



**Escuela Superior  
de Ingeniería y Tecnología**  
Universidad de La Laguna

## Trabajo de Fin de Grado

---

**Aplicación para intercambio de recursos  
entre estudiantes**

*Application for exchange of resources between students*

Ainoa Iglesias Dasilva

---

La Laguna, 13 de julio de 2023

D. **Francisco Javier Rodríguez González**, con N.I.F. 43.618.712-V profesor Asociado de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como tutor

D. **Alejandro Pérez Navas**, con N.I.F. 43.821.179-S profesor Asociado de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como cotutor

## **C E R T I F I C A ( N )**

Que la presente memoria titulada:

*"Aplicación para intercambio de recursos entre estudiantes"*

ha sido realizada bajo su dirección por D. **Ainoa Iglesias Dasilva**, con N.I.F. 77.485.946-G.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 13 de julio de 2023

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-  
NoComercial-SinObraDerivada 4.0 Internacional.

## **Resumen**

*El concepto de las monedas ha evolucionado a lo largo de la historia, adoptando diversas formas y funciones. En este trabajo de fin de grado se realiza un estudio sobre las monedas locales y virtuales, sin entrar en el mundo de las criptomonedas, y se propone una posible implementación de una moneda virtual en el ámbito universitario.*

*En el marco de este proyecto, se ha desarrollado un prototipo para dispositivos Android que utiliza una moneda virtual local. Los usuarios tienen la capacidad de intercambiar recursos como apuntes, conocimientos y proyectos, utilizando la moneda como medio de pago. Se ha creado un formulario que permite a los usuarios publicar sus ofertas o solicitudes, teniendo los demás usuarios la posibilidad de aceptarlas. Además, el prototipo incluye un monedero virtual que muestra el saldo de monedas disponibles y las transacciones realizadas.*

*Estas monedas también pueden ser utilizadas para canjear elementos dentro de la aplicación, como entradas para eventos universitarios o créditos académicos.*

*Con el fin de evaluar la viabilidad del proyecto, se ha realizado un estudio detallado en el apartado 4 de esta memoria, donde se analizan los posibles métodos de comercialización de una aplicación de este tipo, los tiempos y costes del proyecto además de un estudio del retorno de la inversión.*

*El desarrollo de esta aplicación ofrece nuevas oportunidades para fomentar la colaboración y el intercambio de recursos entre los estudiantes universitarios, promoviendo así un entorno de aprendizaje enriquecido y una mayor participación en las actividades académicas.*

**Palabras clave:** Moneda Virtual, Moneda Local, Android, Universidad

## **Abstract**

*The concept of currencies has evolved throughout history adopting various forms and functions. In this final degree project, a study on local and virtual currencies is carried out, without entering the world of cryptocurrencies, and a possible implementation of a virtual currency in the university environment is proposed.*

*As part of this project a prototype has been developed for Android devices that uses a local virtual currency. Users have the ability to exchange resources such as notes, knowledge and projects, using the currency as a means of payment. A form has been created that allows users to post their offers or requests, and other users can accept them. In addition, the prototype includes a virtual wallet that shows the balance of coins available and transactions made.*

*These coins can also be used to redeem items within the application, such as tickets to university events or academic credits.*

*In order to assess the feasibility of the project, a detailed study has been carried out in section 4 of this report, which analyzes the possible marketing methods for an application of this type, the time and costs of the project and a study of the return on investment.*

*The development of this application offers new opportunities to foster collaboration and resource sharing among university students, thus promoting an enriched learning environment and greater participation in academic activities.*

**Keywords:** Virtual Currency, Local Currency, Android, University

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Definición del problema . . . . .	2
1.2. Justificación . . . . .	3
1.3. Estado actual . . . . .	3
1.4. Tendencias de mercado . . . . .	5
1.5. Objetivos . . . . .	6
<b>2. Estudio previo</b>	<b>7</b>
2.1. Frontend . . . . .	7
2.1.1. Aplicación Móvil . . . . .	7
2.1.2. Aplicación Web . . . . .	8
2.2. Backend . . . . .	9
2.2.1. Servidor . . . . .	9
2.2.2. Despliegue . . . . .	9
2.2.3. Base de datos . . . . .	10
2.3. Lenguajes . . . . .	11
2.4. Elección . . . . .	12
<b>3. Desarrollo</b>	<b>14</b>
3.1. Backend . . . . .	14
3.2. Frontend . . . . .	17
3.2.1. Capa de presentación . . . . .	18
3.3. Despliegue . . . . .	27
<b>4. Estudio de viabilidad económica</b>	<b>29</b>
4.1. Funcionalidades de la aplicación . . . . .	29
4.2. Desarrollo del proyecto . . . . .	30
4.2.1. Costo y duración . . . . .	31
4.3. Modelo de comercialización . . . . .	32
4.3.1. Fuentes de ingresos . . . . .	32
4.3.2. Estudio del retorno de inversión (ROI) . . . . .	33
<b>5. Conclusiones y líneas futuras</b>	<b>35</b>
<b>6. Summary and Conclusions</b>	<b>36</b>
<b>7. Presupuesto</b>	<b>37</b>

# Capítulo 1

## Introducción

### 1.1. Definición del problema

Las monedas locales y virtuales[6] han ganado popularidad en los últimos tiempos debido a los avances digitales y el éxito observado en la implementación de muchas de ellas. Pero, ¿qué son estas monedas?. A lo largo de la historia las formas de pago y el dinero han cambiado significativamente, pero siempre cumplen tres funciones:

- Medio de cambio: permite el intercambio de bienes, servicios o conocimientos.
- Unidad de cuenta: permite dar valor a las cosas y fijar un coste al intercambio.
- Reserva de valor: el dinero mantiene algo de su valor por lo que puede acumularse y usarse en el futuro.

Las monedas locales y virtuales cumplen estas características, pero es importante destacar que no son consideradas legales y que por su limitación y alcance no pueden ser un sustituto del dinero de curso legal.

Al hablar de monedas virtuales y locales, no nos referimos a criptomonedas<sup>1</sup>, sino a aquellas monedas digitales, tokens o puntos usados en un ámbito y comunidad específicos que permiten su uso para comprar elementos únicamente en ese contexto, es decir, son un medio de cambio y que pueden tener relación con el dinero real o no, y pueden ser un medio de pago alternativo o complementario en el caso de las monedas locales.

La finalidad de las monedas locales, también conocidas como moneda social[27][4], es ser una alternativa al medio de pago de curso legal

---

<sup>1</sup>**Criptomoneda:** Es un activo digital con un cifrado criptográfico que garantiza la titularidad e integración de las transacciones y controla su copia. No están limitados a un entorno local y no están controlados por los gobiernos.

usado en un ámbito geográfico muy limitado para fomentar su actividad económica y el consumo de productos de kilómetro 0.

En el caso de las monedas virtuales, su uso es más extenso pero de forma general son usadas como medio de cambio de productos virtuales o como medio para fomentar el mayor uso de una aplicación.

En base al estudio realizado de las monedas virtuales y locales, se quiere implementar, en el ámbito universitario, el uso de una aplicación con una moneda virtual propia, de forma que se motive a los estudiantes a interactuar y ayudarse entre ellos y a crear y participar más en actividades de la universidad.

## **1.2. Justificación**

La aplicación presentada tiene como objetivo principal promover la interacción y colaboración entre estudiantes de diversos grados universitarios, así como fomentar su participación en actividades académicas. Para lograrlo, se ha desarrollado un sistema que permite a los estudiantes intercambiar conocimientos y recursos entre sí, lo que contribuye a fomentar la colaboración y la creatividad.

Para esto, se espera que la aplicación tenga las siguientes funciones:

- Un sistema que permita a los estudiantes hacer intercambios entre ellos de sus conocimientos o recursos, lo que puede fomentar la colaboración y creatividad de los estudiantes.
- Utilizar una moneda virtual como medio de pago para los servicios ofrecidos por otros estudiantes.
- Crear un medio de cambio de estas monedas fuera del ámbito de la aplicación. Una posibilidad es permitir el intercambio de un número de monedas por una cantidad a determinar de créditos. El objetivo de esta medida es promover el uso de la aplicación.

## **1.3. Estado actual**

Aunque existen numerosas aplicaciones dirigidas a estudiantes[32], no es fácil encontrar alguna que haga uso de este tipo de monedas. Además el propósito de este proyecto es permitir el intercambio de recursos más allá de los simples apuntes.

En cuanto a las aplicaciones que utilizan monedas virtuales y locales, se pueden encontrar numerosos casos de éxito, especialmente en el ámbito de los videojuegos y en áreas geográficas pequeñas.

Un ejemplo de moneda local en España es el "Elio"[8], implementado por el Ayuntamiento de Lebrija, un pueblo de Sevilla, con el objetivo de impulsar el comercio local y brindar apoyo económico a las familias. Esta moneda local ha sido ampliamente aceptada. Cada familia recibe una cantidad de Elios, que varía entre 50 y 200 según el número de hijos, y pueden utilizarlos para pagar el 50 % del valor de los productos, abonando el resto en euros. Sin embargo, su uso se limita a los comercios adheridos al proyecto, que en 2020 contaba con 165 establecimientos. Esta iniciativa promueve los negocios locales y evita la salida de dinero del pueblo. Es importante destacar que esta moneda no es una criptomoneda.



(a) Moneda Elio.



(b) Puntos Reciclos.

Por otro lado, entre las aplicaciones o proyectos con monedas virtuales, encontramos Plasticon[26] un proyecto nacido en Uruguay para impulsar el reciclaje usando una moneda virtual para dar valor a los residuos, en este caso es un proyecto ideado para usar blockchain en el futuro. Es común encontrar este tipo de monedas en muchas aplicaciones, plataformas y videojuegos, donde se utilizan como una forma de monetización[11]. Algunos ejemplos son juegos como Fortnite, la plataforma Riot<sup>2</sup> o la aplicación de reciclaje Reciclos<sup>3</sup>. Estas monedas propias del juego o plataforma se pueden obtener tanto mediante compras con dinero real y legal como a través de la participación en el propio servicio, y permiten la adquisición de elementos para mejorar la experiencia de juego o uso de la plataforma.

---

<sup>2</sup>**Riot:** esta plataforma usa las 'esencias azules' como forma principal de pago en para desbloquear contenido de League of Legends y pueden conseguirse jugando. También cuenta con Riot Point que se consiguen a través de dinero real.

<sup>3</sup>**Reciclos:** Aplicación para promover el reciclaje. Premia a los usuarios que reciclen con unos puntos llamados Reciclos que pueden usar para canjear por premios.

## 1.4. Tendencias de mercado

Actualmente, está en debate cuál será el futuro de los métodos de pago, hoy en día, tanto a nivel mundial como en la mayoría de los países el pago en efectivo es el método más usado[7], sin embargo, el aumento en el uso de tarjetas bancarias y pagos con el móvil es un hecho y se prevé que siga aumentando hasta haciendo incluso desaparecer el efectivo en algunos países[13]. Esto es debido a numerosas variables entre ellas el aumento del uso de aparatos digitales y sus avances tecnológicos y la seguridad que proporciona a los usuarios al ser más complicado los robos y fraudes.

Esto también está promoviendo, que las aplicaciones fuera del ámbito de los videojuegos, se decanten por el uso de monedas dentro de sus app que puedan ser obtenidas con dinero real y almacenadas en monederos virtuales. A continuación, se detallan algunos beneficios de esta práctica[1]:

- **Generar ingresos:** La implementación de monedas virtuales puede proporcionar una nueva fuente de ingresos para las aplicaciones. Estas monedas pueden ser adquiridas mediante compras con dinero real o a través de la realización de ciertas actividades dentro de la aplicación. Al permitir que los usuarios compren monedas, las aplicaciones pueden obtener beneficios económicos adicionales.
- **Fomentar la participación y retención de los usuarios:** Las monedas virtuales son una herramienta eficaz para incentivar a los usuarios a participar de manera más activa en la aplicación y para aumentar la retención. Al otorgar recompensas y desbloquear beneficios exclusivos a través de las monedas virtuales, las aplicaciones pueden motivar a los usuarios a realizar acciones específicas, como completar tareas, interactuar con otros usuarios o alcanzar determinados hitos. Esta estrategia crea un sentido de logro y satisfacción, lo que aumenta la implicación y el compromiso de los usuarios con la aplicación.
- **Personalización y gamificación<sup>4</sup>:** La tendencia de la gamificación[9] se ha vuelto cada vez más popular en diversos ámbitos. La gamificación consiste en aplicar elementos y mecánicas propias de los juegos en contextos no relacionados con los juegos en sí. Mediante el uso de monedas virtuales, las aplicaciones pueden introducir un sistema de recompensas y desafíos que fomenta la participación activa de los usuarios. Esto crea una experiencia más divertida y motivadora, generando competitividad y colaboración entre los usuarios.

