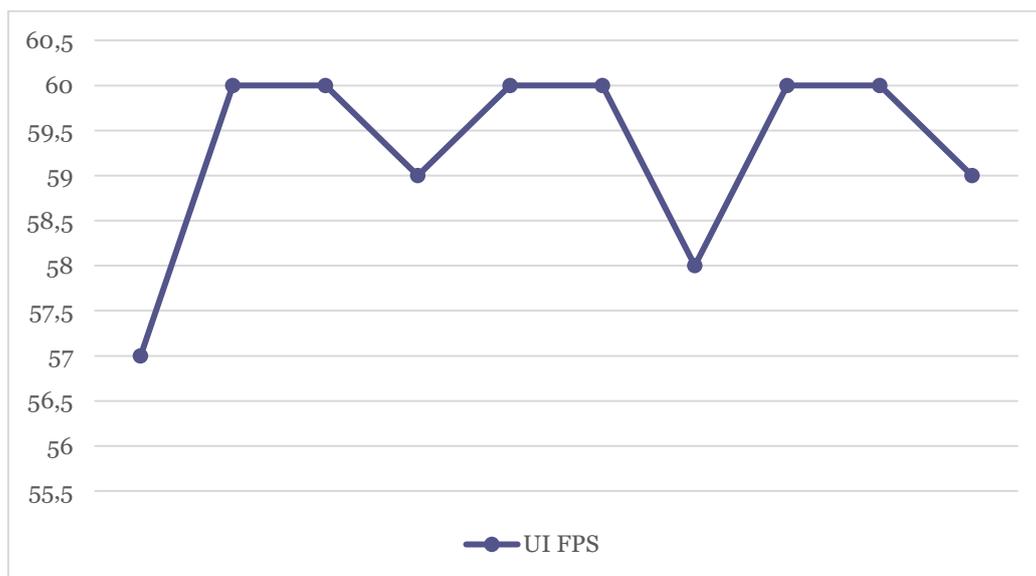


Figura 28. Resultados de la prueba de rendimiento de la IGU

La gráfica de la Figura 28 muestra los fotogramas que se han registrado de la interfaz gráfica por cada segundo. Es decir, en el primer segundo se dispararon 57 fotogramas, en el tercer segundo 60, etc. Hay diez registros que corresponden a los diez segundos de duración que tuvo la prueba.

El resultado global de la herramienta es el que se muestra en la Figura 29:

Performance



Figura 29. Resultado de la prueba Flipper Performance Monitor

²⁸ Repositorio de Flipper Performance Monitor:
<https://github.com/bamlab/react-native-flipper-performance-monitor>

5.6. Metodología y cronología

Para la organización y seguimiento del trabajo se ha utilizado la herramienta Worki ²⁹, una utilidad de soporte para la implementación del enfoque ágil basado en el framework TUNE-UP Process [24]. Éste integra las prácticas ágiles de los métodos ágiles más populares. En específico, se utilizan procesos de Scrum, Kanban, Lean Development y Extreme Programming (XP).

5.6.1. Línea de trabajo

La línea de trabajo del proyecto está formada por:

- Product Owner (el tutor de la práctica): Entidad encargada de gestionar el contenido del proyecto y la prioridad de cada unidad de trabajo.
- Backlog: Contenedor de unidades de trabajo que se organizan en base a la prioridad de incorporación al producto final.
- Desarrollador del producto software

Se puede generar diferentes líneas de trabajo para cada aspecto del proyecto, pero en este caso el proyecto consta de una sola línea de trabajo con el conjunto de unidades en un *backlog*. En este contexto basta con este planteamiento debido a que tanto la parte “cliente” como la parte “equipo” es unipersonal.

5.6.2. Estructura y flujo de trabajo

El trabajo se ha almacenado en el contenedor *backlog* (o pila de trabajo) y se ha mantenido en todo momento un orden en base a la prioridad para asegurar que siempre se está desarrollando el trabajo definido como más prioritario. Esta organización ha sido útil para identificar algún bloqueo al visualizar que una unidad de trabajo más prioritaria estaba pausada y se había dado el turno a una con prioridad inferior. Además la estructura del *backlog* ha ido cambiando pues a medida que evolucionaba el producto software se iba añadiendo o retirando trabajo. El flujo de trabajo se define por un tablero kanban que establece la consecución de actividades que se realizan por cada unidad de trabajo. La definición del flujo de trabajo se puede descubrir en la Figura 30.

²⁹ Sitio web de la herramienta: <https://cliente.tuneupprocess.com/web>

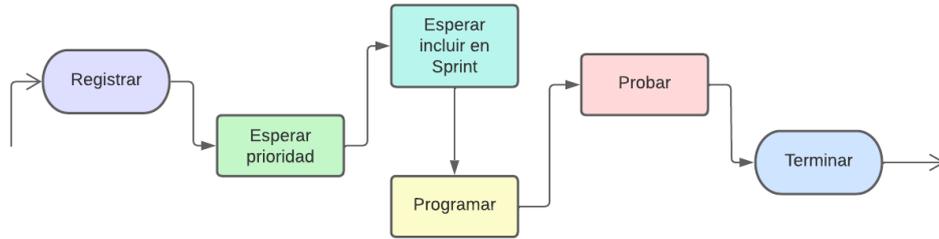


Figura 30. Workflow de unidad de trabajo

5.6.3. Planificación y seguimiento del trabajo

La planificación del trabajo se ha realizado mediante Sprints (bloques temporales de tres semanas) donde se han incorporado las unidades de trabajo en base al alcance y la envergadura de cada unidad. Todos los bloques han tenido la misma duración en semanas y se han registrado consecutivamente. Para el registro, se organiza una reunión con el Product Owner una semana antes de la finalización de cada Sprint, se conversa del estado del proyecto y si es necesario registrar algún fleco de trabajo. Si procede algún cambio, se realiza sobre el backlog. Finalmente se mueven las unidades de trabajo más prioritarias y en función del alcance al Sprint próximo (una semana vista). Este concepto se puede visualizar de forma gráfica en la Figura 31.

