



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

## Trabajo de Fin de Grado

---

**FreeMyPark: Aplicación para la gestión de  
aparcamiento**

*FreeMyPark: Parking management application*

Ángel Augusto Capote Pérez

---

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 Ingeniería Inofrmática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Alejandro Pérez Nava**, con N.I.F. 43.821.179-S profesor asociado de Universidad adscrito al Departamento de Ingeniería Inofrmática y de Sistemas de la Universidad de La Laguna, como cotutor

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

Que la presente memoria titulada:

*"Título del Trabajo"*

ha sido realizada bajo su dirección por D. **Francisco Javier Rodríguez González**, con N.I.F. 43.618.712-V.

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

# Agradecimientos

A mis padres, por el apoyo y amor que me han dado a lo largo de mi vida  
que me ha ayudado a poder lograr mis objetivos.

A mis hermanos, por nunca dejar de creer en mí.

A mi pareja, por el apoyo incondicional en los momentos difíciles.

A mis amigos, que me han ayudado como refugio para desconectar cuando  
requería.

# Licencia



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

## **Resumen**

*Conseguir aparcamiento a pie de calle en las ciudades de España a día de hoy es un duro reto que implica una inversión de tiempo y combustible, tirando de suerte para conseguir una plaza disponible.*

*Vemos una necesidad en el hecho de que los usuarios desean tener una alternativa para encontrar aparcamiento a pie de calle y no acabar cediendo a los aparcamientos privados, donde los precios son cada vez más altos.*

*Es objetivo del presente trabajo de fin de grado diseñar, crear y comercializar una aplicación móvil multiplataforma, disponible tanto para iOS como para Android, que gestione puestos de aparcamiento a pie de calle, donde los usuarios pueden ofrecer la plaza que van a desocupar, o aceptar una plaza que va a quedar disponible, a cambio de una modesta tarifa preestablecida.*

*La implementación de esta aplicación repercutirá en el bienestar de la comunidad, fomentando la colaboración mutua en busca de un beneficio común.*

**Palabras clave:** aparcamiento, España, vehículo, pie de calle, precio, tiempo, aplicación móvil, comunidad.

## **Abstract**

*Finding a place to park on the street in Spanish cities today is a tough challenge that requires an investment of time and fuel, as well as a lot of luck to find an available parking space.*

*We see a need in the fact that users want to have an alternative to find parking on the street and not end up giving in to private parking, where prices are increasingly higher.*

*The objective of this thesis is to design, develop and market a cross-platform mobile application, available for both iOS and Android, that manages on-street parking spaces, where users can offer the space they are going to vacate, or accept a space that becomes available, in exchange for a modest pre-established fee.*

*The implementation of this application will have an impact on the well-being of the community, promoting mutual cooperation in search of a common benefit.*

**Keywords:** park, Spanish cities, time, mobile application, community.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Definición del problema . . . . .	2
1.2. Justificación . . . . .	3
1.3. Objetivos del proyecto . . . . .	5
1.4. Estado del arte . . . . .	6
1.4.1. Comparativa con aplicaciones similares . . . . .	6
1.4.2. Leyes y Regulaciones . . . . .	9
1.5. Fases del proyecto . . . . .	10
1.5.1. Análisis y planificación . . . . .	10
1.5.2. Diseño . . . . .	10
1.5.3. Desarrollo . . . . .	10
1.5.4. Pruebas . . . . .	11
1.5.5. Estudio de la viabilidad económica . . . . .	11
<b>2. Estudio previo</b>	<b>12</b>
2.1. Análisis y definición de requisitos . . . . .	12
2.2. Arquitectura del sistema . . . . .	13
2.2.1. Arquitectura Hardware . . . . .	13
2.2.2. Arquitectura Software . . . . .	13
2.3. Selección de tecnologías . . . . .	14
2.3.1. Aplicación . . . . .	14
2.3.2. GraphQL API . . . . .	17
2.3.3. Base de Datos . . . . .	17
2.3.4. Otras tecnologías . . . . .	18
<b>3. Aplicación</b>	<b>20</b>
3.1. Funcionalidades . . . . .	20
3.1.1. Pantalla de Bienvenida . . . . .	20
3.1.2. Registro . . . . .	21
3.1.3. Recuperar Contraseña . . . . .	21
3.1.4. Pantalla Inicial . . . . .	22
3.1.5. Pantalla de Añadir plaza . . . . .	23
3.1.6. Pantalla Aceptar puesto . . . . .	25
3.1.7. Pantalla Añadir fondos . . . . .	26
3.1.8. Pantalla perfil . . . . .	27
<b>4. GraphQL API</b>	<b>28</b>

<b>5. Base de datos</b>	<b>30</b>
5.1. Creación de la base de datos . . . . .	31
<b>6. Estudio de viabilidad económica del desarrollo de una Aplicación Profesional</b>	<b>32</b>
6.1. Características destacadas de la aplicación profesional . . . . .	32
6.2. Desarrollo del proyecto . . . . .	33
6.3. Modelo de Comercialización . . . . .	38
6.3.1. Estudio del Retorno de la Inversión . . . . .	39
<b>7. Conclusiones y líneas futuras</b>	<b>41</b>
<b>8. Summary and Conclusions</b>	<b>42</b>
<b>9. Presupuesto</b>	<b>43</b>

# Índice de Figuras

1.1. Gráfico Aumento de las temperaturas en España . . . . .	3
1.2. Gráfico Aumento poblacional en España . . . . .	4
1.3. Logo de ElParking . . . . .	6
1.4. Logo de TelPark . . . . .	7
1.5. Logo de ParClick . . . . .	7
1.6. Logo de WeSmartPark . . . . .	8
1.7. Fases del proyecto . . . . .	10
2.1. Diagrama de Caso de Uso I . . . . .	13
2.2. Diagrama de Caso de Uso II . . . . .	13
2.3. Logo de Flutter . . . . .	14
2.4. Logo de React Native . . . . .	14
2.5. Gráfico popularidad lenguajes de programación según Tiobe . . . . .	15
2.6. Logo de Visual Studio Code . . . . .	16
2.7. Logo de Simulator . . . . .	16
2.8. Logo de GraphQL . . . . .	17
2.9. Logo de DynamoDB . . . . .	18
2.10 Logo de Github . . . . .	18
2.11 Logo de Trello . . . . .	18
2.12 Logo de ProjectLibre . . . . .	19
2.13 Logo de Overleaf . . . . .	19
3.1. Pantalla de Bienvenida . . . . .	20
3.2. Pantalla de Registro . . . . .	21
3.3. Pantalla de Recuperar Contraseña . . . . .	22
3.4. Pantalla de Inicio . . . . .	23
3.5. Pantalla de Añadir plaza . . . . .	23
3.6. Pantalla de Espera . . . . .	24
3.7. Pantalla de Puesto Publicado . . . . .	25
3.8. Pantalla de puesto aceptado . . . . .	26
3.9. Pantalla de Añadir fondos . . . . .	26
3.10 Pantalla de Perfil . . . . .	27
4.1. Schema de la Base de Datos . . . . .	28
5.1. Logo de AWS Amplify . . . . .	30
5.2. Schema de la base de datos . . . . .	31
6.1. Tareas de cada Fase . . . . .	34
6.2. Recursos Materiales . . . . .	35

6.3. Primera parte Diagrama de Gantt . . . . .	37
6.4. Segunda parte Diagrama de Gantt . . . . .	37
6.5. Tabla ingresos mensuales . . . . .	39
6.6. Gráfica de Gasto Acumulado y ROI . . . . .	40

# Índice de Tablas

1.1. Comparativa aplicaciones . . . . .	8
2.1. Flutter vs ReactNative(10) . . . . .	15
6.1. Tablas recursos personal . . . . .	36
9.1. Tabla de Costes asociados a la elaboración del proyecto . . . . .	43

# Capítulo 1

## Introducción

Ante la evidente dificultad que representa hoy día encontrar libre una plaza para aparcar en las calles de las ciudades de España, se propone una estrategia colaborativa que favorecerá la consecución de plazas disponibles en la zona, en el momento que se necesite.

Esto es una realidad factible con la descarga de una aplicación diseñada para hacer posible la confluencia de los usuarios, ofertantes y demandantes de las plazas de aparcamiento a pie de calle.

De esta manera, se plantea una solución a la masificación de vehículos que circulan por las calles en busca de aparcamiento. Mediante esta aplicación móvil, se ahorrará tiempo y dinero de una manera efectiva a la hora de aparcar en las zonas concurridas de nuestras ciudades.

La presente memoria se estructura en nueve capítulos. En el primer y segundo capítulos se presenta el proyecto, se definen los antecedentes, estado del arte, requisitos, objetivos y alcance del proyecto. Además, se realiza un estudio de las herramientas y tecnologías a implementar. El tercer, cuarto y quinto capítulos están enfocados al desarrollo del prototipo de la aplicación, donde se explican las funcionalidades de la aplicación, la API<sup>1</sup> y la base de datos implementada. El sexto capítulo aborda la comercialización del proyecto, llevando a cabo un estudio de viabilidad económica donde se plantea una estimación de los costes y tiempos para desarrollar una aplicación profesional. Los capítulos séptimo y octavo exponen las conclusiones del proyecto así como una línea futura y el presupuesto con los costes asociados a su desarrollo. Y, por último, el noveno capítulo, donde se indica el presupuesto del coste de tiempo de la elaboración del proyecto.