---

<sup>4</sup>**Gamificación:** método que aplica mecánicas de los juegos en otros ámbitos con el fin de atraer personas, mejorar el aprendizaje, mejorar habilidades...

- Crear economías internas: Una de las ventajas más destacadas de las monedas virtuales es la capacidad de crear una economía interna dentro de la aplicación. Estas monedas permiten a los usuarios intercambiar bienes, servicios o recursos entre ellos, fomentando la colaboración y fortaleciendo la comunidad dentro de la aplicación. Al establecer un sistema de comercio interno basado en la moneda virtual, se crea un ambiente propicio para la interacción entre los usuarios y se promueve el sentido de comunidad y reciprocidad.

## 1.5. Objetivos

El objetivo de este estudio es determinar la viabilidad y la aceptación de una aplicación Android con las características mencionadas. Para lograrlo, se establecen los siguientes objetivos específicos:

- Analizar el mercado para conocer la situación actual y por venir de aplicaciones similares.
- Analizar requisitos tecnológicos, el hardware, software y tecnologías necesarias para desarrollar la aplicación.
- Definir características que hagan atractiva y única la aplicación.
- Desarrollo de la arquitectura:
  - Se puede crear/modificar/eliminar solicitudes.
  - Se gestiona el pago/cobro de las solicitudes.
  - Registro/inicio de sesión.
  - El backend gestiona la base de datos y la API.
  - Aplicación Android que permita lo anterior y visualice los datos de forma intuitiva.
  - Aplicación Web sencilla que permita lo anterior.
- Realizar tests a la aplicación.
- Desarrollar un plan de negocio que contemple el desarrollo de una aplicación de estas características de maneda profesional, para lo cual se realiza un diagrama de Gantt<sup>5</sup>, se calcula los gastos del proyecto, se crea un modelo de comercialización y se realiza el calculo del ROI<sup>6</sup>.

---

<sup>5</sup>**Diagrama de Gantt:** Es una herramienta que ayuda en la gestión y planificación de proyectos visualizando de forma sencilla las tareas y tiempos dedicados a cada una.

<sup>6</sup>**ROI:** Retorno sobre la inversión. Es una métrica que comparar el beneficio con la inversión realizada, por lo que sirve para analizar el rendimiento financiero.

# Capítulo 2

## Estudio previo

A continuación se explican las distintas tecnologías y herramientas estudiadas para realizar este proyecto y cuales fueron seleccionadas[25].

### 2.1. Frontend

#### 2.1.1. Aplicación Móvil

La selección de las tecnologías para desarrollar la aplicación móvil se basó en las estadísticas aportadas por el portal Statista[30]<sup>1</sup> de las frameworks<sup>2</sup> móvil mas usados en los últimos tres años.

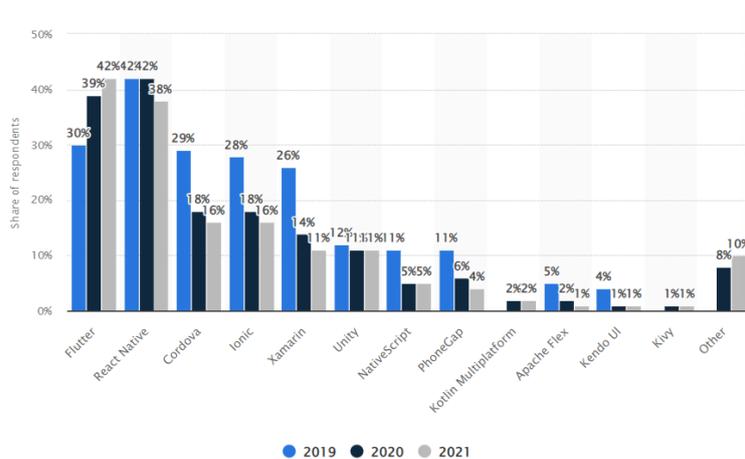


Figura 2.1: Frameworks móvil más usados. Statista

- Android Studio<sup>3</sup>: es el IDE oficial para desarrollar aplicaciones Android usado principalmente con Java y Kotlin. Tiene numerosas funciones que ayudan en la productividad e integra un emulador para dispositivos Android.

<sup>1</sup>**Statista:** Es un portal de estadísticas en línea. Proporciona una gran variedad de datos e información en distintos idiomas y con fuentes fiables de investigaciones de mercado y opción, organizaciones empresariales y gobiernos.

<sup>2</sup>**Framework:** Es un entorno de trabajo estandarizado, con un conjunto de prácticas, conceptos y criterios a seguir.

<sup>3</sup>**Android Studio:** Es el entorno de desarrollo oficial para la plataforma Android. Incorpora características que permiten un trabajo mas fluido, como el simulador virtual de Android o la compilación basada en Gradle.

- React Native[22]: framework basado en JavaScript y ReactJS para el desarrollo de aplicaciones nativas tanto Android como iOS que genera una aplicación nativa con rendimiento casi igual al nativo. Tiene una curva de aprendizaje sencilla.
- Xamarin: plataforma de código abierto para compilar de forma nativa aplicaciones, con un mejor rendimiento en iOS, Android y Windows con .NET. Es muy útil para crear aplicaciones multiplataforma ya que permite compartir una media del 90 % del código. Suele ser usado junto al lenguaje c# y Visual Studio Code.
- Ionic[15]: junto con Vue, Angular o React, permite la creación de aplicaciones móviles multiplataforma. Esta basado en HTML, CSS y JavaScript.
- Flutter[14]: es un SDK<sup>4</sup> creado por Google para el desarrollo multiplataforma de aplicaciones móvil. Utiliza un lenguaje de programación propio llamado Dart que actualmente no es muy conocido pero es similar a Java y C#.
- NativeScript-Vue: plugin de NativeScript para crear aplicaciones móviles nativas usando Vue. Vue es un framework que está cogiendo cada vez más popularidad en el desarrollo web ya que tiene una curva de aprendizaje muy fácil. NativeSript-Vue se usa con JavaScript, CSS<sup>5</sup> y HTML<sup>6</sup>.

### 2.1.2. Aplicación Web

Las herramientas de la aplicación web se han buscado partiendo de los datos obtenidos de las estadísticas de Stack Overflow<sup>7</sup> para el año 2023[31]. Entre las más usadas se han buscado las que son más acordes al proyecto .

- React JS[18]: biblioteca JavaScript creada por Facebook para componentes de la interfaz de usuarios. Tiene una curva de aprendizaje asequible y tiene código común con React Native.
- Vue.js[19]: framework progresivo de JavasCript para crear interfaces de usuarios. Tiene la curva de aprendizaje más sencilla entre los framework mas usados (React, Angular, Vue) y usa HTML y CSS.
- Next.js[17]: es un framework basado de React caracterizado por el renderizado del lado del servidor y enrutamiento automático y es sencillo de usar.

---

<sup>4</sup>**SDK:** Software Development Kit, es un conjunto de herramientas para programar aplicaciones.

<sup>5</sup>**CSS:** Es un lenguaje de diseño de documentos escritos en lenguajes de marcado como HTML.

<sup>6</sup>**HTML:** Es un lenguaje de marcado usado para la creación de páginas web.

<sup>7</sup>**Stack Overflow** Es un sitio web público de preguntas y respuestas enfocado en los profesionales TI. Tiene una gran aceptación y afluencia de usuarios y cada año saca un análisis en torno de Desarrollo Software basado en los miembros registrados.

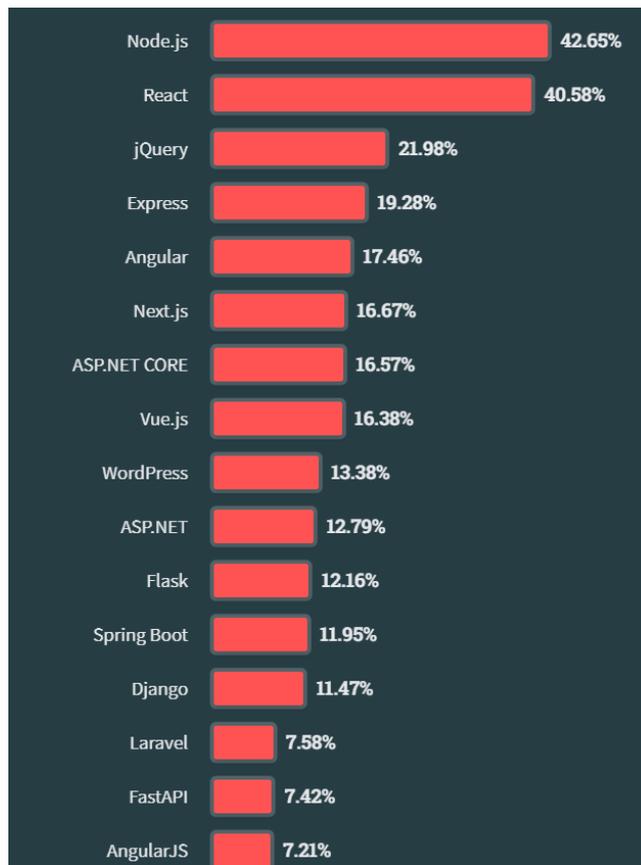


Figura 2.2: Frameworks web más usados. *Stack Overflow, 2023*

## 2.2. Backend

### 2.2.1. Servidor

- Node JS[24]: entorno que trabaja en tiempo de ejecución para JavaScript, de código abierto, multiplataforma, tiene un gran rendimiento y hace uso de NPM<sup>8</sup>.
- Express JS[12]: framework backend más popular desarrollado sobre Node Js. Contiene todas las funcionalidades para desarrollar un aplicación permitiendo el manejo de peticiones HTTP, gestionar los puertos, añadir middlewares... Es minimalista y de uso fácil.

### 2.2.2. Despliegue

Las opciones para el despliegue se tuvieron en cuenta según los datos observados en Stack Overflow, si tenían opción gratuita y recomendaciones.

- Digital Ocean: es un BaaS, es un servicio de infraestructura en la nube que ofrece alojamiento. Destaca por su servicio calidad-precio y ofrece numerosos servicios como droplets, almacenamiento y otros. Cuenta con planes gratuitos.
- Firebase: es un BaaS y como las anteriores, cuenta con un plan gratuito que permite un amplio uso de sus funciones.

<sup>8</sup>**NPM:** Es un manejador de paquetes de Node, la herramienta por defecto para JavaScript para instalar paquetes y manejar comodamente las dependencias de un proyecto.

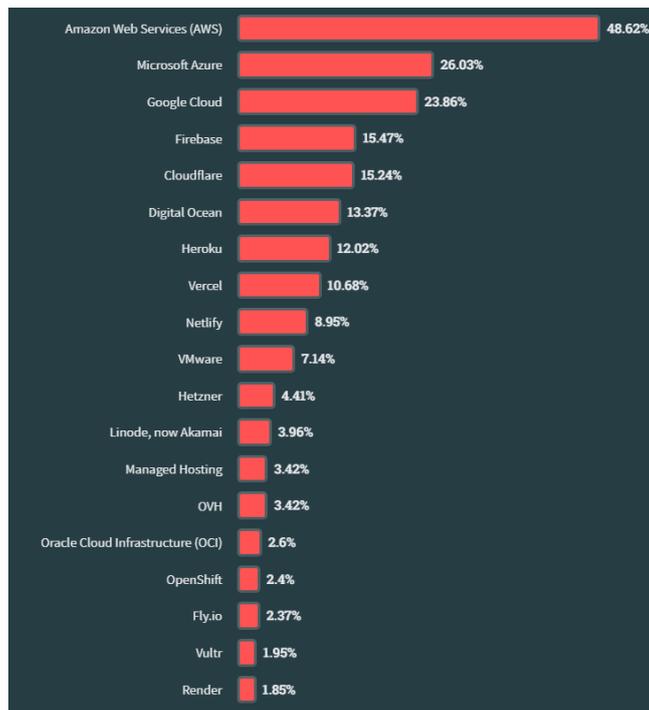


Figura 2.3: Plataformas cloud más usadas. *Stack Overflow, 2023*

- **Render:** es una plataforma de alojamiento y despliegue de aplicaciones web y API. Proporciona numerosas características y destaca por su facilidad de uso e integración con GitHub. Entre sus planes cuenta con uno gratuito para estudiantes.
- **Vercel:** es una plataforma de despliegue de aplicaciones web que se enfoca en dar una experiencia rápida y eficiente. Sirve para aplicaciones web estáticas y dinámicas. Vercel está integrado con Next.js y React. Ofrece una opción gratuita para proyectos personales.

### 2.2.3. Base de datos

Al igual que los anteriores apartados, se hizo una selección basado en los gráficos de bases de datos más usados de Stack Overflow y su sencillez.