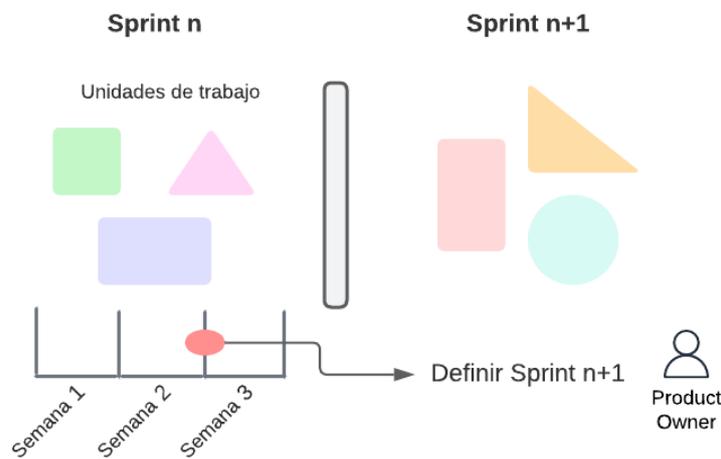


Figura 31. Planificación y seguimiento del trabajo

Esta planificación fomenta la entrega continua de nuevas funcionalidades que la parte cliente puede ver y manejar. El resultado es un producto que puede presentar cambios con mayor frecuencia pero que estos cambios son más soportables en cuanto a su coste debido a la alta frecuencia y a detectarlos en fases tempranas.

Por tanto, se utiliza la modalidad flexible de la definición de los bloques temporales, pues en este caso es más conveniente porque es una aproximación más realista de como el

trabajo puede ir sufriendo modificaciones en este contexto. Pues se trata de un producto software de nueva incorporación para la parte cliente. Es decir, tal como se aprecia en la Figura 32, se pueden definir algunas pequeñas modificaciones sobre el trabajo realizado y cambiar la prioridad de unidades de trabajo, así como añadir o retirar del backlog.

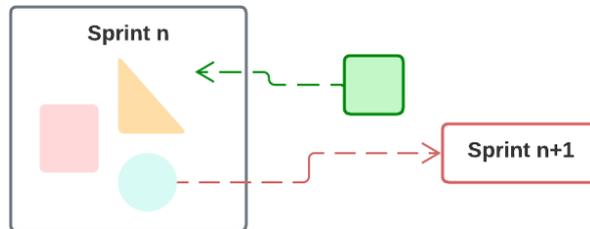


Figura 32. Modificaciones en un Sprint

5.6.4. Plan de trabajo

La estimación del esfuerzo para proyectar el alcance del proyecto se realiza mediante Horas Ideales de Programación (HIP), es decir, el tiempo que se emplearía en terminar una unidad de trabajo si se realiza la actividad de programar ininterrumpidamente. Fragmentando estas horas de programación a una temporalidad de tres semanas. Debido a la naturaleza del proyecto, que es realizado sobre una práctica de empresa, el contrato estipula quince horas semanales. No obstante, además del proyecto que da nombre a este documento han existido otras tareas que conciernen a la práctica. Por este motivo se estima un 75% del esfuerzo contratado para el proyecto. Resultando en la siguiente cifra de HIP (Horas Ideales de Programación) para un esfuerzo (E) contratado de quince horas por semana y tres semanas por cada Sprint:

$$HIP = 3 * 0,75 * E \cong 34 h$$

Gracias al tablero de la herramienta Worki, se puede tener un backlog ordenado por prioridad y definir la estimación del esfuerzo en HIP de cada unidad de trabajo. Tal y como se concreta en la planificación y seguimiento del trabajo, aproximadamente una semana antes de finalizar el Sprint actual, se cambian las unidades de trabajo más prioritarias al estado “Esperar incluir en Sprint” hasta llegar a las 34 HIP de alcance para esas tres semanas próximas. Una vez cambiado a ese estado, se agenda una reunión con el Product Owner para actualizar el estado del proyecto, ver si resulta en la aparición de un fleco de trabajo, alguna reestimación o confirmar la elección de las unidades de trabajo para el siguiente Sprint. En este momento esas unidades de trabajo pasarían al próximo Sprint en estado “Programar”.

Siguiendo con el desarrollo ágil del producto software y teniendo una aproximación de los requisitos del sistema, hay que poblar el backlog de unidades de trabajo y agruparlas

en nodos. Es importante agrupar estas unidades de trabajo en nodos por su temática por varias razones:

- **Mejora la visibilidad:** Se puede ver fácilmente qué áreas del proyecto están avanzando más rápido y cuáles necesitan más atención. Esto mejora la visibilidad y ayuda a priorizar las tareas en función de las necesidades del proyecto.
- **Facilita la planificación:** Permite planificar más fácilmente el trabajo que se llevará a cabo en cada iteración. Teniendo una mejor comprensión del trabajo que se realizará en el futuro, esto permite planificar de manera más efectiva el tiempo y los recursos necesarios para cada tarea.
- **Reduce la complejidad:** Se puede centrar el foco en tareas específicas relacionadas entre sí, lo que permite enfocar la atención y conocimientos en una tarea específica en lugar de tener que cambiar constantemente de tarea y cambiar de contexto.

A continuación se definen los nodos de trabajo y su correspondencia en unidades de trabajo que finalmente se han insertado. Formando en su conjunto el Backlog o pila de trabajo inicial en el proyecto:

Nodo	Unidades de trabajo (UTs)
Listado de proyectos	+ Consumir y mapear listado desde la API + Estructurar los datos en los componentes de IGU + Optimizar el uso de memoria al deslizar sobre los elementos de la lista
Ficha de proyecto	+ Mapear la información de las propiedades recibidas desde el componente de navegación + Cargar contenido multimedia + Flujo de estados y transiciones en los botones de inscripción a los perfiles
Gestión de perfiles	+ Añadir un perfil al listado de perfiles + Modificar un perfil + Insertar perfiles a un proyecto desde el formulario de creación de proyectos + Definir una descripción para cada perfil asociado a un proyecto en concreto
Navegación	+ Navegador de tipo 'Drawer' + Navegadores de tipo 'Stack' para los gestores de proyectos y de inscripciones + Flujo de inserción de navegador con la sesión de usuario (apilar las ventanas que procedan) + Propiedades y mantener el contexto entre ventanas
Gestión de proyecto	+ Formulario de creación de proyecto + Gestionar la subida de contenido multimedia + Modificar el contenido del proyecto + Validación del proyecto por un administrador