---

<sup>1</sup>**API:** mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos.

## **1.1. Definición del problema**

Generalmente, el proceso para buscar aparcamiento consiste en acudir directamente a los parkings privados disponibles a lo largo de la ciudad. Pocas personas se plantean la posibilidad de encontrar una plaza libre en la calle, teniendo en cuenta que el tiempo y el recorrido que muy probablemente implicará lograrlo, les hará renunciar a la idea de probar suerte en la calle y acudir directamente al parking privado más cercano. Un gasto de dinero tanto en combustible como el coste de estos aparcamientos, que al final del mes sorprenderá a los usuarios.

## 1.2. Justificación

Son varios los motivos que han impulsado el desarrollo de este proyecto, los cuales se han dividido en dos puntos que se explican a continuación.

### 1. Impacto medioambiental.

Es una realidad la preocupación generalizada a nivel mundial sobre los impactos del cambio climático en aspectos que restan calidad de vida a las poblaciones. No pretendiendo entrar en detalles sobre las extensas consecuencias derivadas del cambio climático<sup>2</sup>, las condiciones en las que se desarrolla la presente propuesta tienen como base los siguientes dos factores medioambientales.

**El aumento de la temperatura en España.** La temperatura media en España (1) ha aumentado alrededor de 1.7°C desde la época preindustrial. Se puede apreciar en el siguiente diagrama de Hawkins:

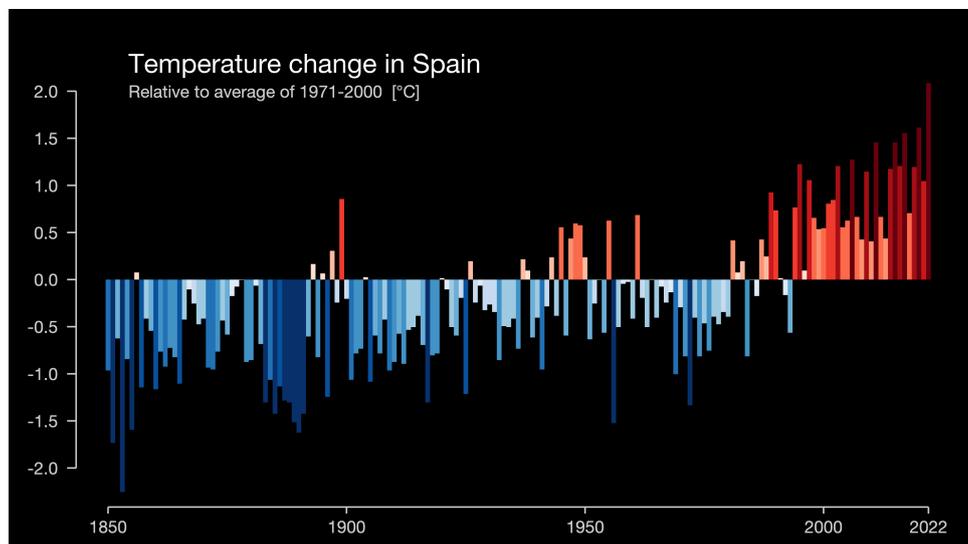


Figura 1.1: Gráfico Aumento de las temperaturas en España

En el diagrama se representa la evolución de las temperaturas medias anuales y se puede observar la tendencia ascendente, más marcada a partir de la década de los setenta y el incremento intenso en la última década mostrada, lógicamente coincidente con los años más cálidos registrados en el presente siglo.

**El empeoramiento de la calidad del aire.** El cambio climático es un factor determinante en el empeoramiento de la calidad del aire. El año 2022 fue el más seco y cálido en España en los últimos 60 años. El factor esencial del repunte de la contaminación atmosférica ha sido, en gran medida, el aumento del tráfico motorizado<sup>3</sup>.

Según Ecologistas en Acción<sup>4</sup> sobre la calidad del aire, de acuerdo a los datos de 2022, se desprende que los niveles de contaminación del año pasado en las ciudades han expuesto a la población a unos niveles insalubres, tomando como referencia los valores máximos de contaminación recomendados por la Organización Mundial de

<sup>2</sup>**Cambio climático:** cambios a largo plazo de las temperaturas y los patrones climáticos.

<sup>3</sup>**tráfico motorizado:** todo tipo de vehículo traccionado por motor.

<sup>4</sup>**Ecologistas en Acción:** confederación de grupos ecologistas de España.

la Salud (OMS).

La política española de adaptación al cambio climático, a través del Plan Nacional de Adaptación al Cambio Climático (PNACC)(2) 2021-2030 en el que se aprueba la Emergencia Climática y Ambiental, sigue líneas de trabajo, donde el tráfico es una de las más importantes materias a tratar.

Teniendo en cuenta que la cantidad de emisiones de CO2 aumentan en la medida en que los vehículos se ven obligados a circular en busca de plazas disponibles para aparcar a pie de calle, nos lleva a entender que estas emisiones se verían disminuidas si la tarea de aparcar fuera más simple y organizada. Por lo tanto, el uso de esta aplicación mejoraría la calidad del aire al contribuir con la disminución del tráfico.

## 2. Aumento poblacional

Según el informe de la empresa Statista<sup>5</sup> sobre el crecimiento poblacional en España (3) para el periodo 2008 a 2022, se puede apreciar un aumento poblacional significativo en los últimos años con una tendencia ascendente.

El aumento poblacional supone el incremento de vehículos privados en circulación debido a que el transporte público se hace insuficiente ante la creciente demanda.

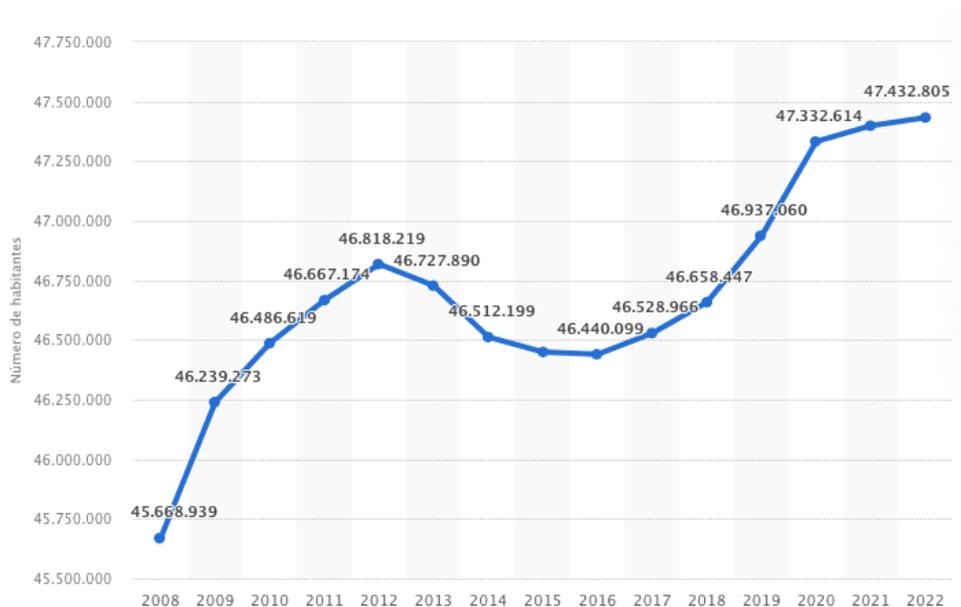


Figura 1.2: Gráfico Aumento poblacional en España

Lo que más preocupa a los ciudadanos españoles respecto a la calidad de vida en las ciudades, según encuestas realizadas por la Dirección General de Transparencia y Calidad en colaboración con la empresa Sigma Dos<sup>6</sup>, es el tráfico, la limpieza, la contaminación del aire y la sanidad.

<sup>5</sup>**Statista**: es un portal de estadística en línea alemán que ofrece datos relevantes procedentes de estudios de mercado.

<sup>6</sup>**Sigma Dos**: empresa española de estudios de mercado y demoscopia.

## 1.3. Objetivos del proyecto

Este estudio tiene como propósito responder a la interrogante de si es posible desarrollar un prototipo funcional que dé solución al problema de aparcamiento en las ciudades, y si además es viable económicamente.

Para lograr estos propósitos, se han definido los objetivos principales de este proyecto:

1. Diseñar e implementar un prototipo funcional de la aplicación.
2. Realizar un Estudio de Viabilidad Económica para el desarrollo de una aplicación profesional.

De modo que, para lograr estos objetivos principales, se realizarán las siguientes tareas:

- Definir los requisitos y el estado del arte.
- Analizar las herramientas y tecnologías disponibles y la posterior selección de las más adecuadas.
- Estudiar y elegir los frameworks<sup>7</sup>, lenguaje de programación y librerías a implementar.
- Diseñar y crear una interfaz de usuario.
- Diseñar y crear una base de datos.
- Diseñar y crear una API que gestione las consultas.
- Realizar un estudio de viabilidad económica del proyecto mediante Diagrama de Gantt<sup>8</sup> y la estimación ROI<sup>9</sup>.