- **Firebase Realtime Database:** Plataforma de desarrollo de apps que cuenta con una base de datos NoSQL que se aloja en la nube. Cuenta con el respaldo de Google, es de pago con una opción gratuita.
- **MongoDB:** es una base de datos no relacional conocida por su escalabilidad y flexibilidad. Utiliza un modelo de documentos que permite el almacenamiento y consulta de datos de manera eficiente. MongoDB es fácil de aprender y usar, lo que lo convierte en una opción popular para aplicaciones web y móviles.
- **MySQL:** sistema de gestión de bases de datos relacionales de código abierto. Es ampliamente utilizado en el desarrollo web y ofrece un conjunto completo de características para almacenar, administrar y consultar datos estructurados. MySQL es conocido por su estabilidad, rendimiento y seguridad, y es compatible con múltiples plataformas.

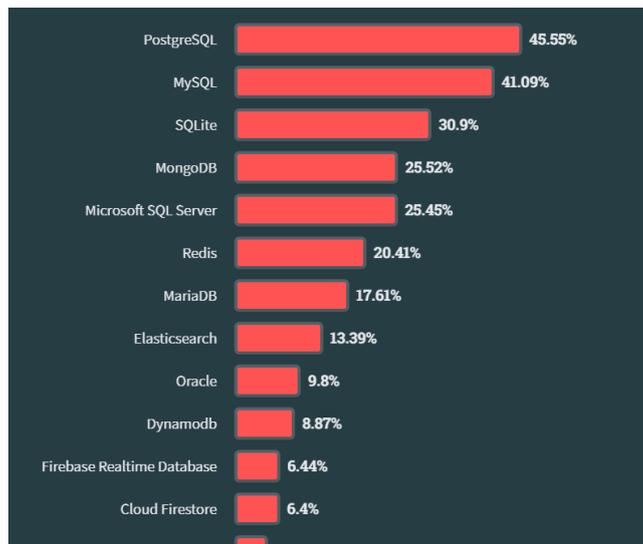


Figura 2.4: Bases de datos más usadas. *Stack Overflow, 2023*

## 2.3. Lenguajes

De igual manera, se obtuvo los datos de los lenguajes más populares en la misma plataforma y se eligieron lo más acordes a este proyecto.

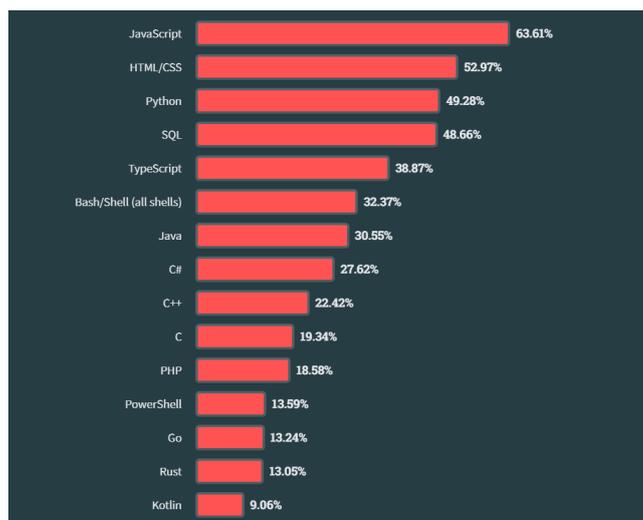


Figura 2.5: Lenguajes más usados. *Stack Overflow, 2023*

- Javascript: lenguaje tanto para el lado del cliente como del servidor, usado principalmente para el desarrollo de paginas web interactivas.
- Typescript: lenguaje construido sobre Javascript que incluye mejoras adicionales como módulos o el tipado fuerte que puede ayudar a prevenir errores.
- Kotlin: Es un lenguaje de código abierto, interpretado por Java, usado principalmente para aplicaciones Android nativas y puede usarse tanto en la parte del servidor como del cliente. El motivo principal de tener en cuenta este lenguaje es que actualmente es el más popular para desarrollar en Android desbancando a Java y siendo el recomendado oficialmente. Se considera que tiene una curva de aprendizaje fácil, tiempo de programación reducido, es interoperable con Java, flexible y multiplataforma.

- Java: el lenguaje por excelencia para programar aplicaciones Android nativas. Es rápido, seguro y fácil de usar. Es un lenguaje multiplataforma y orientado a objetos. El motivo de su elección es que la gran parte de aplicaciones Android están desarrolladas en este lenguaje, además es uno de los lenguajes de programación más usados ya que tiene múltiples usos y lo que implica que tiene una comunidad muy activa.
- Python: lenguaje de alto nivel interpretado muy usado en desarrollo de aplicaciones web, software, ciencia de datos... Eficiente y fácil de leer.

## 2.4. Elección

Después de hacer un estudio sobre las tecnologías necesarias, la elección final fue usar para el proyecto un stack MERN, es decir, hacer uso de MongoDB Atlas, Express JS, React JS/ React Native y Node JS. Para la elección se tuvo en cuenta lo siguiente:

- La aplicación debe ser para Android Nativo no híbrida.
- Conocimientos previos de las tecnologías para facilitar el progreso de proyecto.
- Backend escalable y personalizable.
- Documentación disponible en la web y comunidad activa, para facilitar tanto el aprendizaje de las herramientas como el poder encontrar solución a los problemas que surjan a lo largo del proyecto.

React Native permite la creación de aplicaciones nativas además de que comparte mucho código con React, también son un framework con una gran comunidad en internet y mucha documentación disponible, aunque no poseo conocimientos en aplicaciones móvil, si he trabajado antes en React, por lo que opté por esta opción entre todas para el frontend. Para la web, concretamente se seleccionó Next.js por las valoraciones positivas que tiene y su estructura. En el backend Firebase fue descartado debido a que es un BaaS ya que, aunque permite centrarse en la aplicación de cara al usuario y reduce el tiempo del proyecto, es menos flexible al no tener control ni personalización sobre el servidor ni base de datos. Además no poseo conocimientos en herramientas de este tipo mientras si lo tengo en MongoDB. Tras estas elecciones la decisión final fué usar la pila MERN que hace uso de Express y Node para el servidor en el backend. Puesto que se eligió Next.js y React Native que están basados en Javascript, el lenguaje escogido fue TypeScript que es una mejora de Javascript y a la

hora de crear un proyecto con React Native se instala con este lenguaje por defecto, también puede usarse tanto en el frontend como el backend.

El motivo final para elegir estas herramientas es que, como se vio en las gráficas anteriores, son de las más usadas hoy en día y van manteniéndose a lo largo de los años, por lo que aprenderlas o mejorarlas mientras se realiza este proyecto ayudará en el futuro.

Respecto al despliegue, debido a la sencillez de configurarlos e integrarlos con GitHub y por sus planes gratuitos escogí usar Render.com para el backend y Vercel para la web. Otro motivo para escoger Vercel es que es la plataforma recomendada por Next.js y viene integrado en sus proyectos.



(a) Stack MERN



(b) Next.js



(c) TypeScript



(d) Render.com



(e) Vercel



(f) Git y GitHub

Por último, otras herramientas y servicios a usar son:

- Visual Studio Code: como editor de todo el proyecto.
- Android Studio, JDK y Node para configurar y usar React Native para Android.
- Mongoose para gestionar la base de datos.
- Tailwind CSS para el diseño de la web.
- Doxygen para documentación del código.
- Git y GitHub para control de versiones.
- Overleaf para el desarrollo de la memoria.

# Capítulo 3

## Desarrollo

### 3.1. Backend

El backend utilizado en la aplicación móvil y web es el mismo y se encarga de gestionar la API para realizar las solicitudes a la base de datos. Su estructura es sencilla, con un archivo principal donde se crea el servidor y se establece la conexión con la base de datos, junto con las rutas de la API. Además, existen varios directorios que organizan el código de manera modular:

- **Models:** contiene los archivos que definen la estructura de los modelos de datos. Se creó una interfaz para cada modelo, la cual extiende la interfaz 'mongoose.Document' y describe la estructura de los datos, incluyendo los atributos, sus tipos de datos y los valores predeterminados. A continuación, se crea el esquema utilizando la clase 'Schema' de Mongoose, implementando la interfaz definida anteriormente. La base de datos se divide en los siguientes modelos:
  - User: Representa a los usuarios de la aplicación y contiene información como el nombre de usuario, la imagen de perfil, las credenciales de identificación como estudiante, el correo electrónico, la contraseña, la fecha de creación, la cartera (wallet), la calificación promedio (nota) y una lista de sus solicitudes creadas. Dentro del esquema se aplican validaciones utilizando el módulo 'validator', como que el nombre de usuario sea alfanumérico, la contraseña tenga al menos 4 caracteres, el correo electrónico sea válido y la credencial sea del tipo 'aluxxxxxxxxx'.
  - Request: Son las solicitudes realizadas por los usuarios en la aplicación, ya sea para ofrecer algo o pedir algo. Se establece un campo 'type' que indica si es una solicitud ('request') u oferta ('offer'), y contiene un título, una descripción, el ID y nombre de

usuario del creador, un precio, la fecha de creación y un booleano que indica si está disponible para ser mostrada a todos los usuarios o mantenerse oculta.

- **Wallet:** Representa la cartera de cada usuario, donde se almacena el número de monedas y un historial de transacciones en forma de lista. Por defecto, cuando se crea un usuario por primera vez, se le asigna una cantidad de 30 monedas.
- **Transaction:** Se utiliza para mostrar a los usuarios un historial de las transacciones realizadas. Contiene un título (correspondiente al título de la solicitud aceptada), un tipo (compra o venta), el número de monedas de la transacción, el nombre de la segunda involucrada, el nombre del propietario de la solicitud y la fecha en que se realizó la operación.

Para evitar inconsistencias en la base de datos, se utiliza el ID generado por Mongoose para hacer referencia a otros objetos. Por ejemplo, en lugar de almacenar una lista completa de solicitudes dentro del objeto de usuario, se almacenan solo los IDs de las solicitudes relacionadas. De esta manera, se evita el almacenamiento y la carga de objetos JSON voluminosos y se previenen problemas como modificar una solicitud pero que no se refleje en el objeto de usuario.

- **Routes:** contiene los ficheros donde se establecen los enrutadores con express.js para manejar las rutas de las operaciones CRUD (get, delete, post, patch). Las rutas base se establecen en el fichero principal del servidor y de ahí se llama a los enrutadores de esta carpeta.

- **User:**
  - /user : para obtener todos los usuario o eliminar todos los usuarios.
  - /user/username/:username : para buscar un usuario por su username.
  - /user/id/:id : para buscar, eliminar o modificar un usuario con su id.
  - /user/register : para crear un nuevo usuario.
  - /user/login : para iniciar sesión en la app.
- **Request:**
  - /request : para crear, eliminar todas u obtener todas.

- /request/id/:id : para eliminar, obtener o modificar una request por su id.
- /request/owner/:id : para buscar la lista de request de un usuario.
- Wallet:
  - /wallet : para crear, eliminar todas u obtener todas.
  - /wallet/id/:id : para eliminar, obtener o modificar una cartera por su id.
- Transaction:
  - /transaction : para obtener todas.
  - /transaction/id/:id : para obtener o modificar una por su id.
  - /transaction/transfer : para realizar la transferencia al comprar o vender.
- **Controllers:** En la carpeta 'Controllers' se gestionan las operaciones CRUD (Crear, Leer, Actualizar y Eliminar). Cada modelo (user, wallet, transaction, request) tiene su propio archivo en esta carpeta. Los métodos en esta parte se encargan de obtener los datos pasados como parámetros o en el cuerpo de la solicitud HTTP, llamar a los métodos correspondientes en los servicios para procesar la petición y devolver la respuesta, ya sea en caso de error o devolviendo un objeto JSON. También se encargan de tareas como el cifrado de contraseñas antes de crear un usuario, utilizando la biblioteca bcrypt, y la generación de tokens de autenticación al iniciar sesión de un usuario. Además, se define qué datos se permiten devolver como respuesta para evitar, por ejemplo, la devolución de contraseñas cuando se busca un usuario. También se gestiona la eliminación en cascada, por ejemplo, al eliminar un usuario, se deben eliminar sus solicitudes, su cartera y las transacciones relacionadas con su historial de cartera.
- **Services:** Contiene los métodos a los que llama Controllers. Estos métodos se encargan de realizar las operaciones directamente en la base de datos utilizando Mongoose. Aquí es donde, si es necesario, se devuelven objetos que incluyen otros objetos relacionados en lugar de solo los IDs. Por ejemplo, al obtener los datos de un usuario, es útil poder ver directamente los datos de su cartera sin tener que hacer una segunda solicitud.

- **Middlewares:** Contiene lo necesario para verificar los tokens de autenticación comprobando si son válidos y no han caducado. Esto es necesario para la mayoría de peticiones ya que no cualquier usuario debe tener permisos para eliminar o editar cualquier información.
- **Config:** Guarda algunas variables necesarias para establecer la conexión con la base de datos, como el puerto y nombre de la base de datos, así como datos relacionados con los tokens, como el tiempo hasta de caducidad y la clave secreta.