Registro y autenticación	<ul style="list-style-type: none"> + Iniciar sesión + Registro de nuevo usuario + Verificar la dirección de correo electrónico + Comprobar si existe un token válido guardado + Recuperar la contraseña
Buscador y filtros	<ul style="list-style-type: none"> + Buscador por texto en título y descripción + Optimización de la indexación de la BD + Filtrado por perfiles + Filtrado por validez para TFG/TFM + Combinación y comportamiento entre filtros activos además del buscador por texto
Solicitudes	<ul style="list-style-type: none"> + Solicitar un perfil a un proyecto + Visualizar listado de candidatos + Abrir documento presentado en la solicitud + Aceptar/Rechazar un candidato
Notificaciones	<ul style="list-style-type: none"> + Notificar de cambio en el estado de la solicitud de algún perfil tanto para el postulante como creador + Notificaciones para la autenticación (código de verificación o de restablecimiento de contraseña)
Perfil del usuario	<ul style="list-style-type: none"> + Vista de perfil de usuario (componentes) + Modificar datos de la cuenta + Cambiar imagen de perfil + Gestionar documento + Cerrar la sesión activa y salir

5.6.5. Cronología del trabajo

El planteamiento de la metodología y la creación del backlog inicial apunta a la cronología del proyecto. Es decir, de qué manera se han llevado a cabo las diferentes unidades de trabajo a lo largo del tiempo. La Figura 33 ilustra las referencias temporales más relevantes a nivel general en la cronología del proyecto. En ella se puede ver la distribución de los sprints, la entrega de las diferentes iteraciones del producto, etc.

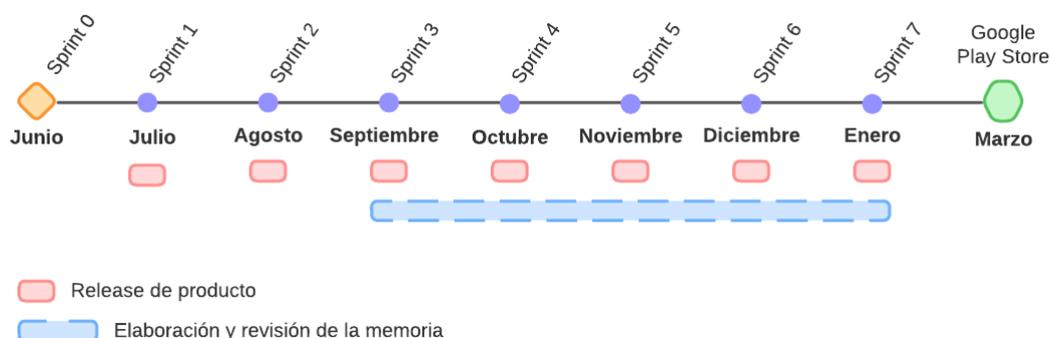


Figura 33. Cronología del proyecto

El proyecto comenzó en junio de 2022 y se dividió en sprints de aproximadamente 34 horas cada uno. El primer sprint (Sprint 0) se llevó a cabo en junio de 2022 y el resto fueron seguidos de manera consecutiva repartidos aproximadamente en cada mes del año. Durante cada finalización del sprint, se realizó una release de MVP de la app, es decir, se lanzó una nueva versión de la app móvil con nuevas funcionalidades incorporadas (proceso iterativo e incremental del producto).

A partir de septiembre de 2022, se comenzó la elaboración y revisión de la memoria del proyecto, lo que llevó alrededor de ochenta horas. Además, se dedicaron alrededor de 40 horas a reuniones y al proceso de definición de los requisitos. Finalmente, en marzo de 2023 se realiza la release de la app en Google Play store, después de haber pasado por varias iteraciones y revisiones durante los sprints previos.

Teniendo en cuenta ocho sprints de unas treinta y cuatro horas cada uno, ochenta horas dedicadas a la memoria y el tiempo para reuniones y especificación, en el proyecto se han empleado alrededor de casi cuatrocientas horas.

Sprint 0 (13/06/2022 – 01/07/2022)

En el primer sprint se centra en la selección de tecnología y contexto del proyecto, así como en la creación básica del mismo y la lista de proyectos activos. Este sprint fue muy importante ya que sienta las bases del proyecto y define su alcance. Algunas de las tareas incluidas fueron la creación del esquema de base de datos, la definición de las tecnologías a utilizar y la creación del repositorio de código.

Sprint 1 (01/07/2022 – 22/07/2022)

En el segundo sprint se creó la ficha de proyecto en la lista de proyectos y se añadió la gestión de perfiles y el alta de proyecto. Uno de los retos principales fue diseñar una ficha de proyecto atractiva e informativa, así como permitir la alta de proyectos con una buena usabilidad. También se trabajó en la navegación desde la lista a la ficha detallada de proyecto al pulsar, para facilitar el acceso a la información.

Sprint 2 (25/07/2022 – 22/08/2022)

En el tercer sprint se trabajó en el registro y autenticación de usuarios, los roles de usuario y el buscador de proyectos. Se implementó un sistema de registro y autenticación robusto y seguro, que permitió a los usuarios acceder a su perfil y las funcionalidades del sistema. También se implementó un buscador de proyectos que permitía a los usuarios encontrar proyectos relevantes para ellos (búsqueda textual).

Sprint 3 (29/08/2022 – 12/09/2022)

En el cuarto sprint se añadió la opción de postularse a un proyecto, se mejoró la ficha de proyecto, se reorganizaron los menús y se añadió el código de proyecto. Uno de los

mayores objetivos fue permitir a los usuarios postularse a proyectos de forma sencilla y transparente, asegurando que el proceso fuera lo más intuitivo posible. También se mejoró la ficha de proyecto para incluir más información y se reorganizaron los menús para mejorar la navegación.

Sprint 4 (12/09/2022 – 03/10/2022)

En el quinto sprint se trabajó en la gestión de postulaciones de un proyecto, la opción de búsqueda de proyectos y la gestión de imágenes en la ficha de proyecto. Se añadió la opción de gestionar postulaciones, permitiendo a los usuarios aceptar o rechazar candidatos. También se mejoró la experiencia de búsqueda de proyectos, permitiendo a los usuarios filtrar por diferentes criterios y se añadió la gestión de imágenes en la ficha de proyecto para mejorar la presentación de este.