---

<sup>7</sup>**Framework:** marco o esquema de trabajo utilizado por programadores para realizar el desarrollo de software.

<sup>8</sup>**Diagrama de Gantt:** herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas a lo largo de un tiempo determinado.

<sup>9</sup>**ROI:** cálculo del retorno de la inversión.

## 1.4. Estado del arte

### 1.4.1. Comparativa con aplicaciones similares

Para llevar a cabo el estado del arte, este proyecto se ha centrado en el estudio del mercado y la identificación de posibles competidores en el sector.

A continuación, se muestran las aplicaciones existentes en el mercado junto a sus principales características.

#### **ElParking**

Aplicación que permite buscar, reservar y pagar por espacios de estacionamiento en diferentes ubicaciones de España. Proporciona información en tiempo real sobre la disponibilidad en los aparcamientos y dispone de una opción para realizar el pago directamente desde la app.

Cuenta con otras funcionalidades tales como pago de zona azul, zona verde y otras zonas reguladas a través de la app, sin necesidad de ir al parquímetro, además de información relativa a la gasolinera más cercana.



Figura 1.3: Logo de ElParking

## **TelPark**

Esta aplicación permite buscar, reservar y pagar por espacios de estacionamiento en varias ciudades de España. Proporciona información en tiempo real y permite realizar el pago directamente desde la app.

Permite recibir notificaciones y recordatorios de tiempo de aparcamiento y ofrece la posibilidad de extender el tiempo de aparcamiento en caso que sea necesario.



Figura 1.4: Logo de TelPark

## **ParClick**

App que permite buscar, reservar y pagar por espacios de estacionamiento. Cuenta con más de 1400 aparcamientos situados en diferentes ubicaciones de España y otros países europeos como Francia, Italia, Portugal, entre otros.

Ofrece información en tiempo real incluyendo tarifas y disponibilidad. A diferencia del resto, ofrece descuentos y promociones especiales en algunos estacionamientos a la vez que tiene una opción de cancelación flexible.



Figura 1.5: Logo de ParClick

## **WeSmartPark**

Aplicación que permite buscar, reservar y pagar por espacios de estacionamiento en las ciudades de Madrid y Barcelona.

La aplicación conecta personas que cuentan con una plaza de garaje privado libre con aquellos que buscan aparcamiento. Las plazas de los propietarios son alquiladas por minutos obteniendo un beneficio tres veces mayor que alquilando por el método

tradicional.

Proporciona información en tiempo real, permite el pago directo desde la app y tiene servicios adicionales como el alquiler de plazas de garaje a largo plazo.



Figura 1.6: Logo de WeSmartPark

Todas las aplicaciones citadas anteriormente tienen varios puntos en común, recogidos en la siguiente tabla comparativa. Se ha añadido el prototipo objeto de este proyecto:

	<b>elParking</b>	<b>TelPark</b>	<b>ParClick</b>	<b>WeSmartPark</b>	<b>Prototipo</b>
Búsqueda y reserva	Sí	Sí	Sí	Sí	Sí
Disponibilidad	Sí	Sí	Sí	Sí	Sí
Notificaciones	No	Sí	No	No	No
Extensión de tiempo	Sí	Sí	No	No	Infinito
Descuentos	No	No	Sí	No	No
Cancelación flexible	No	No	Sí	No	No
Cobertura	España	España	Europa	España	España
Servicios extra	No	No	No	Sí	No
Parking a pie de calle	No	No	No	No	Sí
Comunidad	No	No	No	Sí	Sí

Tabla 1.1: Comparativa aplicaciones

En general, todas estas aplicaciones ofrecen características similares como la búsqueda, reserva y pago de espacios de estacionamiento, así como información en tiempo real sobre la disponibilidad. Sin embargo, las diferencias pueden radicar en la cobertura geográfica, descuentos/promociones especiales, opciones de cancelación flexible y servicios adicionales ofrecidos.

La gran diferencia con la aplicación a desarrollar radica en el origen de la plaza de garaje a obtener, ya que en todas las aplicaciones existentes son de aparcamientos privados o plazas de garajes privadas. En cambio, la aplicación que se propone ofrece plazas disponibles a pie de calle.

## 1.4.2. Leyes y Regulaciones

Para desarrollar una aplicación de esta índole, se debe cumplir con una serie de leyes y regulaciones.

- **Protección de Datos Personales.** Se debe cumplir la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales (4) (LOPDGDD) y el Reglamento General de Protección de Datos (RGPD) de la Unión Europea. Debe obtenerse el consentimiento explícito de los usuarios para recopilar, almacenar y procesar sus datos personales.
- **Ley de Servicios de la Sociedad de la Información y Comercio Electrónico(5) (LSSI-CE).** Esta ley establece los requisitos sobre la información que debes proporcionar a los usuarios, siendo estos los términos y condiciones de uso de la aplicación y la política de privacidad.
- **Ley de Propiedad Intelectual.** Ley que protege los derechos de propiedad intelectual<sup>10</sup>, como el software a utilizar en la aplicación.

Se debe tomar en cuenta que lo mencionado anteriormente son leyes de carácter general, por lo tanto, al desarrollar la aplicación, habrá también normativas especiales de acuerdo a la comunidad autónoma donde se implemente la aplicación.

Es por ello que se contará con un Especialista en Cumplimiento Normativo para garantizar el correcto cumplimiento de las leyes y regulaciones.

---

<sup>10</sup>**Propiedad Intelectual:** protección del producto del intelecto humano, sea en los campos científicos literarios, artísticos o industriales.

## 1.5. Fases del proyecto

Para la consecución del objetivo principal descrito anteriormente, se han definido cinco fases en las que se estructura el desarrollo del proyecto.



Figura 1.7: Fases del proyecto

### 1.5.1. Análisis y planificación

En esta fase inicial, se lleva a cabo un estudio previo con el objetivo de comprender el estado del arte y poder definir los requisitos del proyecto y las necesidades de los usuarios.

- Definición de los requisitos que debe cumplir la aplicación.
- Definición de objetivos.
- Búsqueda de herramientas y tecnologías más adecuadas para desarrollar el prototipo.

### 1.5.2. Diseño

Se crea un diseño detallado del prototipo de la aplicación partiendo de lo definido en la fase anterior:

- Diseño arquitectura general.
- Diseño interfaz de usuario.
- Diseño estructura de la base de datos.

### 1.5.3. Desarrollo

Se desarrollan los diseños generados en la fase anterior con las herramientas y librerías elegidas en la fase inicial.

- Desarrollo front-end.
- Desarrollo back-end.
- Configuración e implementación de la API GraphQL.

#### **1.5.4. Pruebas**

En esta fase se realiza una serie de pruebas sobre el código desarrollado en busca de posibles errores o problemas de rendimiento. Se prueba en diferentes escenarios para garantizar su correcto funcionamiento en diversos dispositivos y plataformas. Se verifica que:

- Los componentes de la aplicación funcionan correctamente.
- La API responde correctamente a las solicitudes de la aplicación.
- La base de datos almacena los datos como es esperado.

#### **1.5.5. Estudio de la viabilidad económica**

En esta última fase, se llevará a cabo una estimación de la factibilidad de desarrollar un proyecto profesional.

- Desarrollo del proyecto
- Modelo de comercialización
- Estimación de costes mediante el Diagrama de Gantt
- Estimación de ingresos y ROI

# Capítulo 2

## Estudio previo

Para desarrollar un buen producto, es de gran importancia saber qué se quiere obtener, las herramientas y tecnologías que existen y ver lo que se adapta a las necesidades del proyecto. Es por ello que este capítulo define los requisitos del proyecto, la arquitectura general del sistema y las tecnologías seleccionadas para implementarlo.

### 2.1. Análisis y definición de requisitos

El análisis de requisitos es esencial para establecer una visión clara del proyecto, guiar el diseño y desarrollo del software y establecer criterios de aceptación y prueba. Esto garantiza que el software creado cumpla con las expectativas y satisfaga exitosamente las necesidades del usuario final.

Para el desarrollo de la aplicación, se han definido los siguientes requisitos a tener en cuenta a la hora de elegir las herramientas y tecnologías:

- La aplicación debe contar con dos usuarios:
  - Oferente de plaza.
  - Demandante de la plaza ofertada.
- Debe contar con geolocalización para determinar la ubicación en tiempo real de los usuarios en todo momento.
- Debe tener un sistema de mapas para orientarse en el entorno geográfico.
- Debe tener un sistema de navegación para guiar a los usuarios.
- Debe existir una moneda que funcione como método de pago para reservar las plazas.
- Debe permitir la compra de saldo(puntos) mediante pagos en la aplicación a través de tarjetas de crédito, paypal, entre otras.