## 3.2. Frontend

El frontend de la aplicación y la web son muy similares en cuanto a funcionamiento y estructura, con las claras variaciones de sus respectivos frameworks. Se hace uso de Context y Hooks de React para establecer un estado global y se ha organizado en los siguientes directorios.

- **Context:** Se encarga de mantener un estado global en la aplicación. En este caso, almacena los datos del usuario que ha iniciado sesión, un booleano que indica si el usuario está activo o no, y se definen tres métodos: login, updateUser y logout. Estos métodos se encargan de almacenar, actualizar y eliminar los datos del usuario actual en función de las acciones de inicio de sesión, actualización de datos o cierre de sesión.
- **Provider:** Gestiona los métodos definidos en el contexto y se encarga de proveerlos a los componentes que los necesiten.
- **Components:** En este directorio se encuentran los componentes (página) de la aplicación. Hace uso del Context para almacenar y obtener los datos del usuario que ha iniciado sesión, y también llama a Services para hacer peticiones al servidor.
- **Services:** Se encarga de realizar las peticiones al servidor usando la librería Axios y recibe las respuestas correspondientes.

### 3.2.1. Capa de presentación

El prototipo desarrollado cuenta con diversas pantallas y funcionalidades. Además, cuenta con botones y características que se muestran pero no están activos. Las pantallas existentes son las siguientes:

#### ■ Aplicación Móvil

- Welcome: Aquí iniciará la aplicación y dará las opciones para ir a Iniciar Sesión, Registrarse o Entrar como invitado.

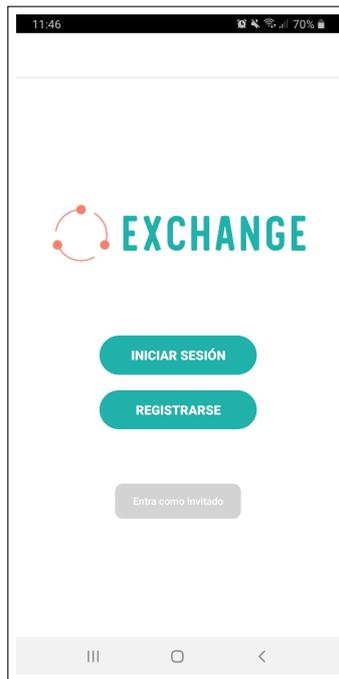


Figura 3.1: Pantalla Inicial de la App.

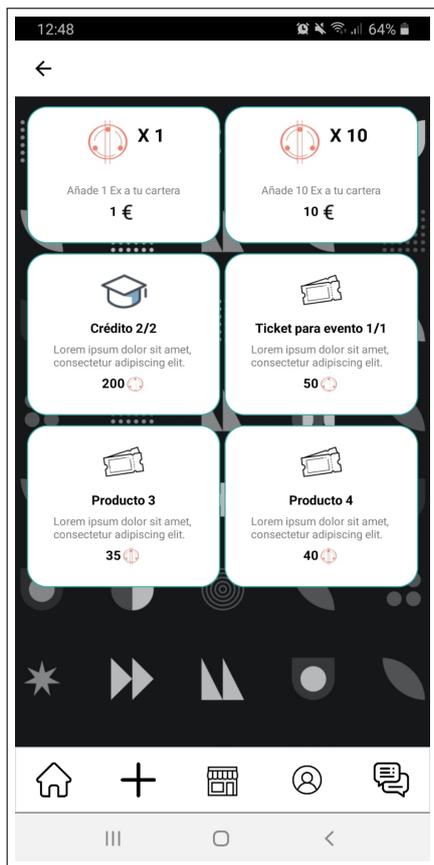
- Navbar App: La aplicación cuenta con una barra inferior que se muestra en todas las pantallas. Esta barra varía según si hay un usuario con la sesión iniciada o no, mostrando los botones para ir al home, a la tienda, al formulario de solicitudes, al perfil y chat en caso de estar conectado o, en caso contrario, solo tendrá los botones para ir al inicio de sesión, tienda y registro.

- Home: Esta pantalla se verá tras iniciar sesión o al entrar como invitado. Mostrará una lista con las últimas 10 solicitudes añadidas por los usuarios. Cada solicitud puede seleccionarse para verla en detalle y aceptarla.



Figura 3.2: Home de la App.

- Shop: Esta pantalla solo existe en el prototipo de la aplicación. Se puede acceder a ella desde la barra de navegación inferior y mostrará una lista de ítems que el usuario podría obtener. Si se pulsa en el botón para obtener monedas, se verá una simulación muy simple en la que se le dará a escoger un método de pago (visa, paypal o byzum), un formulario donde se recogerían los datos (está desactivado) y si se pulsa en el botón se añadirán las monedas al monedero.



(a) Tienda en la App.



(b) Métodos de pago.



(c) Formulario para datos de pago.



(d) Aviso compra correcta.

## ■ Web

- Home: Muestra una lista con las últimas 10 solicitudes añadidas por los usuarios y cuenta con un botón que abre un formulario para rellenar y registrar una solicitud. Será la primera pantalla que se vea al entrar y variará un poco según si se entra con un usuario registrado o como un invitado ya que no se mostrará el botón del formulario de solicitud si no ha iniciado sesión.

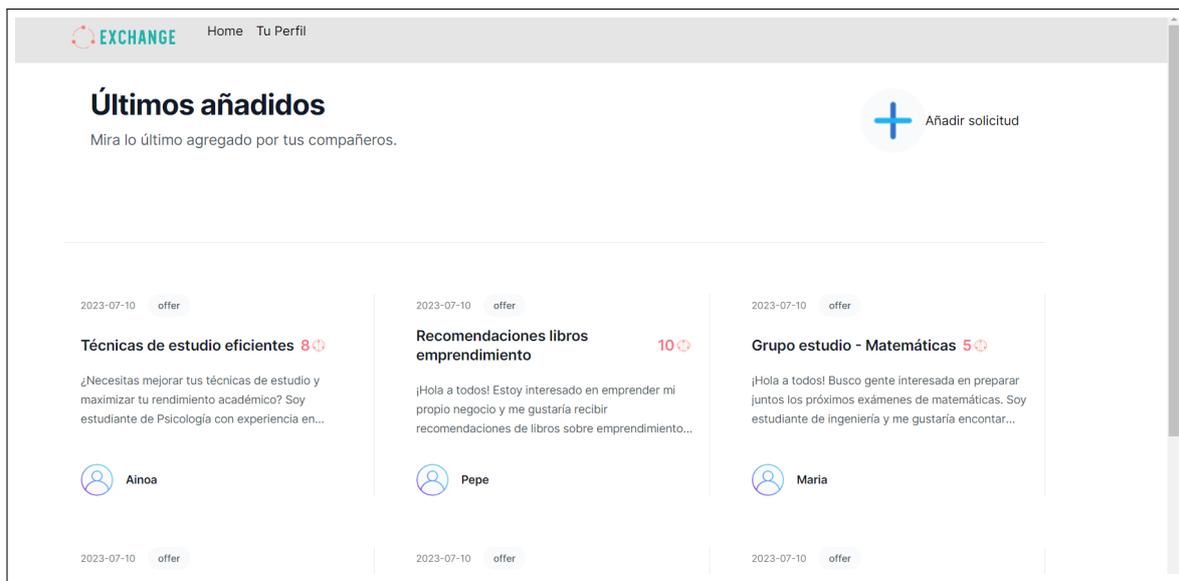
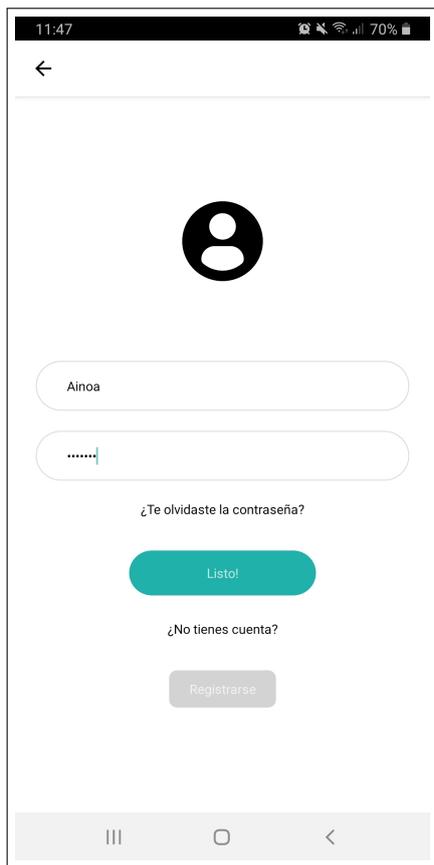


Figura 3.4: Home de la Web.

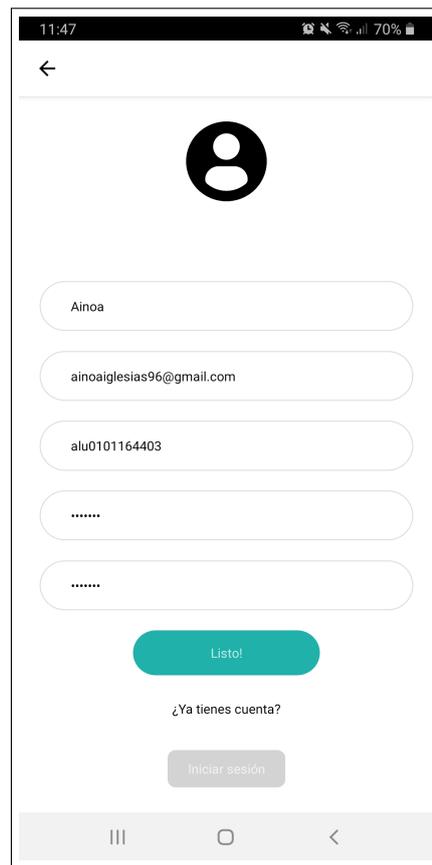
- Navbar Web: La web cuenta con una barra superior que se muestra en todas las pantallas. Esta barra de navegación varía según si hay un usuario con la sesión iniciada o no. Si es el caso, mostrará los botones para ir al home y al perfil, y en otro caso mostrará los botones para iniciar sesión y registrarse.

## ■ Pantallas comunes

- Login y Register: Estas pantallas tienen un formulario para registrar a un usuario por primera vez y para iniciar sesión en la aplicación. En esta parte se hace una comprobación para que las contraseñas introducidas deban tener al menos 4 caracteres, entre ellos una letra mayúscula, una letra minúscula y un número.



(a) Pantalla Inicio Sesión



(b) Pantalla de Registro.

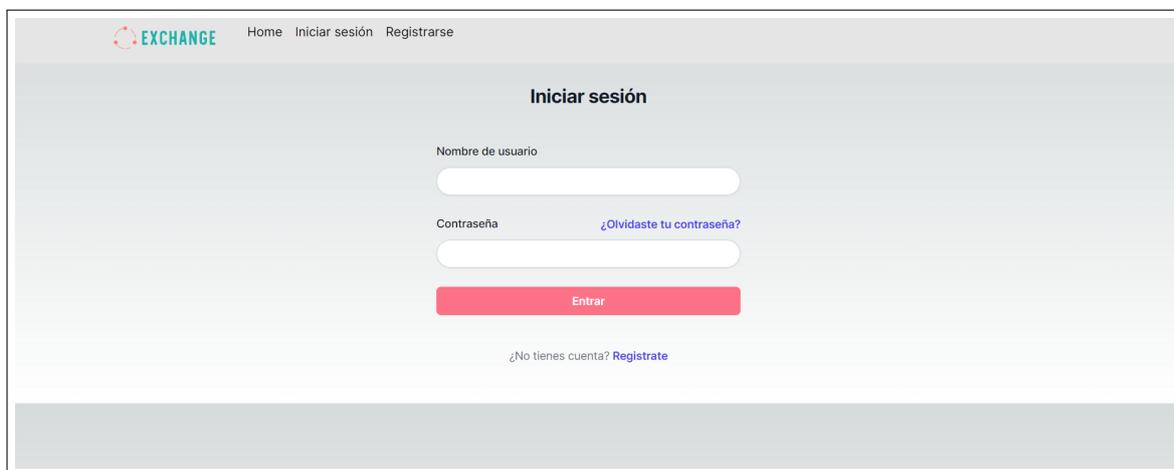


Figura 3.6: Pantalla Inicio Sesión Web.