Retrospectiva

Durante estos cinco sprints surgieron algunos problemas e inconvenientes, como la necesidad de redefinir ciertos requisitos, la aparición de bugs en el código y la necesidad de hacer ajustes, mejoras en el diseño, tareas de refactoring y cambios para incrementar la usabilidad. Estos se incluyeron como parte de trabajo del siguiente Sprint 5.

Sprint 5 (14/11/2022 – 05/12/2022)

En este sprint trabajó en varias tareas relacionadas con la mejora de la interfaz de usuario. Se recogieron varios errores en la navegación y se mejoró la usabilidad de la aplicación. Además, se trabajó en la implementación del sistema de alertas para informar a los usuarios sobre nuevos mensajes y notificaciones.

Sprint 6 (05/12/2022 – 26/12/2022)

El sprint comenzó con la integración de varias APIs para permitir a los usuarios acceder a información adicional. Se mejoró la búsqueda de la aplicación para incluir resultados de las API y se trabajó en la implementación de un sistema de recomendaciones personalizadas. También se puso el foco en mejoras de la seguridad de la aplicación y se ha corrigió un problema con la pérdida de datos.

Sprint 7 (26/12/2022 – 23/01/2022)

El posible último sprint (hasta una futura continuación del proyecto) se afrontó el reto de aplicar varias mejoras en la funcionalidad de la aplicación, incluyendo la capacidad de personalizar el perfil de usuario y la implementación de un sistema de gestión de archivos. Además, se trabajaron mejoras en la accesibilidad de la aplicación y en la corrección de errores y problemas.



6. Conclusiones y trabajo futuro

TeamMaker es un producto que surgió a raíz de una reunión con el tutor que da nombre a lo que es hoy en día este TFG. Un entrelazado de ideas bien planteadas y con buenos argumentos. En ese momento nació todo un proceso de meses de trabajo transversal, desde la definición semi formal de los requisitos hasta un producto desplegado. Cabe mencionar que los requisitos no han sido fijos durante todo el proyecto (también debido al enfoque ágil). Todos los objetivos propuestos en las fases iniciales del diseño e implementación del producto se han ido cumpliendo. Algunos sprints fueron más complicados que otros. Sin embargo, hay un objetivo que por cuestiones de limitación del alcance no se ha podido cumplir. Ese objetivo es el de tener un sistema de notificaciones completo. La aplicación cuenta con un robusto sistema de notificaciones por email, pero no se ha logrado finalizar todos los casos de uso donde además hay implicada una notificación *push* (notificación del sistema operativo).

La aplicación ha demostrado ser una solución atractiva y necesaria en el ámbito académico, ya que en las fases previas al lanzamiento con el producto mínimo viable ya se planteaba la intención de permitir que alumnos de la asignatura PIN de la escuela pudiesen publicar las ideas de proyectos para buscar compañeros que quisieran desarrollarla conjuntamente. La formación de equipos multidisciplinares es fundamental en proyectos innovadores y la falta de mecanismos y herramientas que faciliten la comunicación entre estudiantes de diferentes titulaciones y escuelas es una necesidad actual en la mayoría de las universidades. La aplicación TeamMaker ofrece una solución práctica y efectiva para este problema.

El siguiente paso (que ya se ha puesto en marcha) es la publicación de la aplicación en la plataforma Google Play Store la primer puesta en escena para el gran público de la comunidad universitaria. No obstante, la idea es que esta plataforma pueda ser migrada a otras universidades que deseen contar con esta herramienta. Se ha diseñado para que la adopción sea un proceso lo más fácil posible. Pues la aplicación se encuentra al alcance de el gran público y tan solo con unos ajustes de unos minutos se puede aceptar a otras organizaciones dentro de la comunidad de TeamMaker.

Durante el proceso he ganado muchas habilidades a nivel académico y profesional. Comenzando por adquirir la capacidad de saber realizar un estudio de las tecnologías del mercado y ser capaz de determinar cuáles son las mejores opciones hasta aprender casi todas las tecnologías que engloban TeamMaker desde cero. Partir de un lenguaje de programación (JavaScript) y unos conocimientos en arquitecturas para la red obtenidos en la asignatura Tecnología de sistemas de información en la red que se imparte durante un semestre en la universidad y trasladar esa base hacia ser capaz de producir en un entorno real un producto software viable, robusto y funcional. Siguiendo con este apartado, las asignaturas Diseño de Software y Proceso de Software han sido de gran utilidad pues mucho del conocimiento adquirido ha sido fácilmente extrapolable al proyecto. Sobre todo la parte metodológica en el trabajo y la de cuidar el aspecto técnico del producto para que sea de calidad. Otra asignatura que ha sido de gran ayuda para la especificación y documentación del proyecto es Análisis y especificación de requisitos.

Gracias a las aportaciones docentes de esta materia he podido aprender la implicación que tiene el proceso del análisis y la especificación en un producto software.

Durante el desarrollo del proyecto, se han identificado algunas áreas de mejora que podrían ser abordadas en futuras actualizaciones de la aplicación. Por ejemplo, se podría incluir una funcionalidad que permita a los usuarios calificar la colaboración de los demás miembros del equipo al finalizar el proyecto. Esto podría ayudar a futuros usuarios a seleccionar a los miembros del equipo adecuados para sus proyectos y mejorar la calidad de las colaboraciones. Otra posible mejora sería la inclusión de una función de chat en la aplicación para que los miembros del equipo puedan comunicarse entre sí. Esto podría mejorar la comunicación y la colaboración en los equipos y hacer que la aplicación sea aún más útil para los usuarios.

Mencionar que el soporte de este proyecto no termina aquí, la plataforma va a seguir siendo mantenida y se van a seguir añadiendo mejoras y nuevas funcionalidades a medida que se analice la respuesta de los usuarios frente a la plataforma. La idea es que otros estudiantes de la escuela sigan con el proyecto y se vayan generando nuevas iteraciones del producto, para adaptarse a las necesidades del momento y para evolucionar el servicio hacia la comunidad universitaria. Esto es la primera fase del producto final, el desarrollo, el despliegue y la puesta en marcha con la comunidad. Pero se espera que haya otros trabajos de fin de grado asociados a TeamMaker y que la aplicación perdure en el tiempo.