A partir de lo recogido anteriormente, se han desarrollado los diagramas de caso de uso(6) de la aplicación:

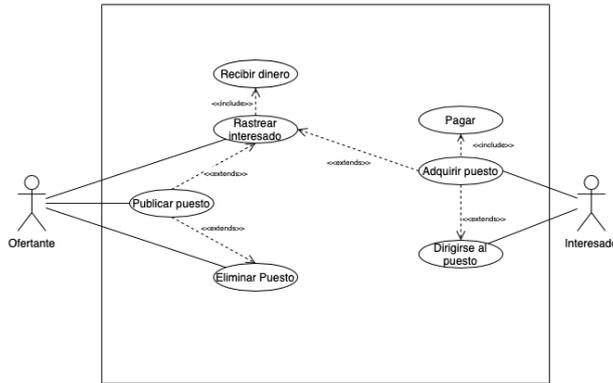


Figura 2.1: Diagrama de Caso de Uso I

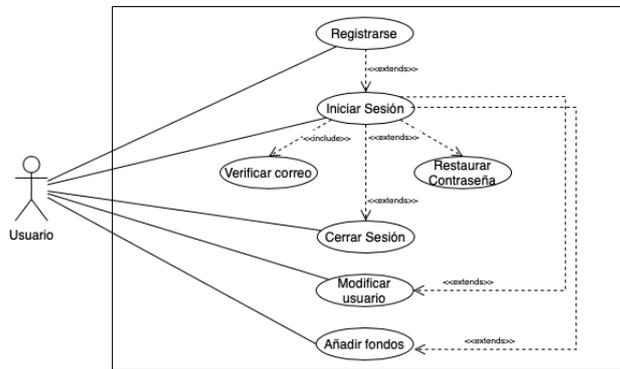


Figura 2.2: Diagrama de Caso de Uso II

## 2.2. Arquitectura del sistema

### 2.2.1. Arquitectura Hardware

Para que la aplicación funcione se requiere de un dispositivo móvil inteligente que disponga de conexión internet y geolocalización.

El servidor está almacenado en la nube, por lo que no se requiere de elementos físicos para almacenarlo, ya que es proporcionado por el proveedor del servicio.

### 2.2.2. Arquitectura Software

La arquitectura de la aplicación se basa en una arquitectura Cliente-Servidor(7), en ella se diferencian las siguientes partes:

- **Frontend.** Se encarga de la interfaz de usuario, incluye el diseño y la navegación de la aplicación.
- **API.** Se encarga de la comunicación e interacción entre el frontend y el backend.
- **Backend.** Se encarga de la lógica y el funcionamiento interno de la aplicación.

El flujo de consulta a la base de datos consiste en que el frontend utiliza la API para enviar solicitudes al backend y obtener los datos necesarios.

## 2.3. Selección de tecnologías

Previamente identificada la arquitectura, se comentarán los recursos y tecnologías implementados para realizar este proyecto.

### 2.3.1. Aplicación

El proyecto es una aplicación nativa multiplataforma, es decir, apta tanto para iOS como para Android. De esta manera, se abarcaría la casi totalidad de sistemas operativos que hay en el mercado.

#### Framework y Lenguaje de programación

A la hora de desarrollar aplicaciones nativas multiplataforma existen dos opciones: **Flutter** y **ReactNative**. A continuación, se explican brevemente y se justifica cuál ha sido la elegida para la elaboración del proyecto.

Flutter(9) es un SDK<sup>1</sup> desarrollado por Google para crear aplicaciones móviles tanto para Android como para iOS.



Figura 2.3: Logo de Flutter

ReactNative(8) es un framework de código abierto creado por Facebook para desarrollar aplicaciones para Android, iOS, macOS, Android TV, Web y Windows. Tiene como finalidad sustituir los componentes HTML de React por los que ofrece React Native obteniendo una aplicación real nativa.

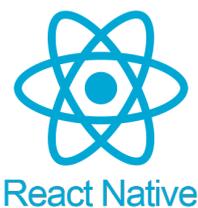


Figura 2.4: Logo de React Native

---

<sup>1</sup>SDK: kit de desarrollo de software.

Tabla comparativa entre Flutter y ReactNative:

	<b>Flutter</b>	<b>ReactNative</b>
Fecha lanzamiento	Mayo 2017	Marzo 2015
Creador	Google	Facebook
Lenguaje de programación	Dart	JavaScript
Plataformas soportadas	Android, iOS	Android, iOS, Web
Código abierto	Si	Si
Documentación	Extensa	Extensa
Comunidad	pequeña, en crecimiento	Extensa
Aplicaciones que la usan	Google, Alibaba, Tencent	Facebook, Instagram, Uber, Salesforce

Tabla 2.1: Flutter vs ReactNative(10)

Una vez analizadas las dos opciones en detalle, se ha optado por usar **ReactNative** para la elaboración de este proyecto por los siguientes motivos:

- Usa JavaScript, uno de los lenguajes de programación más usados actualmente, según se muestra en la gráfica2.5.
- Posee librerías de geolocalización necesarias para el proyecto.
- Mayor comunidad, por lo que existe mayor documentación y foros de ayuda a los que poder consultar.

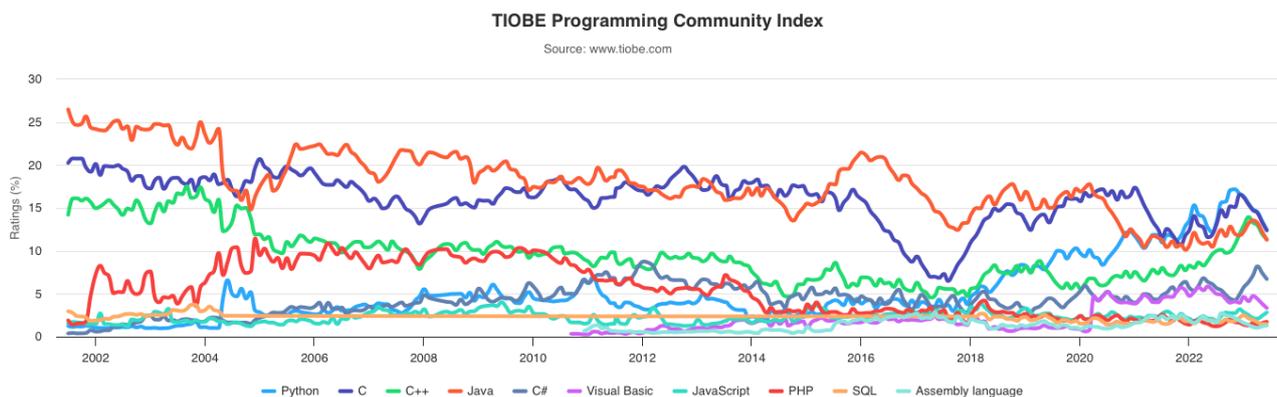


Figura 2.5: Gráfico popularidad lenguajes de programación según Tiobe

## Librerías

Para elaborar la aplicación hemos dispuesto de las siguientes librerías:

- **react, react-native.** Librerías básicas de React.
- **react-native-elements** Librería con objetos básicos, como los iconos.
- **@react-navigation/native.** Librería para la navegación entre pantallas.
- **react-native-maps.** Librería de mapas iterativos.
- **aws-amplify.** Librería para la integración de AWS Amplify.
- **moment.** Librería para el manejo de fechas y tiempos.

## Entorno de desarrollo

- **Visual Studio Code.** Editor de código fuente que ofrece a los desarrolladores una experiencia de desarrollo ágil y altamente personalizable. Posee una amplia gama de características y extensiones que lo hacen muy popular entre los programadores.



Figura 2.6: Logo de Visual Studio Code

- **Simulator.** Aplicación de desarrollo para iOS que permite a los desarrolladores probar y depurar sus aplicaciones en un entorno virtual. Permite verificar el rendimiento y la funcionalidad de las aplicaciones sin necesidad de un dispositivo físico emulando dispositivos iOS. Es una herramienta útil para el desarrollo y prueba de aplicaciones iOS antes de su implementación en dispositivos reales.



Figura 2.7: Logo de Simulator

### 2.3.2. GraphQL API

GraphQL API es una interfaz de programación de aplicaciones (API) que utiliza el lenguaje de consulta GraphQL, lo que permite a los clientes solicitar y obtener datos de manera eficiente y precisa.

Los clientes pueden enviar consultas estructuradas que describan los datos que desean obtener y las relaciones entre ellos utilizando GraphQL. La API de GraphQL interpreta estas consultas y responde con los datos requeridos, lo que reduce la cantidad de datos transferidos y mejora el rendimiento de las aplicaciones.

Debido a que ofrece una forma flexible y eficiente de obtener datos de los servidores, GraphQL API se ha vuelto popular en el desarrollo de aplicaciones web y móviles. Es utilizado por varias empresas y plataformas, como Facebook, GitHub y Shopify, para brindar a sus desarrolladores y clientes APIs poderosas y personalizables.

La principal diferencia entre las API GraphQL y las API REST es que GraphQL ofrece una capa de abstracción adicional, lo que permite al cliente especificar los datos precisos que necesita y obtener mayor flexibilidad y eficiencia. Por otro lado, las API REST definen la estructura de los datos que se envían en respuesta a una solicitud. Además, GraphQL API tiene un único endpoint a diferencia de API REST que puede tener múltiples.

Es por esto que se ha implementado una GraphQL API para la elaboración del proyecto.



Figura 2.8: Logo de GraphQL

### 2.3.3. Base de Datos

Para la elaboración del proyecto se ha elegido AWS Amplify, que integra un servicio propio de AWS llamado Amazon DynamoDB.