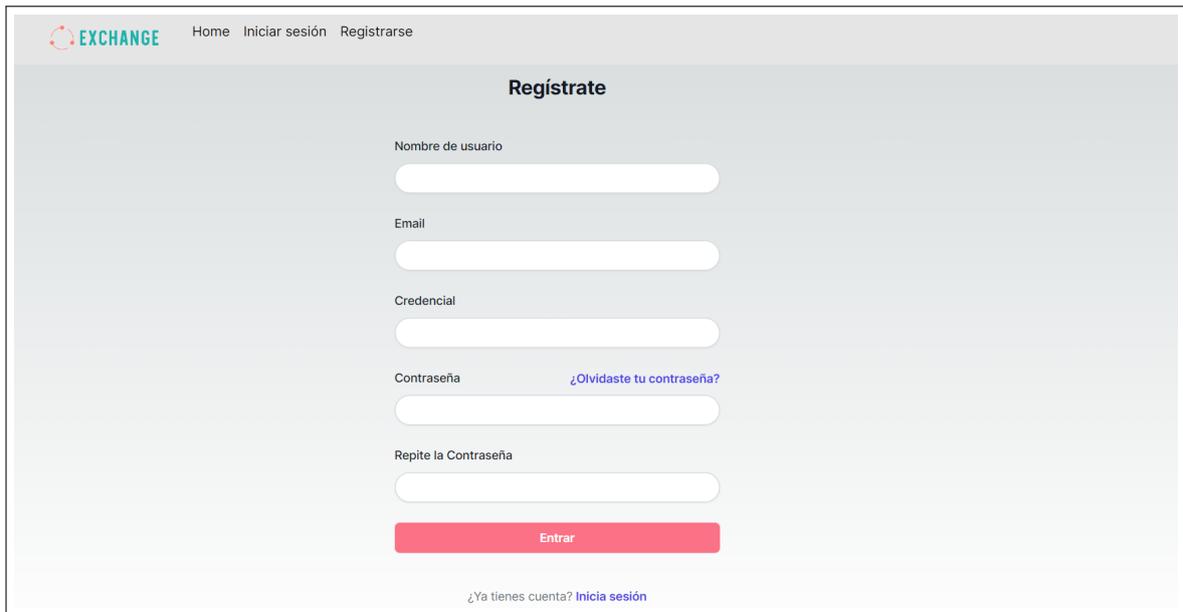


Figura 3.7: Pantalla de Registro Web.

- **Request:** Desde el home se puede pulsar en cualquier solicitud para verla en detalle y en esta sección hay un botón para aceptarla, si se pulsa, se descontará las monedas de la cartera del usuario que lo acepte, se añadirá las monedas en la cartera del dueño de la solicitud y se añadirá al historial de transacciones de cada usuario.

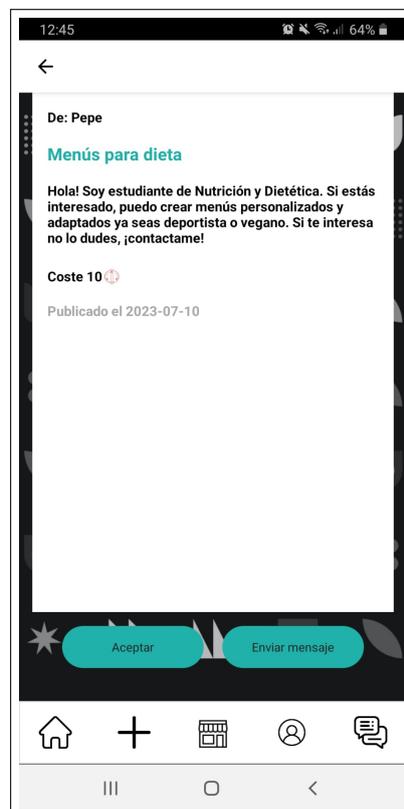


Figura 3.8: Información de una solicitud en la App.

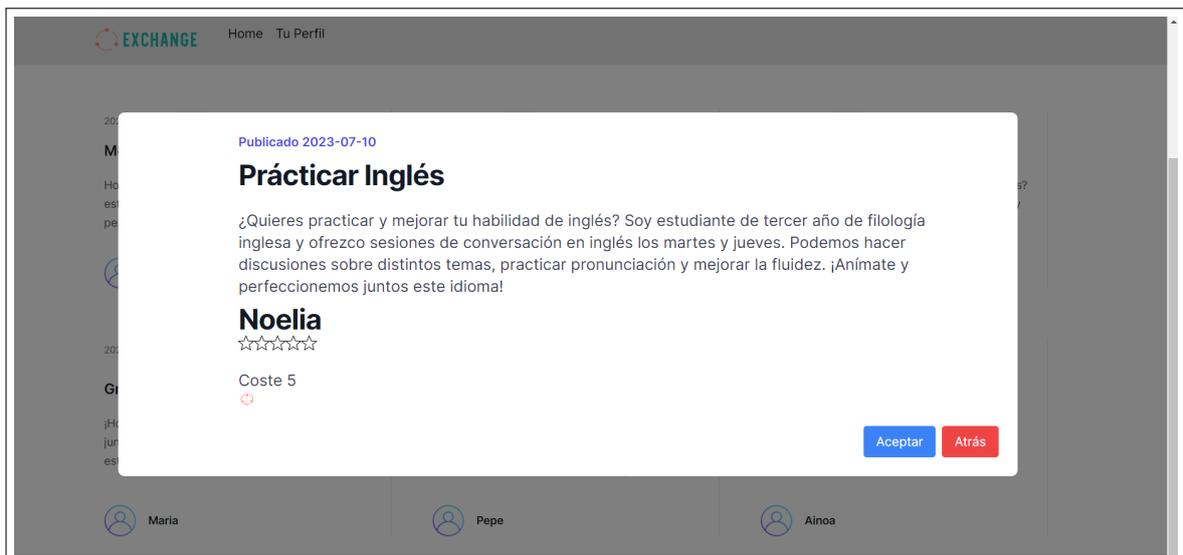


Figura 3.9: Información de una solicitud en la Web.

- Profile: Usando los botones del navbar se puede llegar a la pantalla de perfil de usuario, aquí se verá una pequeña información del usuario, un botón para ir a la cartera del usuario y una información sobre el numero de solicitudes que tiene creadas o activas y las que esperan respuesta por parte de otros usuarios. En el prototipo, si se pulsa en uno de ellos, se abrirá una lista con las solicitudes que tiene el usuario creadas y cada uno tendrá botones para eliminar, editar o bloquear la solicitud (estos botones no están en funcionamiento).

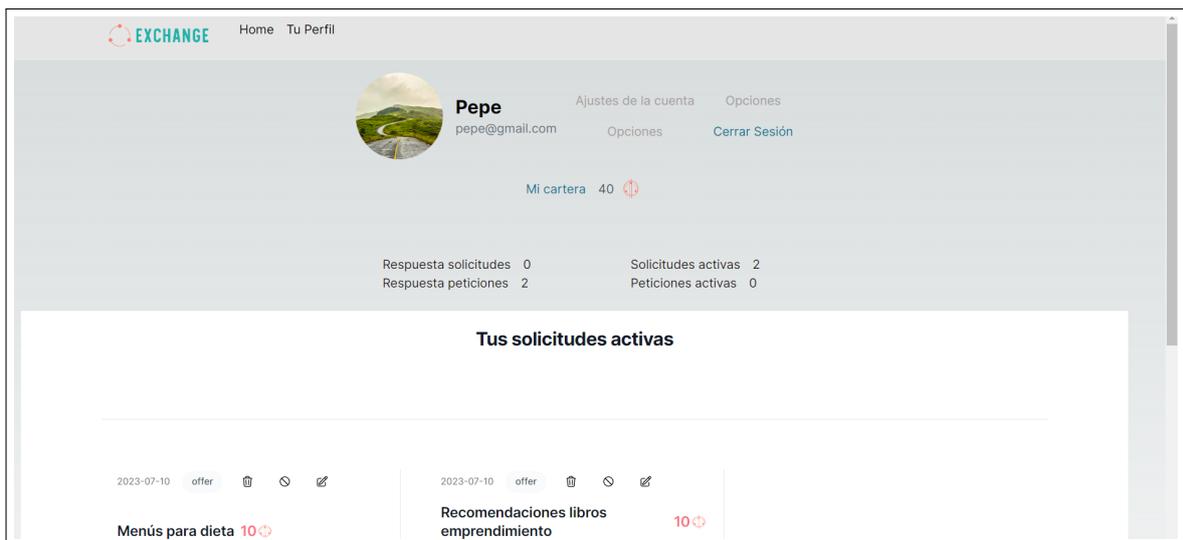


Figura 3.10: Pantalla de perfil de usuario en la Web.

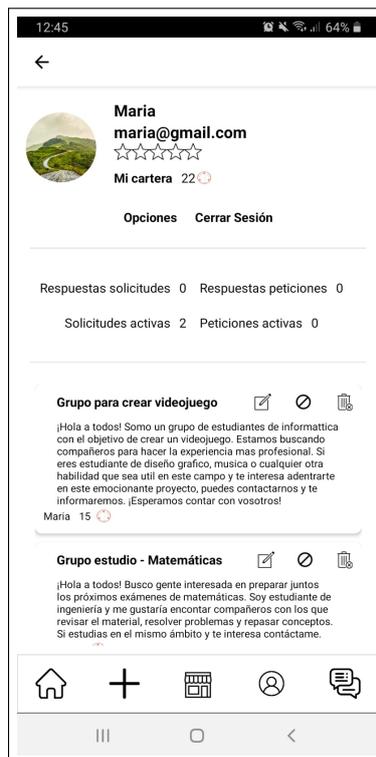


Figura 3.11: Pantalla de perfil de usuario en la App.

- **Wallet:** Desde el perfil de usuario se puede acceder a esta pantalla, en ella se verá una lista a modo de historial con las transacciones del usuario, verá el titulo de la solicitud, quien registró la solicitud, la fecha y si es un ingreso o pago.

EXCHANGE Home Tu Perfil

**Ainoa**  
ainoaglesias96@gmail.com

Mi cartera 23

Titulo solicitud y autor	Fecha	Coste
Técnicas de estudio eficientes <i>por María</i>	2023-07-10	+8
Practicar Inglés <i>por Noelia</i>	2023-07-10	-5
Menús para dieta <i>por Pepe</i>	2023-07-10	-10
Bienvenido. Estas son tus monedas iniciales. <i>por LogoApp</i>	2023-07-10	+30

Figura 3.12: Historial de transacciones en la Web.

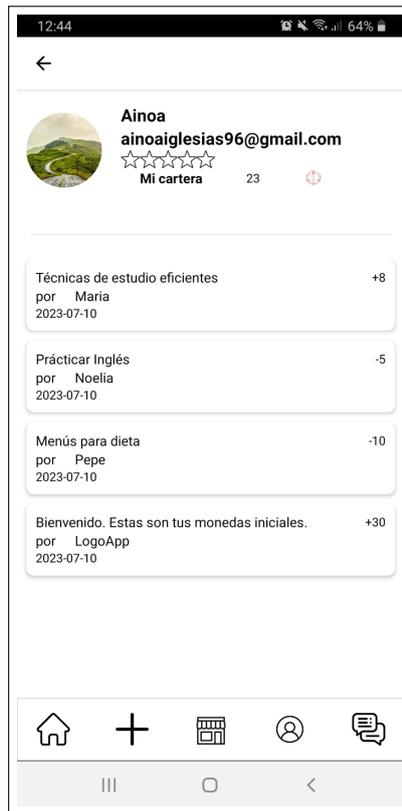


Figura 3.13: Historial de transacciones en la App.

- Formulario para solicitud/oferta: En el caso de la app. para acceder al formulario existe un botón en la barra inferior y en la web se encuentra un botón en el home que abre un modal con el formulario.

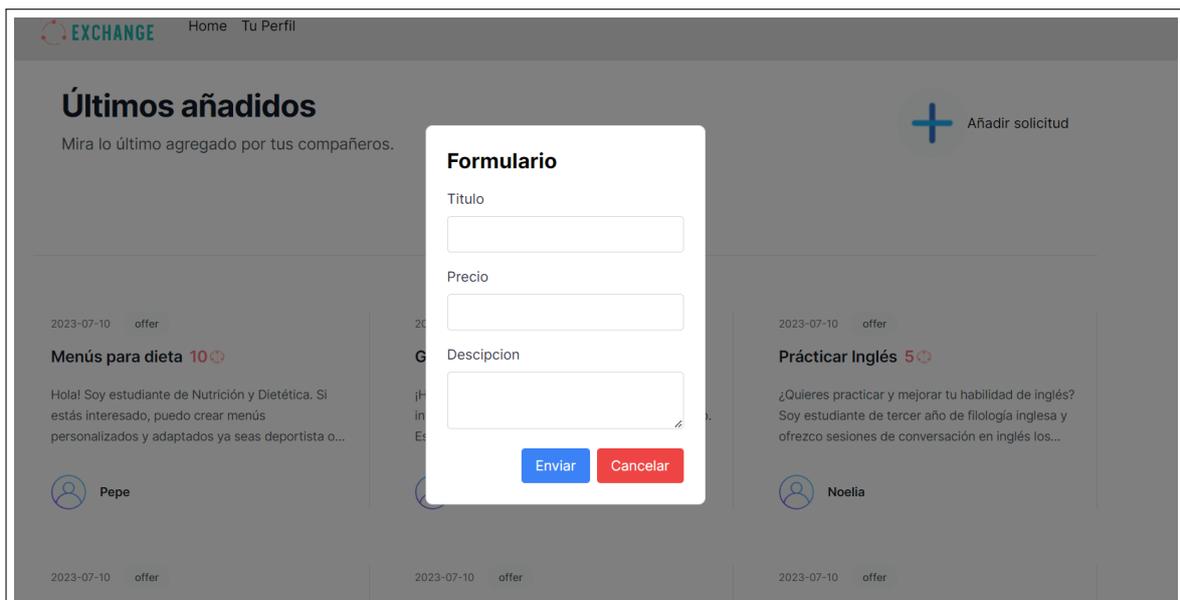


Figura 3.14: Formulario en la Web.