Referencias

- [1] Bartle, E., & Sabatier, V. (2020). The impact of specialisation in higher education on workforce participation: A systematic review. *Studies in Higher Education*, 45(7), 1409-1422.
- [2] OECD. (2018). *The future of education and skills: Education 2030*. OECD Publishing.
- [3] Graça, A. M., Alves, H., & Raposo, M. L. (2017). Multidisciplinarity and innovation in engineering education. *Procedia Manufacturing*, 13, 1361-1368.
- [4] Gómez-Galán, J., Sánchez-García, M., Fernández-Cardador, A., & García-Peñalvo, F. J. (2017). Requirements for collaborative learning using mobile devices in higher education. *Journal of Universal Computer Science*, 23(10), 973-992.
- [5] Chang, H. J., & Lee, M. H. (2017). The emergence of entrepreneurship education: Development, trends, and challenges. *International Journal of Innovation and Technology Management*, 14(03), 1-28.
- [6] World Economic Forum. (2020). *The Future of Jobs Report 2020*. Geneva, Switzerland: World Economic Forum. Recuperado de https://www3.weforum.org/docs/WEF_Future_of_Jobs_2020.pdf [Accedido el 4 Mar. 2023].
- [7] Edmondson, A. C., & Harvey, J. F. (2018). Cross-functional teaming for innovation: Integrating research on teams and knowledge in organizations. *Annual Review of Organizational Psychology and Organizational Behavior*, 5, 235-263. doi: 10.1146/annurev-orgpsych-032117-104640.
- [8] McKinsey & Company. (2018). "High-performing teams: A timeless leadership topic". Recuperado de <https://www.mckinsey.com/business-functions/organization/our-insights/high-performing-teams-a-timeless-leadership-topic> [Accedido el 4 Mar. 2023].
- [9] ISO/TR 56004:2019, "Innovation management - Innovation management assessment - Guidance". Chapter 5.2.2.
- [10] MeetUp. Take a look. Recuperado de <https://www.meetup.com> [Accedido el 12 Mar. 2023]
- [11] Steve "ardalis" Smith. (2022). "Architecting Modern Web Applications with ASP.NET Core and Microsoft Azure". Recuperado de <https://dotnet.microsoft.com/en-us/download/e-book/aspnet/pdf> [Accedido el 9 Mar. 2023].
- [12] Hellman, Erik. *Android App Development with Kotlin: Building Modern Apps for the Google Play Store*. Sebastopol, CA, EE. UU.: O'Reilly Media, 2019. 1ra edición. ISBN 978-1-492-07777-6.

- [13] Song, Y., Lu, J., Liu, Z., & Xie, H. (2016). Understanding mobile app stickiness: Measurement and determinants. *Journal of Systems and Software*, 122, 98-109. doi:10.1016/j.jss.2016.09.009.
- [14] ComScore. (2017). 2017 US Cross-Platform Future in Focus. ComScore website. Recuperado de: <https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/2017-US-Cross-Platform-Future-in-Focus> [Accedido el 9 Mar. 2023].
- [15] DuBois, P. (2005). *MySQL*, 3rd Edition. Sebastopol, CA: O'Reilly Media, Inc., 2-3.
- [16] HTTP/2. (2015). Hypertext Transfer Protocol Version 2 (HTTP/2). Recuperado de: <https://www.rfc-editor.org/rfc/rfc7540.txt> [Accedido el 9 Mar. 2023].
- [17] React. (s.f.). Extracting State Logic into a Reducer. Recuperado de: <https://beta.reactjs.org/learn/extracting-state-logic-into-a-reducer.html> [Accedido el 9 Mar. 2023].
- [18] Tejera Aguirre, N. (s.f.). Design Principles of React. Harvard University. Recuperado de: <https://pll.harvard.edu/course/design-principles-react?delta=0> [Accedido el 9 Mar. 2023].
- [19] React. (s.f.). useContext. Recuperado de: <https://beta.reactjs.org/reference/react/useContext.html> [Accedido el 9 Mar. 2023].
- [20] Sewe, A., Bavota, G., Tufano, M., & Di Penta, M. (2020). How Do Developers Follow Security-Relevant Best Practices When Using NPM Packages? *IEEE Transactions on Software Engineering*, 46(5), 551-570. DOI: 10.1109/TSE.2018.2876097.
- [21] Ms NehaKhatr Valmik, Prof. V. K Kshirsagar (2014). Blowfish Algorithm. *IOSR Journal of Computer Engineering (IOSR-JCE)*. e-ISSN: 2278-0661, Volume 16, Issue 2, Ver. X (Mar-Apr. 2014), PP 80-83
- [22] Kaur, R. (2020). Securing Web Applications with Secure Coding Practices and Integrity Verification. *International Journal of Scientific & Technology Research*, 9(5), 656-660.
- [23] Freeman, E., Robson, E., Bates, B., & Sierra, K. (2021). *Head First Design Patterns: A Brain-Friendly Guide*. O'Reilly Media, Inc.
- [24] Cohn, Mike. *Estimación y planificación ágil*. Primera edición, Pearson Educación, 2006.



Apéndice

A. Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.	X			
ODS 5. Igualdad de género.			X	
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.		X		
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.			X	
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

TeamMaker tiene una relación importante con el Objetivo de Desarrollo Sostenible número 4, "Educación de calidad". La aplicación busca fomentar el trabajo en equipo y la formación multidisciplinar de los estudiantes, lo cual es fundamental para su formación y éxito en el ámbito laboral. Además, la aplicación promueve la colaboración entre estudiantes y profesores, lo cual puede generar un mayor aprendizaje y una mayor calidad en los proyectos desarrollados. En concreto, fijándose en las metas del objetivo, la meta 4.4 pues se centra en la mejora de competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento.

En cuanto al Objetivo de Desarrollo Sostenible número 5, "Igualdad de género", esta ligeramente ligada con el proyecto, ya que la aplicación no hace referencia directa a la equidad de género. Sin embargo, es importante mencionar que la aplicación puede ser una herramienta útil para fomentar la participación de mujeres en proyectos multidisciplinares, lo cual puede contribuir indirectamente a la igualdad de género. En específico la meta 5b busca mediante las tecnologías de la información y las comunicaciones, favorecer el empoderamiento de las mujeres. La aplicación TeamMaker ofrece justo este tipo de facilidades.

La relación con el Objetivo de Desarrollo Sostenible número 8, "Trabajo decente y crecimiento económico", es media, ya que la aplicación puede ayudar a los estudiantes y profesionales a desarrollar habilidades y experiencia en trabajo en equipo, lo cual puede mejorar sus oportunidades laborales y contribuir al crecimiento económico.