#### Sistema gestor de base de datos

AWS Amplify integra Amazon DynamoDB(12) como sistema gestor. Es un servicio de base de datos NoSQL<sup>2</sup> administrado con escalabilidad automática, rendimiento rápido y predecible y alta disponibilidad. Se usa fundamentalmente para aplicaciones web y móviles que requieren un almacenamiento de datos escalable y flexible.

---

<sup>2</sup>**base de datos NoSQL**: base de datos no relacional



Figura 2.9: Logo de DynamoDB

### 2.3.4. Otras tecnologías

#### Sistema de control de versiones

Para el control de las versiones del código se usa GitHub.(13) Es una plataforma en línea que facilita la gestión de proyectos y la creación de software. Los desarrolladores pueden alojar, compartir y trabajar en repositorios de código fuente. El sistema de control de versiones Git de GitHub facilita la colaboración entre desarrolladores y el seguimiento de cambios en el código. GitHub es una plataforma para el almacenamiento y el control de versiones de código.



Figura 2.10: Logo de Github

#### Sistema de gestión del proyecto

Para la asignación de tareas y gestión del proyecto se usa Trello. Es una herramienta visual para la gestión de proyectos. Permite a los usuarios organizar tareas, proyectos y colaboradores con tarjetas y listas. Permite a los equipos visualizar y monitorear el progreso de sus proyectos, asignar tareas, establecer plazos y colaborar con la información.

Es una herramienta popular y fácil de usar para administrar tareas y proyectos en una variedad de entornos personales y profesionales.



Figura 2.11: Logo de Trello

#### Generación de Diagrama de Gantt

Para elaborar el Diagrama de Gantt del estudio de viabilidad económica, se usa ProjectLibre. Es una herramienta de gestión de proyectos gratuita y de código abierto. Ayuda a los usuarios a administrar tareas, recursos, calendarios y seguimiento de progreso al ofrecer funciones para planificar, programar y controlar proyectos.



Figura 2.12: Logo de ProjectLibre

### **Herramienta para generar documentación**

Para la creación de esta memoria, se usa la plataforma Overleaf. Es una plataforma en línea que permite la creación y modificación de documentos académicos en el formato LaTeX<sup>3</sup>. Permite la colaboración en tiempo real y ofrece herramientas para la escritura científica, la gestión de referencias y la creación de varios formatos de salida.



Figura 2.13: Logo de Overleaf

---

<sup>3</sup>LaTeX: sistema de composición de textos orientado a documentos científicos.

# Capítulo 3

## Aplicación

En este capítulo se realiza una descripción en detalle de la aplicación, las funcionalidades y componentes de la interfaz del usuario.

### 3.1. Funcionalidades

#### 3.1.1. Pantalla de Bienvenida

La pantalla de bienvenida es lo primero que el usuario ve al acceder a la aplicación. Ésta contiene las funcionalidades elementales de autenticación, como son inicio de sesión, registro y recuperación de la contraseña. El inicio de sesión puede ser mediante un correo electrónico y contraseña y además se ofrece la opción de iniciar sesión mediante Apple, Google o Facebook.

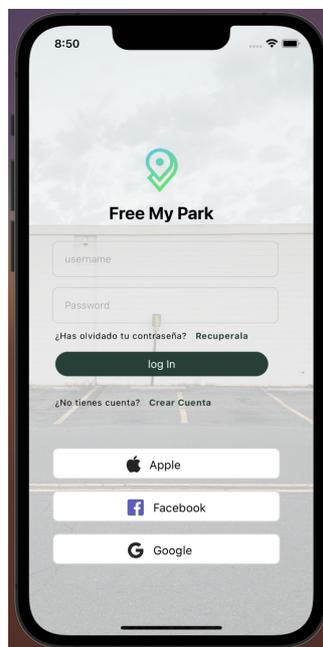


Figura 3.1: Pantalla de Bienvenida

Una vez autenticados, la próxima vez que se acceda a la aplicación, se saltará esta pantalla, accediendo directamente a la pantalla inicial, ya que la sesión se conserva abierta en el dispositivo. Para volver a acceder a esta pantalla, se debe cerrar la sesión del usuario.

### 3.1.2. Registro

La pantalla de registro contiene un formulario con el cual se crea una cuenta en la app. Se requiere de un nombre, correo electrónico y contraseña. Este último campo debe tener una longitud mínima de seis dígitos por seguridad. La cuenta es verificada a través de un código de verificación enviado al correo electrónico.

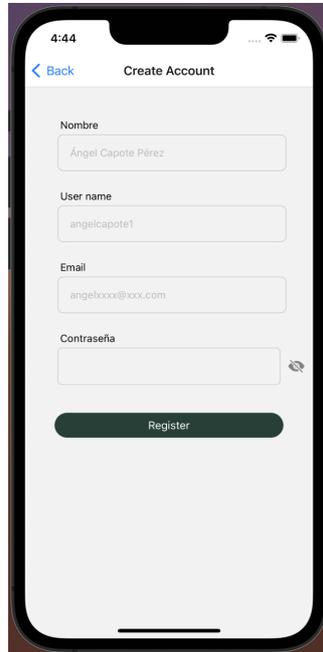


Figura 3.2: Pantalla de Registro

### 3.1.3. Recuperar Contraseña

Esta pantalla permite generar una nueva contraseña en caso de ser olvidada. Funciona a través de un código de verificación que se envía al correo electrónico y con éste, al introducirlo en la app, te permite establecer una nueva contraseña.

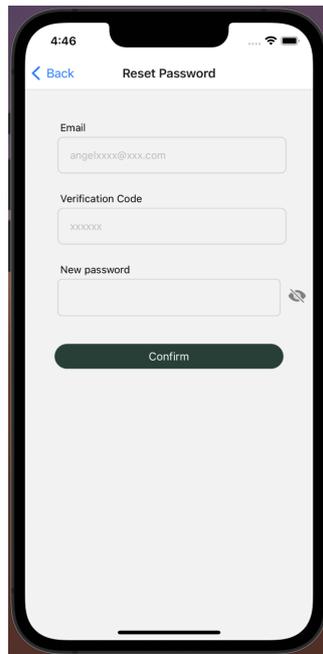


Figura 3.3: Pantalla de Recuperar Contraseña

### 3.1.4. Pantalla Inicial

En la pantalla inicial se despliega un mapa que muestra la ubicación en tiempo real del usuario y también una serie de 'pins' que corresponden a puestos de aparcamiento publicados por otros usuarios. Estos 'pins' vienen listados en la parte inferior donde se proporciona información más detallada.

En la parte inferior derecha de la pantalla, se ubica el botón para publicar un puesto de aparcamiento. Este botón es un círculo de color azul con un coche blanco.

En la parte superior izquierda de la pantalla, se encuentra un avatar que, al pulsar, redirige al perfil del usuario.

En la parte superior derecha, se muestran los puntos de los que dispone la cuenta, los cuales al ser pulsados redirigen a la pantalla de Añadir fondos.

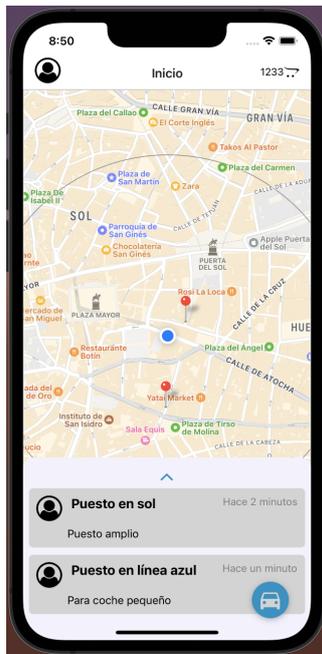


Figura 3.4: Pantalla de Inicio

### 3.1.5. Pantalla de Añadir plaza

Contiene un formulario donde se introduce el nombre y una descripción (opcional). A continuación, muestra un mapa donde se sitúa el pin que correspondería al puesto que se está publicando, se verifica que la ubicación es la correcta y se confirma. Una vez completado todos los campos, se pulsa 'publicar'.

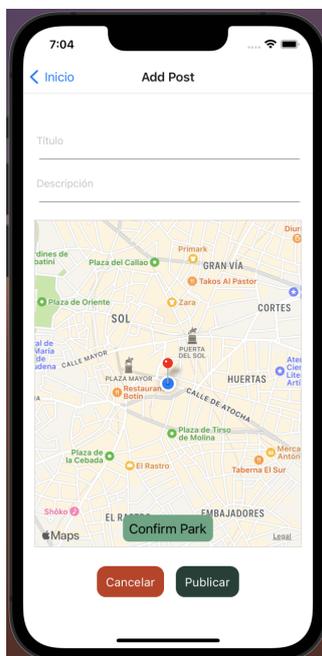


Figura 3.5: Pantalla de Añadir plaza

## Pantalla de Espera

Una vez publicado el puesto, aparece la pantalla de espera donde el oferente se mantendrá a la espera de que otro usuario acepte su plaza. Al ser aceptada la oferta, se notifica al oferente y aparecerá un temporizador en la pantalla que indica al tiempo que tiene el demandante para llegar al lugar. También, se puede consultar la ubicación del demandante. En caso de agotarse el tiempo, tiene la opción de seguir esperando hasta que el usuario llegue o reportar la ausencia. En caso de reportar ausencia, la publicación del puesto se cancela y se vuelve a ofertar la plaza.