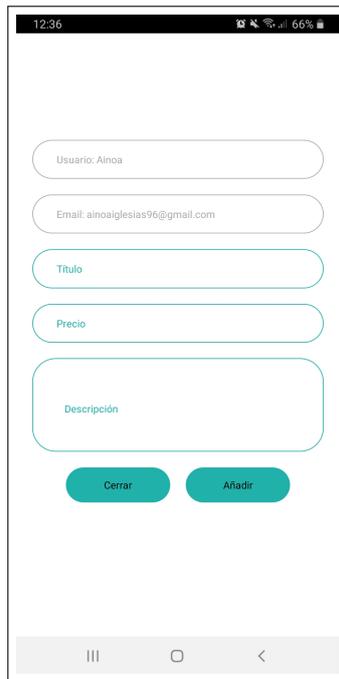


Figura 3.15: Formulario en la App.

### 3.3. Despliegue

- Backend: Para desplegar el backend lo primero fue registrarse en Render.com e integrarlo a en GitHub. Tras crear un nuevo proyecto en Render a partir del repositorio donde está el backend, se configuró las variables de entorno necesarias, en este caso la url de la base de datos, el puerto y la dirección IP. Una vez configurado y funcionando correctamente se puede acceder a través del enlace que ofrece la interfaz de Render.

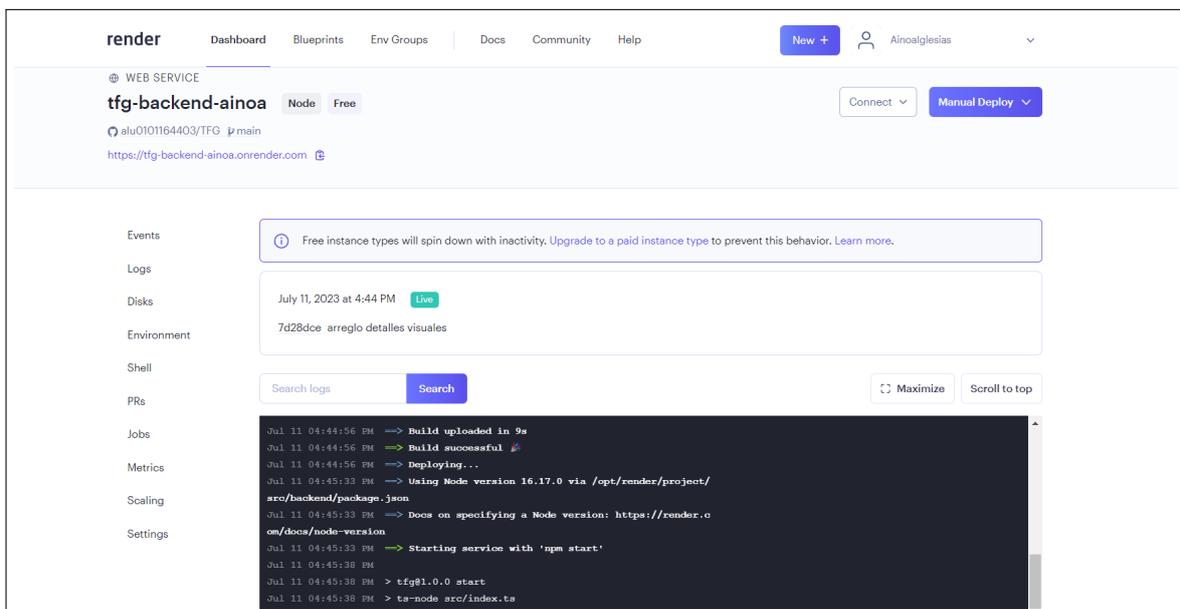


Figura 3.16: Interfaz de Render. *Render.com*

- **Web:** Al igual que para el backend, fue necesario registrarse en Vercel e integrarlo al repositorio en GitHub. Después, se puede crear un proyecto ligado a la carpeta del repositorio en el que está en código de la web. En este caso no fue necesario configurar nada ya que el enlace al backend en Render está incluido en el código. Como puede verse en la imagen, Vercel ofrece una serie de url con las que compartir la web.

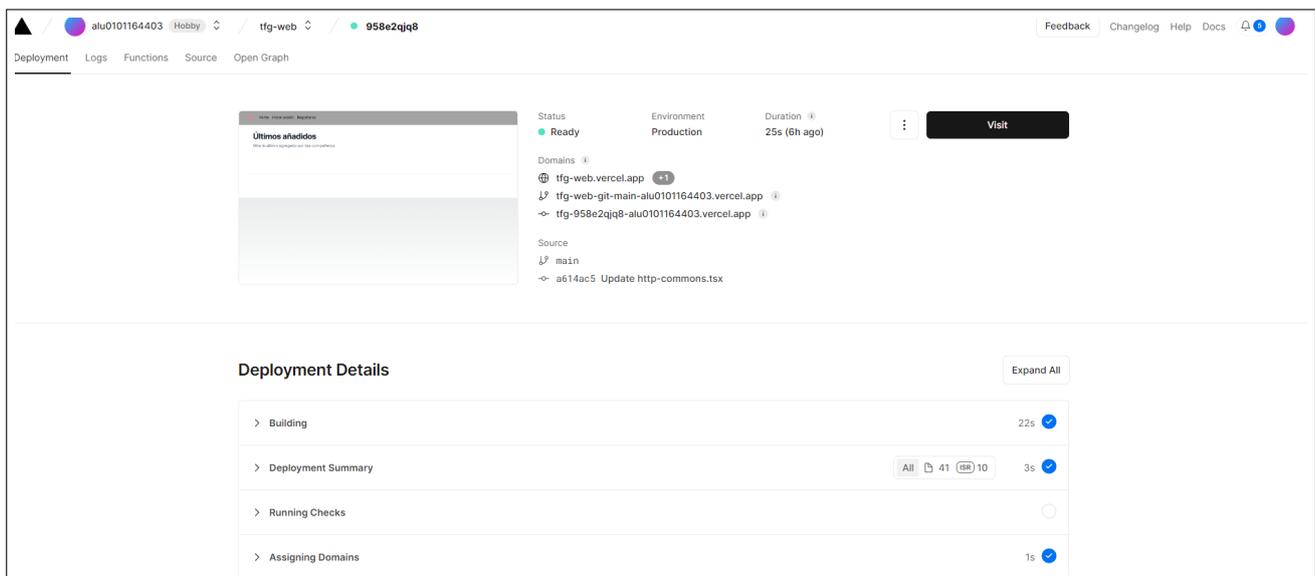


Figura 3.17: Interfaz de Vercel. *Vercel.com*

- **App:** Para compartir la aplicación Android se generó el APK a través del comando de React Native `'npx react-native run-android --variant=release'` con lo que se compila el proyecto y se genera un archivo APK en la carpeta `'/release'` del proyecto. Hay que tener en cuenta que esto genera un APK sin firmar por lo que <sup>1</sup>

<sup>1</sup>Puede ver el repositorio con el prototipo en el siguiente enlace: <https://github.com/alu0101164403/TFG>

# Capítulo 4

## Estudio de viabilidad económica

En este punto se hace un análisis para conocer la viabilidad del proyecto planteando el desarrollo del prototipo de manera profesional. Por este motivo, para conocer las características y el alcance de la aplicación profesional, es necesario saber en qué consistirá exactamente la aplicación, por ello el primer punto de este apartado hablará sobre las funcionalidades de la app.

### 4.1. Funcionalidades de la aplicación

El prototipo actual cuenta con una interfaz simple y unas funciones básicas, a esto se le podría añadir más características y para esto se ha planteado el uso de la aplicación para más tipos de comunidades además de la estudiantil pero con la misma base de intercambio de recursos y cooperación. Las funcionalidades serían las siguientes:

- Creación de una comunidad. La aplicación permite que un usuario o una organización cree una comunidad que puede ser pública o privada. En ambos casos habrá uno o más administrados encargados de gestionar los productos de la tienda entre otras cosas.
  - Pública: Cualquier usuario registrado puede entrar a esta comunidad. Puede añadirse limitaciones como el ámbito geográfico, el tipo de recursos a intercambiar o el tipo de usuarios a quien se dirige (estudiantes, manitas, madres/padres).
  - Privada: Puede servir para comunidades más cerradas como puede ser una empresa en la que se busque la participación y cooperación entre sus empleados. Cuenta con un sistema de solicitud/aceptación de usuarios.
- Límite de solicitudes/peticiones activas por usuario.
- Chat para comunicarse con otros usuarios.
- Chat privado para las comunidades.
- Sistema de validación para aceptar las solicitudes/peticiones (no haciendo un cobro directo como el prototipo actual).
- Funciones adicionales para las comunidades. Al crear una comunidad se dará la opción de añadir funciones existentes en la aplicación, por ejemplo un sistema de logros y metas.

- Al registrarse un usuario, en el home verá comunidades existentes para su zona geográfica y solicitudes/peticiones públicas.
- Límite al número de comunidades al que se puede pertenecer.
- Tienda. Los productos de la tienda estarán gestionados por los administradores de las comunidades. Existirá la opción de compra de monedas de la app pero limitando el número al mes por usuario.
- Sistema de contacto, preguntas y sugerencias.
- Sistema para que un usuario pueda tener contenido en la app (pdf, vídeos...) que pueda mostrar tras recibir las monedas.
- Pasarela de pago.

## 4.2. Desarrollo del proyecto

En esta sección se verán las fases y tareas necesarias para realizar el proyecto, estableciendo costes, recursos y la duración. Se pretende contestar a las preguntas ¿Cuál sería el coste del proyecto si se quiere realizar?, ¿Cuánto tiempo llevaría?, ¿Cuándo se recuperaría la inversión inicial? Haciendo uso de ProjectLibre<sup>1</sup> se ha creado un diagrama de Gantt<sup>2</sup> donde se muestra los tiempos y costes de cada fase:

- **Análisis:** Lo primero en un proyecto es realizar un estudio de mercado exhaustivo, en el que recopilar información sobre la competencia, las tendencias del mercado, estudiar a los posibles usuarios, sus necesidades y comportamientos. Analizar y definir los requisitos para la aplicación y los objetivos y planificar las tareas y costes del proyecto. También se establecerán las reuniones necesarias con el equipo.
- **Diseño:** En esta segunda fase, se estudiará y seleccionará las tecnologías, servicios y hardware que se usarán para llevar a cabo el proyecto. Se diseñará la arquitectura del software y la base de datos. Y se hará los mock-ups<sup>3</sup> de la interfaz de usuario.
- **Desarrollo:** En esta etapa se hace el desarrollo del software. Se implementa y gestiona la base de datos, creando los modelos de datos y las rutas necesarias para el funcionamiento esperado de la aplicación. Se implementa la autenticación, autorización, cookies... para la seguridad de la web y app y se desarrolla en el frontend la idea expuesta en los mock-ups. Además se harán los test unitarios de todo el proyecto y una persona se encargará de testear que todo funcione correctamente antes de publicar el trabajo.
- **Despliegue:** En esta última fase se selecciona los proveedores para los servidores, se hará las configuraciones necesarias y se desplegará la web y el backend. Por ultimo se publicará la app en la Play Store y se testeará que todo funcione correctamente.

---

<sup>1</sup>**ProjectLibre:** Software de código abierto para administración de proyectos.

<sup>2</sup>**Diagrama de Gantt:** Una herramienta gráfica para visualizar los tiempos previstos para tareas o actividades.

<sup>3</sup>**Mock-up:** Es el prototipo que se hace antes de desarrollar la aplicación o web. Muestra la idea, aspecto e interacciones que se espera obtener de la interfaz de usuario

Por último hay que tener en cuenta, que una vez puesto el proyecto en el mercado será necesario tener un mantenimiento todos los meses, ya sea para corregir errores, publicar actualizaciones, implementar nuevas funcionalidades y llevar un análisis sobre rendimiento, usabilidad y uso de la app.