Finalmente, la relación con el Objetivo de Desarrollo Sostenible número 10, "Reducción de las desigualdades", es baja, ya que la aplicación no hace referencia directa a este objetivo. Sin embargo, es importante mencionar que la formación multidisciplinar y el trabajo en equipo pueden contribuir a la inclusión social y la reducción de las desigualdades al permitir que personas de diferentes perfiles y orígenes trabajen juntas hacia un objetivo común.

B. Manual de usuario

Una vez descargada e instalada la aplicación de la tienda o repositorio correspondiente, la pantalla inicial es la de inicio de sesión. En esta pantalla el usuario tiene que introducir sus credenciales o presionar el botón “Regístrate” para crearse una cuenta. Solo las direcciones de correo electrónico de la universidad están permitidas para el registro.

Una vez en la vista principal, se puede descubrir los proyectos deslizando con el dedo hacia arriba para ir desplazando las diferentes fichas de proyectos en la lista.

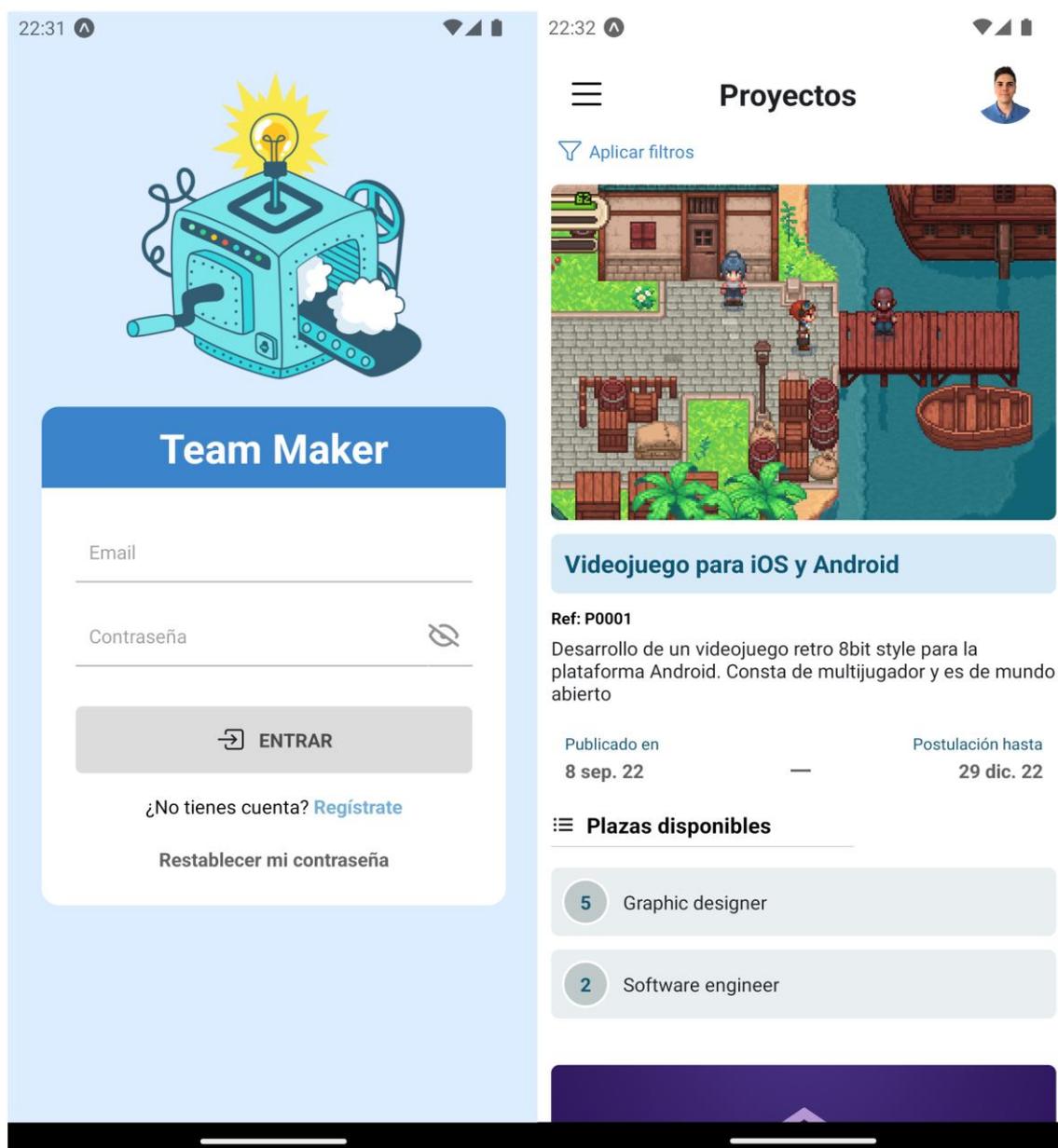


Figura 34. Vistas iniciales de TeamMaker

A continuación se pueden aplicar los criterios de búsqueda específicos. Para ello pulsar el botón de “Aplicar filtros”. Se despliega una hoja desde debajo de la pantalla del dispositivo donde se puede buscar por texto y/o beneficiarse del resto de filtros. Cuando el usuario encuentra un proyecto de su interés pulsar sobre la imagen o el título del proyecto y acceder a la ficha individual.

Desde la ficha individual se puede descubrir el resto de información detallada del proyecto y solicitar unirse a alguno de los perfiles con plazas libres. En caso de que ya se haya solicitado ese perfil, se puede consultar el estado de la solicitud.