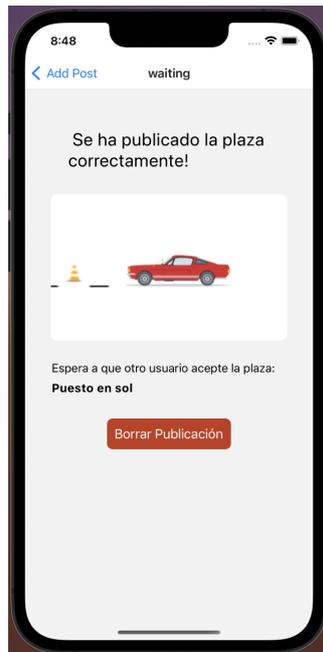


Figura 3.6: Pantalla de Espera

### 3.1.6. Pantalla Aceptar puesto

Una vez pinchado alguno de los puestos situados en la lista en la pantalla inicial, se accede a esta pantalla donde aparece la información detallada del puesto y la ubicación exacta. En caso de estar interesado, se acepta la oferta y se despliega la pantalla de confirmación.

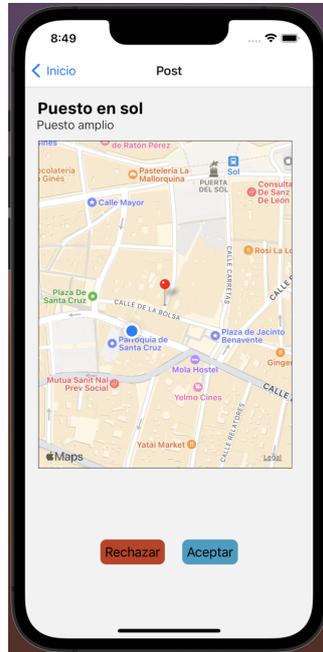


Figura 3.7: Pantalla de Puesto Publicado

### Pantalla confirmación

En esta pantalla aparece un temporizador que indica el tiempo restante que tiene el usuario para llegar a la ubicación. Adicionalmente, se dispone de un botón que dirige a la aplicación GPS del dispositivo y lleva al destino.



Figura 3.8: Pantalla de puesto aceptado

### 3.1.7. Pantalla Añadir fondos

En esta pantalla se realiza la compra de puntos para pagar en la app. Los métodos de pago aceptados son tarjeta, GooglePay, ApplePay o Paypal.

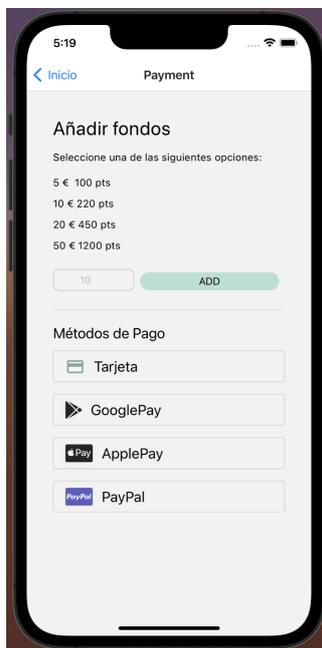


Figura 3.9: Pantalla de Añadir fondos

### 3.1.8. Pantalla perfil

En esta pantalla se muestra toda la información relativa al usuario donde se pueden modificar los datos y cerrar la sesión.



Figura 3.10: Pantalla de Perfil

# Capítulo 4

## GraphQL API

La API actúa como intermediario entre la interfaz de usuario y la base de datos. Como principales diferencias frente a una API REST es que tenemos solo un endpoint y además GraphQL nos permite realizar consultas más personalizadas, sin la necesidad de obtener información no requerida.

Para la creación de la API hemos trabajado con la interfaz de administrador de AWS Amplify, que trabaja a su vez con AWS AppSync. Con la interfaz de administrador creamos gráficamente la base de datos y luego implementamos la interfaz por línea de comando (CLI) que ofrece AWS Amplify para aplicar los cambios sobre el código.

La arquitectura de las GraphQL API es diferente a la de las API REST convencionales. A continuación, se indican las partes principales de la API.

- **Esquema (Schema).** Define la estructura de los datos disponibles y las operaciones que se pueden realizar con ellos.
- **Ti pos de Datos (Data Types).** En GraphQL se pueden crear datos personalizados que representan objetos o entidades en el sistema. En el proyecto, se usa para definir la relación uno a muchos entre la colección User y Post.
- **Consultas (Queries).** Las consultas se usan para obtener información de la Base de Datos. Se implementa la operación Read.

```
type User @model @auth(rules: [{allow: public}]) {
  id: ID!
  name: String
  nickname: String
  bio: String
  avatar: String
  money: Float
  Posts: [Post] @hasMany(indexName: "byUser", fields: ["id"])
  userID: ID
}

type Post @model @auth(rules: [{allow: public}]) {
  id: ID!
  name: String!
  latitude: Float!
  longitude: Float!
  description: String
  userID: ID! @index(name: "byUser")
}
```

Figura 4.1: Schema de la Base de Datos

- **Mutaciones (Mutations).** Las mutaciones modifican la información de la Base de Datos. Se implementan las operaciones Create, Update y Delete.
- **Validación y control de acceso.** Establece reglas para validar y restringir las consultas a los clientes.

En esta aplicación, se han definido reglas de validación, como la autenticación mediante correo electrónico, la restricción de mínimo 6 dígitos la contraseña y la notificación a través de email en vez de SMS <sup>1</sup>.

---

<sup>1</sup>**SMS:** sistema de mensajería en dispositivos móviles

# Capítulo 5

## Base de datos

El proyecto cuenta con una base de datos no relacional DynamoDB que proporciona AWS Amplify(14).

AWS Amplify es un conjunto de servicios y herramientas de Amazon Web Services (AWS) que ayuda a desarrollar e implementar aplicaciones web y móviles. Amplify ofrece una forma rápida y escalable de construir aplicaciones basadas en la nube, centrándose en simplificar el proceso de desarrollo.

La autenticación, el almacenamiento de datos, las API GraphQL, las funciones sin servidor y las notificaciones push son algunos de los servicios administrados que ofrece Amplify. Estos servicios funcionan bien con otras tecnologías de AWS, como Amazon Cognito, para la gestión y autenticación de usuarios, Amazon S3 para el almacenamiento de archivos y Amazon DynamoDB para bases de datos NoSQL.

Aparte de los servicios administrados, Amplify ofrece una biblioteca de JavaScript y una CLI (interfaz de línea de comandos) que facilita la configuración y el acceso a los servicios de AWS a través de la aplicación. Amplify también facilita la implementación y el hospedaje de la aplicación en la nube de AWS, lo que permite escalar y administrar automáticamente la infraestructura necesaria para la ejecución de la aplicación.



Figura 5.1: Logo de AWS Amplify

A diferencia de las bases de datos relacionales, las no relacionales se almacenan en colecciones que contienen documentos que son el equivalente a las tablas y las filas respectivamente.

Para relacionar las tablas entre sí, se almacena el Object Id (identificador) de una colección en otro, como es el caso de la relación uno a muchos entre la colección Post que almacena el ObjectId de la colección Usuario.

## 5.1. Creación de la base de datos

Para la creación de la base de datos se usó la Interfaz de Administrador proporcionada por AWS Amplify con la cual se crean las colecciones con los atributos correspondientes. A continuación, se muestran las dos colecciones creadas.

The image shows two side-by-side panels for defining database collections. The left panel is for the 'User' collection and the right panel is for the 'Post' collection. Each panel has a title bar with a close button (X) and a checkbox.

**User Collection:**

Field name	Type
id	ID!
name	String
nickname	String
bio	String
avatar	String
money	Float
userID	ID

+ Add a field

Relationship name	Related to	Cardinality
Posts	Post	1:n one ...

+ Add a relationship

**Post Collection:**

Field name	Type
id	ID!
name	String!
latitude	Float!
longitude	Float!
description	String
userID	Relationship Source

+ Add a field

+ Add a relationship

Figura 5.2: Schema de la base de datos

# Capítulo 6

## Estudio de viabilidad económica del desarrollo de una Aplicación Profesional

En este capítulo se realiza un estudio de viabilidad económica del desarrollo de una aplicación profesional, estimando la rentabilidad de materializar el proyecto. El estudio está dirigido a fondos de inversión o business angels<sup>1</sup> que deseen invertir en proyectos innovadores como el que se presenta.

En este capítulo veremos cuanto costaría desde el punto de vista de un proyecto de ingeniera del software desarrollar la aplicación profesional.