	Nombre	Duracion	Inicio	Terminado	Costo
1	☐ Desarrollo del proyecto	149 days	6/07/23 8:00	30/01/24 ...	104216,00 €
2	☐ 1. Fase de análisis	51 days	6/07/23 8:00	14/09/23 ...	47384,00 €
3	1.1 Estudio de mercado y competenci	25 days	6/07/23 8:00	9/08/23 17:00	4600,00 €
4	1.2 Análisis requisitos	10 days	9/08/23 8:00	22/08/23 17...	4640,00 €
5	1.3 Definir objetivos	5 days	22/08/23 12:00	29/08/23 13...	2320,00 €
6	1.4 Reuniones(clientes/ equipo...)	51 days	6/07/23 8:00	14/09/23 17...	33864,00 €
7	1.5 Planificación de tareas y costes	7 days	5/09/23 16:00	14/09/23 16...	1960,00 €
8	☐ 2. Fase de diseño	21 days	14/09/23 8:00	12/10/23 ...	10560,00 €
9	2.1 Análisis y elección tecnologías a u	7 days	14/09/23 8:00	22/09/23 17...	3360,00 €
10	2.2 Diseño de la base de datos	5 days	22/09/23 8:00	28/09/23 17...	1000,00 €
11	2.3 Diseño de arquitectura	15 days	22/09/23 8:00	12/10/23 17...	3000,00 €
12	2.4 Diseño mockups y casos de uso	20 days	14/09/23 8:00	11/10/23 17...	3200,00 €
13	☐ 3. Fase de desarrollo	59 days	12/10/23 8:00	2/01/24 1...	40112,00 €
14	3.1 Creación base de datos	7 days	12/10/23 8:00	20/10/23 17...	1512,00 €
15	☐ 3.2 Backend	50 days	12/10/23 8:00	20/12/23 ...	11320,00 €
16	3.2.1 Crear modelos de BBDD	7,5 days	12/10/23 8:00	23/10/23 13...	2520,00 €
17	3.2.2 Creación servidor y API	15 days	15/10/23 8:00	3/11/23 17:00	5040,00 €
18	3.2.3 Implementar autenticación y	10 days	24/11/23 8:00	7/12/23 17:00	2160,00 €
19	3.2.4 Test Unitarios	10 days	7/12/23 8:00	20/12/23 17...	1600,00 €
20	☐ 3.3 Frontend Android	50 days	12/10/23 8:00	20/12/23 ...	11200,00 €
21	3.3.1 Crear paginas (home, login,	25 days	12/10/23 8:00	15/11/23 17...	8000,00 €
22	3.3.2 Integración con API	5 days	24/11/23 8:00	30/11/23 17...	1600,00 €
23	3.3.3 Test	10 days	7/12/23 8:00	20/12/23 17...	1600,00 €
24	☐ 3.4 Frontend Web	50 days	12/10/23 8:00	20/12/23 ...	11680,00 €
25	3.3.1 Crear paginas (home, login,	25 days	12/10/23 8:00	15/11/23 17...	8400,00 €
26	3.3.2 Añadir lógica para conexión	5 days	24/11/23 8:00	30/11/23 17...	1680,00 €
27	3.3.3 Test	10 days	7/12/23 8:00	20/12/23 17...	1600,00 €
28	3.5 Test de funcionamiento del softv	10 days	20/12/23 8:00	2/01/24 17:00	4400,00 €
29	☐ 4. Fase de despliegue	21 days	2/01/24 8:00	30/01/24 ...	6160,00 €
30	4.1 despliegue servidor y frontend	15 days	2/01/24 8:00	22/01/24 17...	4200,00 €
31	4.2 Subir a la pley store	7 days	22/01/24 8:00	30/01/24 17...	1960,00 €

Figura 4.1: Datos usados para el calculo de tiempo y costes. *ProjectLibre*

#### 4.2.1. Costo y duración

Como resultado de lo anterior, se ha estimado que el proyecto supone un tiempo aproximado de 149 días laborales lo que equivale a 7 meses. En cuanto al coste, obtenido con los salarios de los empleados seleccionados, se situaría en unos 104.216 euros, sin

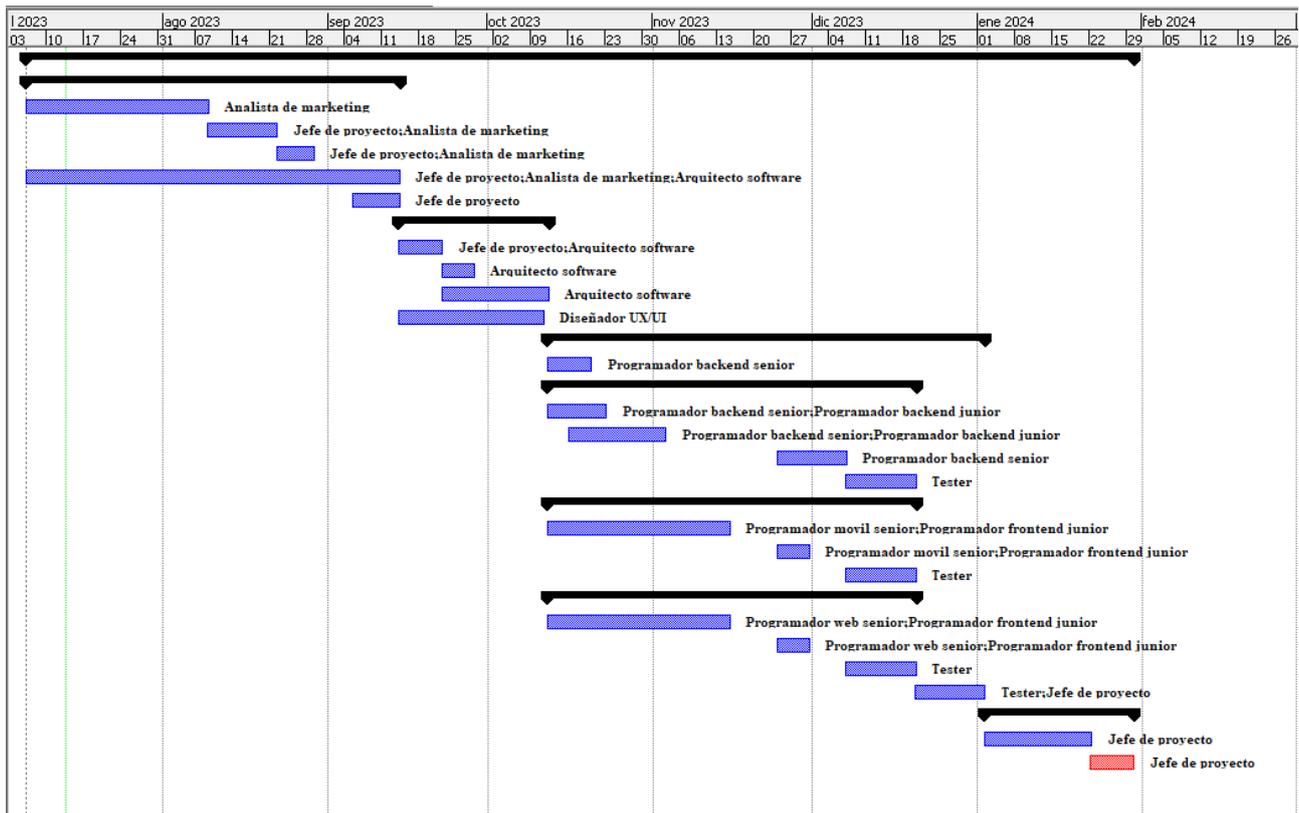


Figura 4.2: Diagrama de Gantt creado. *ProjectLibre*

embargo hay que tener en cuenta más factores, como el coste de los servicios para el despliegue o el certificado SSL para proporcionar seguridad a la web. El coste de los servidores se ha obtenido de otros sitios ya que los planes para equipos de Vercel y Render no tienen precios asignados fijos.

- Servidor backend: Digital Ocean ofrece gran variedad de opciones y precios, se ha estimado que el coste mensual de mantener el backend será unos 170 euros.
- Servidor web: Para la web se estima unos 20 euros mensuales mantener el dominio.
- Certificado SSL: Hay una gran variedad de precios para obtener un certificado SSL y se ha optado por uno de coste mensual de 40 euros al año.
- Google Play: Para poder subir la aplicación a la tienda de Google Play es necesario pagar unos 25 euros una sola vez.
- Mantenimiento: Se mantendrá dos programadores full stack para encargarse de los errores y de crear nuevas funciones según la demanda de los usuarios. Esto supondrá un coste de unos 5000 euros al mes.

## 4.3. Modelo de comercialización

### 4.3.1. Fuentes de ingresos

Una vez terminada la aplicación, se pueden barajar varias posibilidades para obtener los beneficios necesarios para recuperar la inversión realizada[1]:

- **Vender la App:** Una opción es vender la aplicación en tiendas digitales para móviles como Google Play Store, o buscar compradores que puedan estar interesados, teniendo en cuenta que este trabajo se especificó que era dirigida a estudiantes universitarios, se podría vender a universidades. Sin embargo, esta aplicación puede usarse en numerosos ámbitos y por tanto se podrían buscar otro tipo de compradores.
- **Publicidad:** Tanto en las webs como en las aplicaciones móviles es muy común ver anuncios con los que se monetizan. Una opción sería hacer uso de Google Ads y Google AdMob[2]<sup>4</sup> para generar ingresos sin ahuyentar a los usuarios. Esto también ayudaría a tener una mejor posición de las búsquedas en la Play Store y en Google.
- **Planes de suscripción:** se pueden establecer un sistema de suscripción mensual, es un sistema que vemos en numerosas webs y aplicaciones de todo tipo que se ajustan a cada usuario. En este caso se podría contar con la opción gratuita, en la cual se visualizarán anuncios y habrá ciertas limitaciones como puede ser el número de ofertas que un usuario puede registrar a la vez o al mes.
- **Compras dentro de la app:** Como se ha visto antes, una posible función que puede tener la aplicación es la posibilidad de comprar monedas con dinero real, lo cual supondría un beneficio. De la misma manera se pueden ir añadiendo mas productos en la tienda.

Teniendo en cuenta las características de la app y los modelos más usados[28] se ha optado por el modelo de suscripción y publicidad integrada. Este modelo da la oportunidad a los consumidores de usar y probar la aplicación gratuitamente por lo que el público al que se llega es mas amplio. Además, al contar con una tienda, se obtienen ingresos también con las compras in-app. Las opciones de suscripción existentes serán las siguientes:

- **Free:** Este es el modelo básico, generará sus ingresos por medio de la publicidad[29] (banners, anuncios intersticiales y vídeos) y las compras dentro de la app. Estos usuarios tendrán limitaciones en cuanto al número de solicitudes/peticiones que pueden tener activas y el números de comunidades al que pertenecer.
- **Premium:** Este modelo da ventajas a los usuarios como eliminar la publicidad y quitar las limitaciones del modelo Free. La cuota de esta suscripción será de 4 euros.
- **Basic Comunity:** para usuarios que quieran crear una comunidad, podrán escoger el modelo de comunidad estándar o añadir un número limitado de las características que otorga la aplicación. La cuota de esta suscripción será de 12 euros.
- **Profesional Comunity:** para comunidades creadas por empresas, esta opción ofrece la ventaja de poder solicitar funcionalidades personalizadas y añadir sin limites las que ya existan en la app. La cuota de esta suscripción será de 60 euros.

#### 4.3.2. Estudio del retorno de inversión (ROI)

Para obtener el punto de retorno de la inversión inicial se ha tenido en cuenta los gastos de mantenimiento estimados tras publicar la app que consistirán en tener dos

---

<sup>4</sup>**Google Ads y Google AdMob:** Son un servicio de Google que se usa para ofrecer publicidad patrocinada.

programadores full stack para corregir fallos y añadir funciones o aspectos nuevos según las demandas de los usuarios, además de las cuotas mensuales y anuales de servicios como el certificado ssl y los servidores.

Para obtener los ingresos, se ha hecho una estimación del número de usuarios haciendo compras in-app de un 5 % [28] de los usuarios. Y para los ingresos generados por los anuncios la estimación se ha basado en las publicaciones de distintas webs y vídeos.

Por último, es necesario saber cuánto crecerá el número de usuarios, y si lo hará [20], tras publicarse la aplicación. Para conocer si tendrá éxito la aplicación con estas características, se ha buscado posibles competidores y existe una gran variedad de sitios con el propósito de intercambio de recursos o conocimientos y/o con el fin o la opción de crear comunidades virtuales que tienen bastante éxito. Algunos ejemplos son MeetUp [21], We are Testers [3], Bunz [5], Grupos de Facebook o Mighty Network [23].