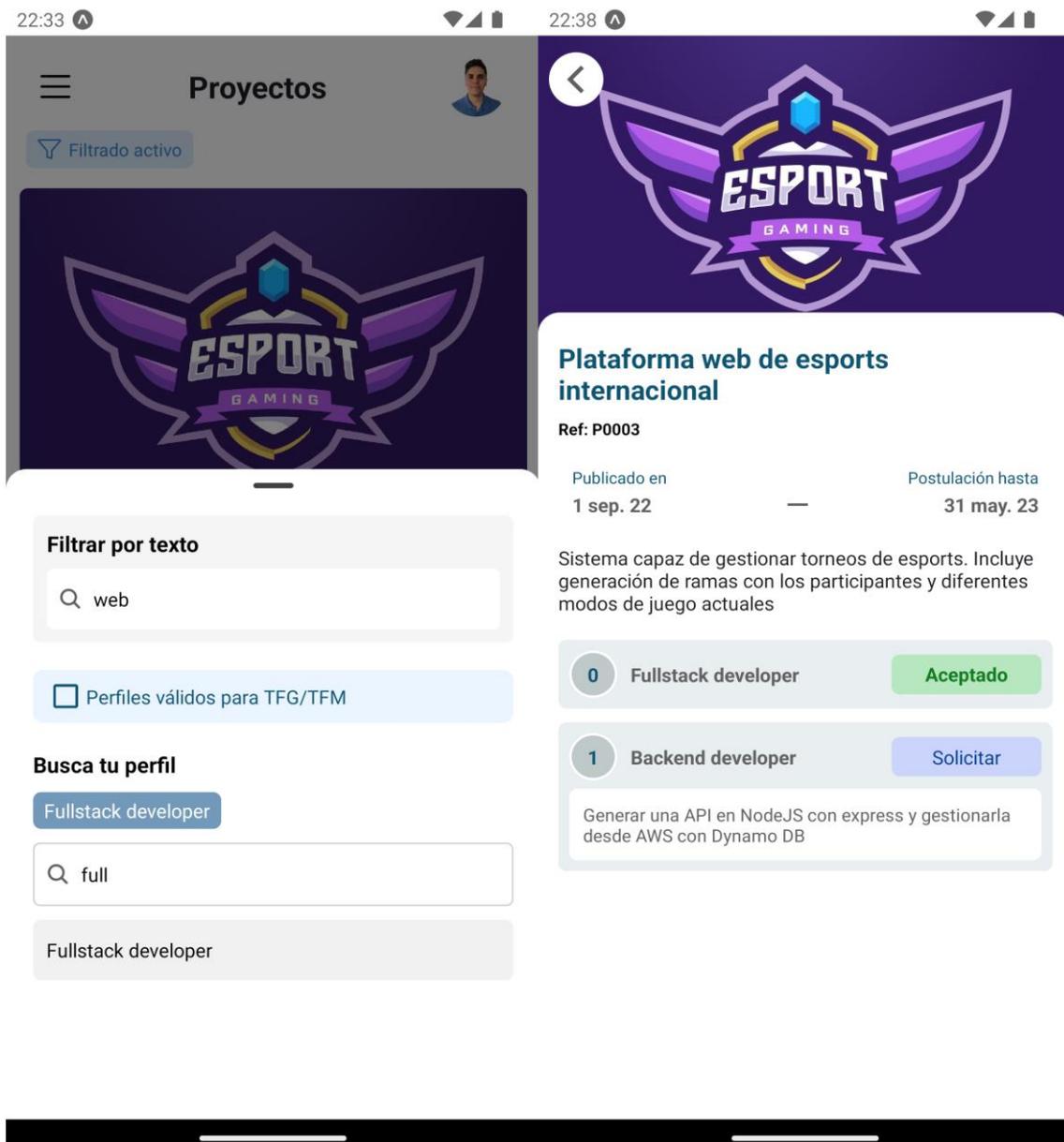


Figura 35. Herramienta de búsqueda y ficha personal de TeamMaker

TeamMaker: Plataforma para constituir equipos de trabajo multidisciplinares

Para acceder al menú de la aplicación hay que pulsar sobre el botón de las tres barras que se encuentra en la parte superior izquierda de la aplicación. En ese punto se abre un menú de tipo deslizante desde el lateral izquierdo del dispositivo. Para moverse entre ventanas simplemente hay que pulsar la sección donde se quiere navegar. En la parte más baja se encuentra el botón de cerrar sesión.

Para ver los proyectos creados pulsar sobre “Mis proyectos”. En esta ventana se muestran los proyectos que el usuario ha publicado y su estado. Puede ver si hay alguna solicitud de algún usuario de la comunidad pendiente de gestionar. Se indica a modo de alerta (además de las notificaciones que recibe desde la aplicación). Pulsar sobre cualquiera de los perfiles para acceder al gestor de solicitudes. También se puede editar los datos del proyecto desde esta vista de la aplicación.