### 6.1. Características destacadas de la aplicación profesional

En el capítulo 33 se explica el desarrollo de un prototipo que representa un mockup<sup>2</sup> de cómo podría ser la aplicación final. Sin embargo, durante su desarrollo se observó que los factores tiempo y costo condicionan la implementación de varios objetivos. Por tal razón, para el desarrollo de la aplicación profesional se implementarán las siguientes mejoras:

- Aplicar un algoritmo que limite la adquisición de plazas, siendo la proporción mínima una plaza ofertada para un máximo de cinco plazas demandadas. De esta manera, se pretende lograr la participación de los usuarios en ambos roles.
- Aplicar un filtro para que solo se muestren plazas dentro de un radio de seis kilómetros.
- Opción de pago integrada. Implementación de una pasarela de pago seguro y un TPV.
- Implementación de un sistema de navegación interno. Por un lado, para guiar al usuario demandante y, por otro lado, ser rastreado por el usuario ofertante.
- Sistema de mensajería interna entre los usuarios.
- Notificaciones emergentes.

---

<sup>1</sup>**Business angels:** persona jurídica que ofrece financiación a nuevos proyectos con el fin de obtener una ganancia futura

<sup>2</sup>**Mockup:** fotomontaje a través del cual los diseñadores gráficos presentan propuestas a sus clientes

## 6.2. Desarrollo del proyecto

EL proyecto se determina mediante un enfoque secuencial, es decir, las fases son realizadas seguidamente. Antes de pasar a la siguiente fase, se completa la fase en la que se estaba trabajando. Este método proporciona una estructura clara y facilita la gestión del proyecto.

A continuación, se comentarán las seis fases en las que se ha dividido el proyecto:

- **Fase de análisis y definición de requisitos:** Esta primera fase comprende la definición de los objetivos y requisitos de la aplicación, estudio de viabilidad y la elaboración del documento de especificaciones funcionales y técnicas. En esta etapa, se obtiene una visión clara del alcance del proyecto y se establecen los plazos y recursos necesarios.
- **Fase de Diseño de la interfaz de Usuario:** En esta fase se diseña la interfaz y la experiencia de usuario de la app. Se elabora un diseño visual y se establecen las arquitecturas de la app y de la base de datos. Esta fase se centra en establecer la base técnica necesaria para el desarrollo y crear una experiencia de usuario intuitiva y atractiva.
- **Fase de desarrollo:** Durante esta fase se implementa la aplicación. Se establece el entorno de desarrollo y se desarrolla tanto el frontend(UI/UX) como el backend (lógica y gestión de datos). Para garantizar que la aplicación funcione correctamente, se realizan pruebas unitarias y se integran servicios externos.
- **Fase de Pruebas:** A lo largo de esta etapa, se realiza una serie de pruebas para garantizar la calidad de la aplicación. Los casos de prueba se diseñan y realizan para verificar que cumplen con los requisitos y funcionalidades definidos. Se realizan pruebas para el funcionamiento, rendimiento, seguridad, compatibilidad y usabilidad.
- **Ajustes y optimización:** Esta fase se dedica a analizar los resultados de las pruebas y realizar ajustes y optimizaciones en la aplicación. Los errores se corrigen, el rendimiento se mejora y la experiencia de usuario se optimiza. Esta etapa tiene como objetivo refinar y pulir la aplicación antes de su lanzamiento.
- **Fase de Lanzamiento:** En esta fase final se realiza la preparación y el despliegue de la aplicación para su distribución. Los recursos y activos necesarios se preparan, los servidores se configuran y la aplicación se publica en las tiendas de aplicaciones correspondientes. Se realiza la promoción y publicidad para dar a conocer la aplicación. Una vez lanzada, se monitorea y se resuelven problemas potenciales al principio.

Una vez definidas las fases, se listarán todas las tareas a realizar y su tiempo de duración.

Nombre	Duracion	Inicio	Terminado
<b>☐ Fase de Análisis y Definición de requisitos</b>	<b>30 days?</b>	<b>7/07/23 8:00</b>	<b>17/08/23 17:00</b>
Definir objetivos y requisitos	5 days?	7/07/23 8:00	13/07/23 17:00
Realizar estudio de viabilidad y estado del arte	10 days?	14/07/23 8:00	27/07/23 17:00
Crear documento de especializaciones técnicas y funcionales	5 days?	28/07/23 8:00	3/08/23 17:00
Definir casos de uso y diagramas de flujo	5 days?	4/08/23 8:00	10/08/23 17:00
Establecer el alcance y los plazos del proyecto	5 days?	11/08/23 8:00	17/08/23 17:00
<b>☐ Fase de Diseño</b>	<b>46 days?</b>	<b>7/07/23 8:00</b>	<b>8/09/23 17:00</b>
Diseñar UI y UX	5 days?	7/07/23 8:00	13/07/23 17:00
Crear prototipos interactivos	5 days?	21/08/23 8:00	25/08/23 17:00
Definir la arquitectura de la aplicación	5 days?	28/08/23 8:00	1/09/23 17:00
Diseñar la estructura de la base de datos	5 days?	4/09/23 8:00	8/09/23 17:00
<b>☐ Fase de Desarrollo</b>	<b>75 days?</b>	<b>11/09/23 8:00</b>	<b>22/12/23 17:00</b>
Desarrollar el backend de la aplicación	20 days?	11/09/23 8:00	6/10/23 17:00
Integrar APIs y servicios externos	10 days?	9/10/23 8:00	20/10/23 17:00
Desarrollar el frontend de la aplicación	20 days?	23/10/23 8:00	17/11/23 17:00
Implementar la navegación y las interacciones de la interfaz	15 days?	20/11/23 8:00	8/12/23 17:00
Realizar pruebas unitarias y de integración	10 days?	11/12/23 8:00	22/12/23 17:00
<b>☐ Fase de Pruebas</b>	<b>40 days?</b>	<b>25/12/23 8:00</b>	<b>16/02/24 17:00</b>
Planificar y diseñar casos de prueba	5 days?	25/12/23 8:00	29/12/23 17:00
Realizar pruebas funcionales para validar la funcionalidad	10 days?	1/01/24 8:00	12/01/24 17:00
Realizar pruebas de compatibilidad en diferentes dispositivos y siste	10 days?	15/01/24 8:00	26/01/24 17:00
Documentar y corregir los errores y problemas encontrados	5 days?	29/01/24 8:00	2/02/24 17:00
Realizar pruebas de aceptación con usuarios finales	10 days?	5/02/24 8:00	16/02/24 17:00
<b>☐ Fase de Ajustes</b>	<b>25 days?</b>	<b>19/02/24 8:00</b>	<b>22/03/24 17:00</b>
Analizar los resultados de las pruebas y retroalimentación del usuar	5 days?	19/02/24 8:00	23/02/24 17:00
Corregir errores y realizar ajustes en el código	10 days?	26/02/24 8:00	8/03/24 17:00
Mejorar la usabilidad y la experiencia de usuario	10 days?	11/03/24 8:00	22/03/24 17:00
<b>☐ Fase de Lanzamiento</b>	<b>36,818 da...</b>	<b>25/03/24 8:00</b>	<b>14/05/24 15:32</b>
Preparar los recursos y activos necesarios para el lanzamiento	5 days?	25/03/24 8:00	29/03/24 17:00
Desplegar la App en los servidores y plataformas	5 days?	1/04/24 8:00	5/04/24 17:00
Realizar pruebas finales en el entorno de producción	5 days?	8/04/24 8:00	12/04/24 17:00
Promocionar y publicar la aplicación	10 days?	15/04/24 8:00	26/04/24 17:00
Monitorear y corregir posibles problemas	11,818 d...	29/04/24 8:00	14/05/24 15:32

Figura 6.1: Tareas de cada Fase

Para empezar a desarrollar el proyecto, se deben definir los recursos necesarios. Estos recursos son tanto materiales como de personal.

### Recursos materiales:

Recursos materiales		
Recurso	Precio	
Pagos cada mes	141,50 €	
Pagos cada año	118,00 €	Frecuencia
Dominio	21,00 €	Mensual
Servicios AWS Amplify	120,50 €	Mensual
Licencia iOS	100,00 €	Anual
SSL	18,00 €	Anual
Movil iOS	800,00 €	Único
Movil Android	500,00 €	Único
Licencia Play Store	25,00 €	Único

Figura 6.2: Recursos Materiales

Como se observa en la tabla, hay recursos que solo requieren de un único pago como son los dispositivos móviles tanto iOS como Android y la licencia en la Play Store. Adicionalmente, hay algunos recursos que suponen un gasto mensual como son el dominio y los servicios que nos proporciona AWS AMplify, además de otros que son gastos anuales como el SSL<sup>3</sup> y la licencia de iOS.

Los servicios de AWS Amplify incluyen:

- Alojamiento web y almacenamiento.
- Autenticación y autorización de usuarios.
- Base de datos.
- Notificación y mensajería.

En resumen, el gasto de recursos materiales inicial se estima en 1.584,50€. Adicionalmente, habrá un gasto mensual de 141,5€ y un cobro anual de 121€, lo que asciende a un total de 1.819€ anuales aproximadamente.

---

<sup>3</sup>SSL: tecnología que se usa para establecer una conexión segura entre dispositivos.

### Recursos de personal:

Se ha realizado una tabla con los cargos requeridos, su función y el coste por hora. Los datos han sido obtenidos de Talent <sup>4</sup>.