Tras el análisis, la estimación obtenida es que se empezará a recuperar la inversión en la semana 88, lo que equivale a 21 meses.<sup>5</sup>

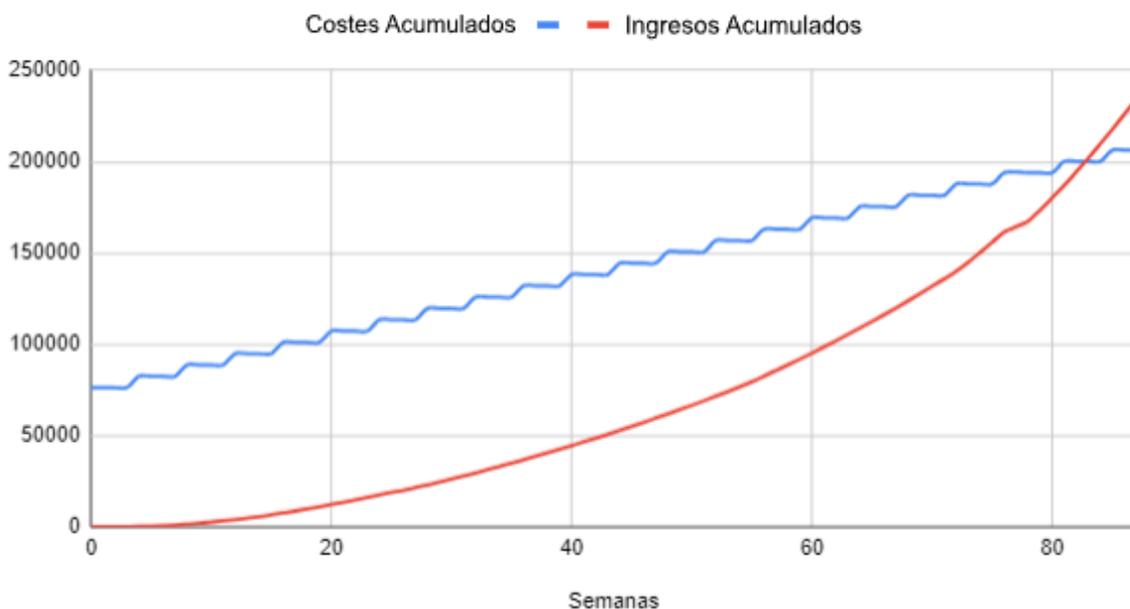


Figura 4.3: Gráfico con ingresos acumulados y gastos acumulados.

TRIMESTRE	Plan Free	Plan Premiun	Basic Comunity	Profesional Comunity	Ingresos	Gasto	Gasto acum	Ingresos acum	ROI
1	222	11	8	4	3739,56	18570	91874	3739,56	88364,44
2	490	9	10	6	12491,4	18570	102254	13328,08	89155,92
3	700	15	15	7	17076,48	18570	117824	28510,36	89543,64
4	895	20	18	8	20615,48	18570	133394	48344,24	85279,76
5	1399	25	18	8	26576,76	18570	149004	73672,36	75561,64
6	1836	28	16	9	36734,72	18570	164574	106667,52	58136,48
7	3870	35	20	14	61034,36	18570	180144	158870,68	21503,32
8	4200	37	20	16	19367,4	18570	185334	190162,08	-4598,08

Figura 4.4: Datos trimestrales obtenidos en el ROI.

<sup>5</sup>Puede ver un excel con el análisis en este enlace: <https://docs.google.com/spreadsheets/d/1d2UljEXnkvUWHmh90tvHZuKXtNudKPfrdNxlCYE7W9A/edit#gid=0>

# Capítulo 5

## Conclusiones y líneas futuras

Como conclusión, el prototipo desarrollado[16] ha logrado cumplir los objetivos establecidos en términos de funcionalidad y diseño, consiguiendo una plataforma donde los usuarios pueden compartir sus conocimientos, habilidades y recursos a través de una moneda virtual local. Hay que tener en cuenta que es un prototipo y que cuenta con un diseño y funcionalidades simples.

Sin embargo, hay espacio para expandir y mejorar la aplicación en futuras iteraciones. Una posible línea de desarrollo es la incorporación de más servicios y características útiles para los estudiantes. Esto podría incluir la creación de un registro de logros y metas, donde los usuarios deben cumplir ciertos objetivos académicos o extracurriculares para ganar monedas adicionales. Esta funcionalidad no solo promovería la participación y el compromiso de los estudiantes, sino que también les brindaría un incentivo adicional para alcanzar sus metas educativas.

Además, se podría considerar la implementación de un sistema de validación y aprobación de solicitudes. Actualmente, las solicitudes de intercambio se realizan directamente entre los usuarios, pero se podría introducir un proceso en el cual el propietario de la solicitud debe revisar y aprobar la propuesta antes de que se complete el intercambio. Esto permitiría una mayor comunicación y coordinación entre los usuarios, brindándoles la oportunidad de discutir los detalles y establecer acuerdos antes de comprometerse con el intercambio.

Otra área de mejora potencial es la integración de un chat o sistema de mensajería dentro de la aplicación. Esto facilitaría la comunicación entre los usuarios, lo que resultaría especialmente útil para aclarar dudas, negociar los términos de los intercambios y establecer una mayor confianza entre ellos.

Es importante destacar que este concepto de aplicación basada en una moneda virtual local y el intercambio de recursos puede extenderse a otros ámbitos y comunidades, tal como se planteaba en el apartado 4.1. Por ejemplo, podría adaptarse a entornos empresariales, donde los empleados podrían intercambiar habilidades y conocimientos relevantes para sus funciones. También podría ser implementado en comunidades locales, promoviendo el intercambio de servicios y productos entre los residentes para fortalecer la economía local y fomentar la colaboración entre vecinos.

# Capítulo 6

## Summary and Conclusions

In conclusion, the developed prototype has managed to meet the established objectives in terms of functionality and design, achieving a platform where users can share their knowledge, skills and resources through a local virtual currency. However, it has to be taken into account that it is a prototype and has a simple design and functionalities.

However, there is room to expand and improve the application in future iterations. One possible line of development is the incorporation of more useful services and features for students. This could include the creation of an achievement and goal tracker, where users must meet certain academic or extracurricular goals to earn additional coins. This functionality would not only promote student participation and engagement, but also provide students with an additional incentive to achieve their educational goals.

In addition, the implementation of an application validation and approval system could be considered. Currently, exchange requests are made directly between users, but a process could be introduced in which the owner of the request must review and approve the proposal before the exchange is completed. This would allow for greater communication and coordination between users, giving them the opportunity to discuss details and establish agreements before committing to the exchange.

Another area of potential improvement is the integration of a chat or messaging system within the application. This would facilitate communication between users, which would be particularly useful for clarifying doubts, negotiating the terms of exchanges and establishing greater trust between them.

Importantly, this concept of an application based on a local virtual currency and the exchange of resources can be extended to other areas and communities. For example, it could be adapted to business environments, where employees could exchange skills and knowledge relevant to their roles. It could also be implemented in local communities, promoting the exchange of services and products among residents to strengthen the local economy and foster collaboration among neighbours.

# Capítulo 7

## Presupuesto

Se ha realizado un presupuesto[10] del desarrollo de este trabajo de fin de grado. Para ello, se ha hecho una aproximación del tiempo para cada tarea, dando un total de 220 horas y usando el sueldo medio de un programador full stack en España, de 17 euros por hora. Esto da un presupuesto total de 3740 euros.

Tarea	Tiempo	Coste total
Análisis y estudio previo	40	680
Desarrollo memoria	30	510
Desarrollo Backend	40	680
Desarrollo Frontend App	40	680
Desarrollo Frontend Web	25	425
Documentación	15	255
Despliegue y testeo	15	255
Otros (reuniones, seminarios...)	15	255
<b>Total</b>	<b>220</b>	<b>3740</b>

Figura 7.1: Presupuesto del TFG.

# Bibliografía

- [1] Actualizatec. Cómo monetizar una app: las 11 mejores estrategias. <https://actualizatec.com/blog/como-monetizar-una-app/>, 2023.
- [2] Google AdMob. Descubra cuánto podría ganar con admob. <https://admob.google.com/intl/es-419/home/>.
- [3] We are Testers. Comunidades online. <https://www.wearetesters.com/comunidades-online?gclid=CjwKCAjwwb6lBhBJEiwAbuVUSjxpmSgQPX9Jb2QtFBBf5P29yRZZfMBwE>, 2023.
- [4] Diputació Barcelona. Guía metológica: Implementación de monedas locales desde la administración pública: una herramienta para la dinamización comercial local. <https://www.diba.cat/es/web/comerc/moneda-local>, 2022.
- [5] Bunz. Página web principal de bunz. <https://bunz.com/>, 2023.
- [6] Banco de España. Monedas virtuales y locales: las paramonedas, ¿nuevas formas de dinero? <https://repositorio.bde.es/handle/123456789/11245>, 2018.
- [7] Bando de España Eurosistema. Estudio sobre el hábito del uso del efectivo en 2022. <https://www.bde.es/wbe/es/areas-actuacion/billetes-monedas/estudios-publicaciones-relacionadas-efectivo/estudio-sobre-habitos-uso-efectivo/>, 2023.
- [8] Ayuntamiento de Lebrija. La nueva moneda de lebrija. <https://web.clickoin.com/elio/>, 2020.
- [9] Universidad Politécnica de Madrid. Fundamentos de la gamificación. <https://oa.upm.es/35517/>, 2015.
- [10] Ainoa Iglesias Dasilva DRIVE. Si quiere comprobar los documentos excel o el apk del proyecto. <https://drive.google.com/drive/u/0/folders/1kAE1mu9h-JaMbSh8kNaxy4LD47Ys6vzC>, 2023.
- [11] ESSC. Los videojuegos y las cuasi-monedas digitales. <https://www.economiassc.com/post/los-videojuegos-y-las-cuasi-monedas-digitales>, 2020.
- [12] Express. Página oficial de express. <https://expressjs.com/>, 2023.
- [13] Sector financiero de PWC. Los medio de pago, un paisaje en movimiento. <https://www.pwc.es/es/publicaciones/financiero-seguros/medios-pago-paisaje-movimiento.html>, 2023.
- [14] Flutter. Página web oficial de flutter. <https://flutter.dev/>, 2023.

- [15] Ionic Framework. Documentación oficial de ionic. <https://ionicframework.com/docs>, 2023.
- [16] GitHub. Repositorio del prototipo realizado en este tfg. <https://github.com/alu0101164403/TFG>.
- [17] Next JS. Documentación oficial de next.js. <https://nextjs.org/docs>, 2023.
- [18] React Js. Documentación oficial de react.js. <https://legacy.reactjs.org/docs/getting-started.html>, 2023.
- [19] Vue JS. Documentación oficial de vue.js. <https://vuejs.org/guide/introduction.html>, 2023.
- [20] LinkedIn. Why are in-app communities the fastest growing trend? <https://www.linkedin.com/pulse/why-in-app-communities-fastest-growing-trend-plussaw-1e/>, 2021.
- [21] MeetUp. La plataforma de las personas: donde los intereses se convierten en amistades. <https://www.meetup.com/es-ES/>, 2002.
- [22] React Native. Documentación oficial de react native. <https://reactnative.dev/docs/getting-started>, 2023.
- [23] Mighty Network. Página web principal de mighty network. <https://www.mightynetworks.com/?via=viktoriiia&gclid=CjwKCAjwwb6lBhBJEiwAbuVUShjsyHKTZGm17yF1ijRTB7vIda8II8nzjjPIwLR0tn0is90k4s2EWhBwE>, 2002.
- [24] Node.js. Documentación oficial de node.js. <https://nodejs.org/en/docs>, 2023.
- [25] Stack Overflow. Developer survey 2023. <https://survey.stackoverflow.co/2023/>, 2023.
- [26] PlastiCoin. Plasticoín, el valor de reciclar. <https://www.plasticoín.com.uy/>, 2023.
- [27] rec.barcelona. Rec, citizen currency final report. [https://rec.barcelona/wp-content/uploads/2020/02/informe-final\\_EN\\_compressed.pdf](https://rec.barcelona/wp-content/uploads/2020/02/informe-final_EN_compressed.pdf), 2019.
- [28] Statista. Most popular app monetization methods by publishers from the united states as of march 2023. <https://www.statista.com/statistics/1119916/app-monetization-methods-united-states-app-publishers/>.
- [29] Statista. Distribución porcentual del poder de monetización de la publicidad integrada en aplicaciones móviles (in-apps) de enero a noviembre de 2017. <https://es.statista.com/estadisticas/601047/cuota-de-poder-de-monetizacion-de-los-formatos-de-in-app-a-nivel-mundial/>, 2017.
- [30] Statista. Portal de statista. <https://es.statista.com/>, 2023.
- [31] Stack Overflow Survey. Página web de stack overflor survey 2023. <https://survey.stackoverflow.co/2023/>.
- [32] Wuolah. Descarga y venta de apuntes. <https://wuolah.com/>, 2023.