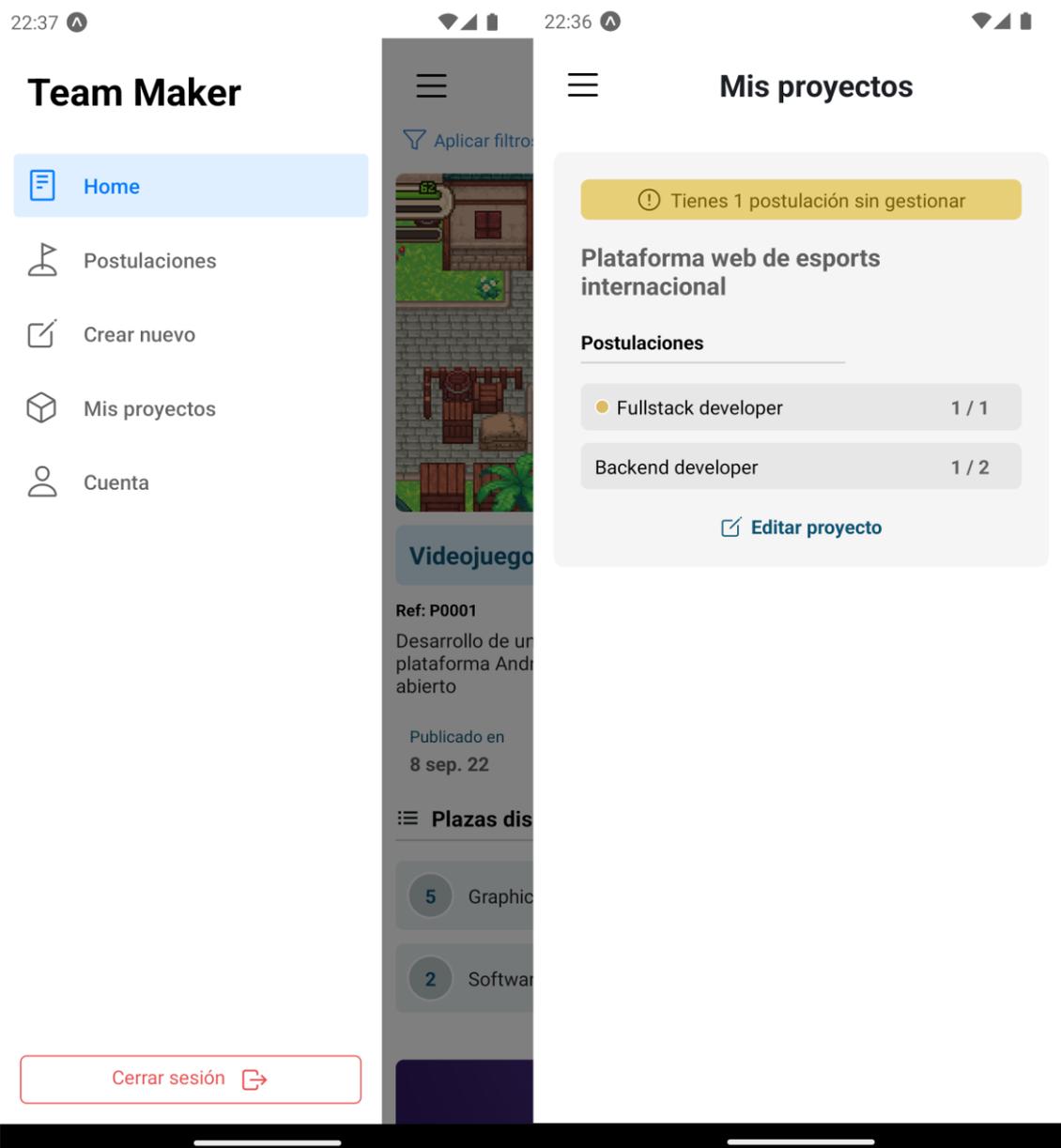


Figura 36. Navegación y gestión de mis proyectos

En la vista de solicitudes de un perfil del proyecto seleccionado, se puede de forma rápida e intuitiva conocer el estado de las plazas aceptadas respecto de las ofertadas. A continuación se muestra el listado de solicitantes, donde el usuario creador del proyecto puede ver los datos de presentación y contacto de cada uno de los solicitantes y manejar el estado de la solicitud. En cualquier momento se pueden deshacer los cambios en la respuesta enviada.

Como solicitante el usuario puede ver en el apartado “Postulaciones” el estado de las solicitudes que ha realizado a los perfiles de otros proyectos. De forma muy visual el usuario puede ver si por ejemplo ha sido aceptado, la solicitud sigue pendiente, etc.

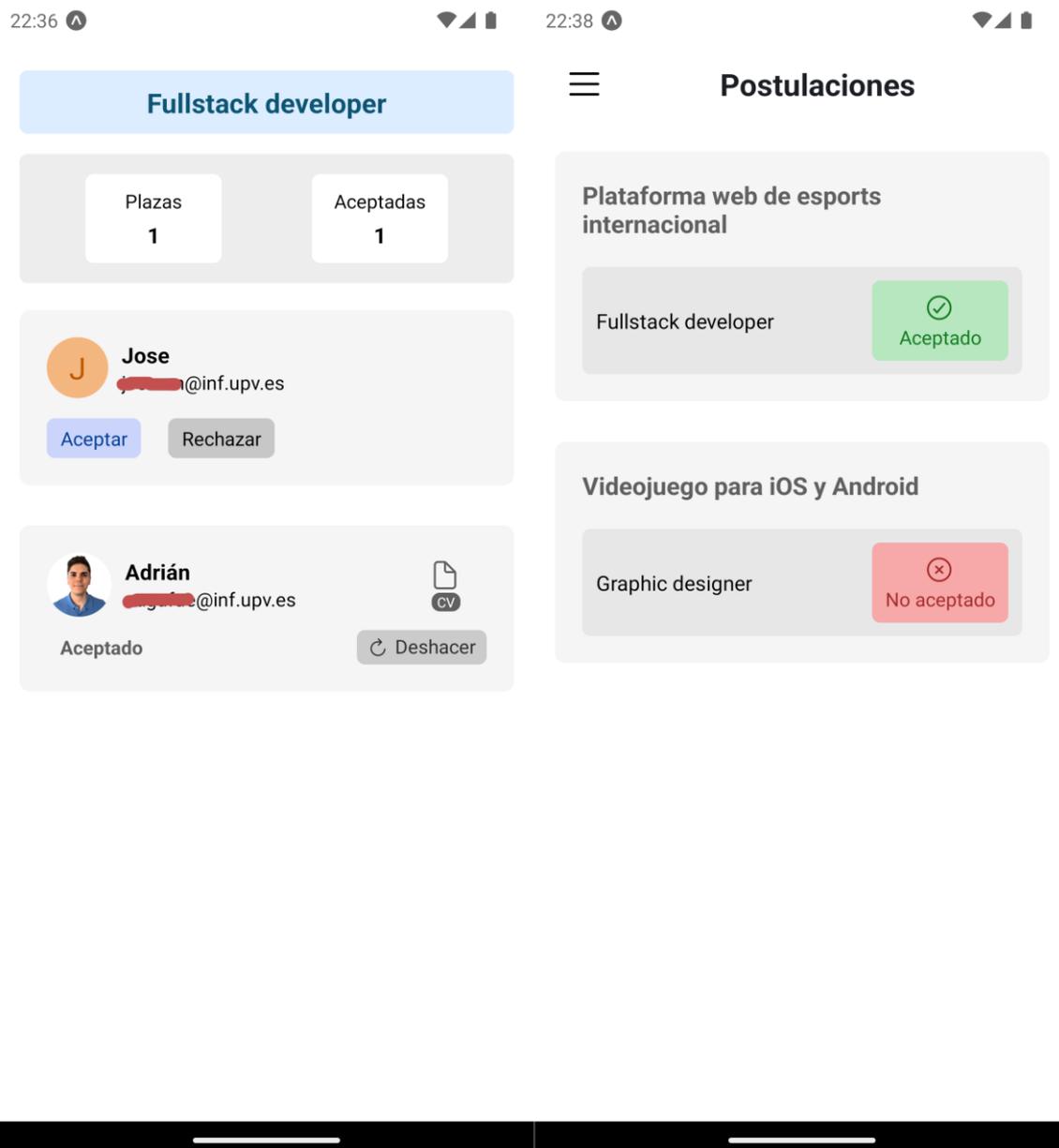


Figura 37. Manejo de las solicitudes a perfiles