Cargo	Funcionalidad	Salario por hora	Horas trabajadas
Gerente de Proyecto	Responsable de liderar y coordinar el proyecto, gestionar los recursos y garantizar el cumplimiento de los objetivos	29,23€	440 horas
Analista de Proyectos	responsable de recopilar y analizar los requisitos comerciales, determinar las funcionalidades y establecer casos de uso.	11,79€	560 horas
Diseñador UI/UX	Responsable del diseño de la experiencia de usuario, la usabilidad y la interfaz gráfica de la aplicación.	15,38€	480 horas
Diseñador Gráfico	Responsable de crear activos visuales atractivos que promuevan la aplicación, como imágenes, videos y materiales gráficos.	10,56€	440 horas
Desarrollador de backend	Experto en programación y desarrollo del lado del servidor, se ocupa de la implementación de la lógica de negocio, la gestión de datos y la integración con servicios externos.	16,15€	680 horas
Arquitecto de software	Responsable de diseñar y establecer la arquitectura técnica de la aplicación, garantizando su escalabilidad, rendimiento y seguridad.	26,92€	440 horas
Desarrollador de frontend	Se encarga de la creación de la interfaz de usuario, la interacción con el usuario y la experiencia del usuario, siendo un experto en programación y desarrollo del lado del cliente.	17,18€	720 horas
Ingeniero de Pruebas	A través de pruebas unitarias, de integración y funcionales, verifican la calidad y el funcionamiento adecuado de la aplicación.	13,85€	720 horas
Especialista en Cumplimiento Normativo	Encargado de garantizar que se cumplen todos los requisitos legislativos	10,77€	360 horas
Analista de Calidad	Encargado de asegurar el correcto funcionamiento de la aplicación	11,28€	720 horas
Especialista en Pruebas de Seguridad y Usabilidad	Se enfoca en medir y evaluar el rendimiento de la aplicación, identificar cuellos de botella, posibles brechas de seguridad y optimizar el rendimiento.	14,07€	720 horas
Gerente de Marketing	Responsable de la estrategia de marketing y la promoción de la aplicación.	17,95€	80 horas
Especialista en Marketing Digital	Encargado de ejecutar las estrategias de marketing en línea, como publicidad en redes sociales, optimización de motores de búsqueda (SEO) y campañas de email marketing.	11,67€	80 horas
Especialista en Analítica de Aplicaciones	Encargado de monitorear y analizar datos relacionados con la adquisición de usuarios y el comportamiento de los usuarios	12,38€	80 horas

Tabla 6.1: Tablas recursos personal

<sup>4</sup>Talent: página web de estimación de salarios en España

Una vez definidos los recursos y las tareas, se crea el Diagrama de Gantt (15)

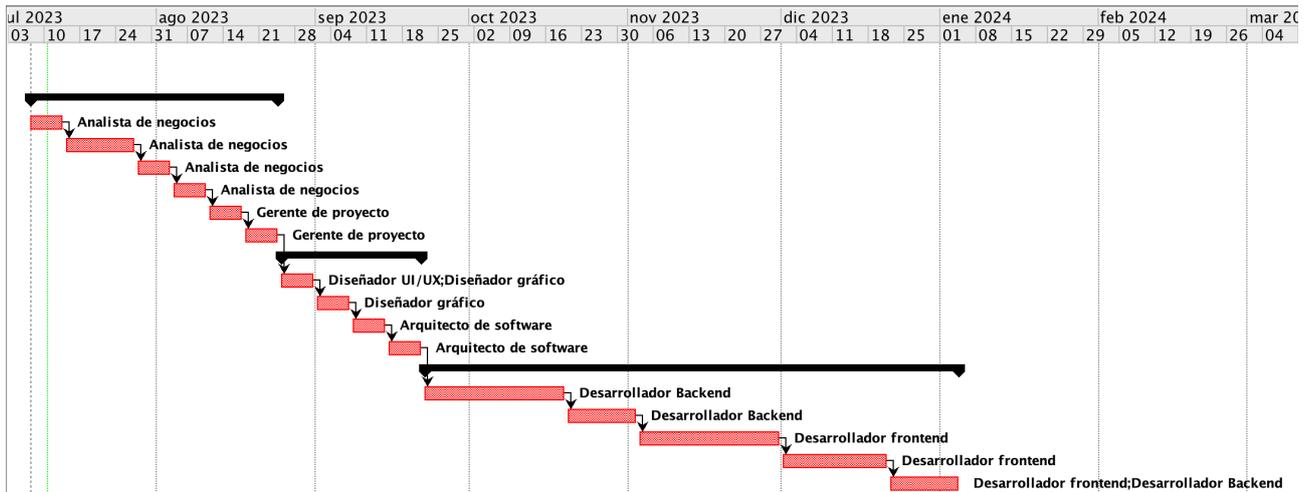


Figura 6.3: Primera parte Diagrama de Gantt

Esta primera parte agrupa las tareas de la fase de Análisis y Definición de Requisitos, la Fase de Diseño y la de Desarrollo.

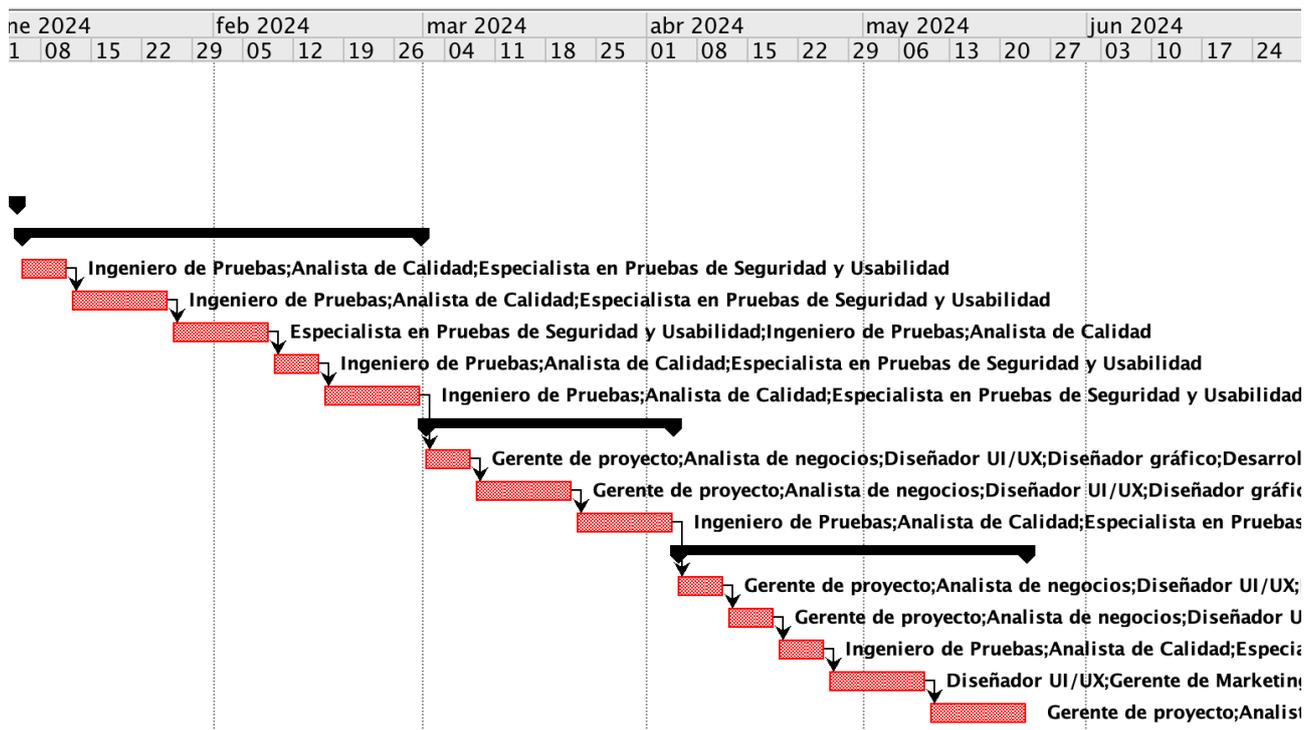


Figura 6.4: Segunda parte Diagrama de Gantt

En la segunda parte, se agrupan las tareas correspondientes a la fase de Pruebas, la fase de Ajustes y la de Lanzamiento.

De acuerdo con el Diagrama de Gantt realizado, el coste de desarrollar el proyecto profesional será de 106.535,00€ y tomará un tiempo estimado de 230 días.

Para publicar y promocionar la aplicación, se ha diseñado una **estrategia de marketing**, que consiste en:

■ **Marketing en línea:**

- Creación de perfiles en redes sociales para la posterior publicidad en estas. (1000€/mes)
- Campañas de Google AdWords (búsqueda y display) (1.500€)
- Utilizar estrategias de optimización de motores de búsqueda (SEO) y publicidad en redes sociales para aumentar la visualización.

■ **Colaboraciones y asociaciones:**

- Colaborar con empresas locales y eventos para promocionar y dar conocer la aplicación.

■ **Relaciones públicas y marketing de contenido:**

- Envío de comunicados de prensa a medios relevantes e influencers. (300€ / comunicado)

La estrategia de marketing tiene un costo estimado de 2800€ al mes.

### 6.3. Modelo de Comercialización

Una vez finalizado el desarrollo del proyecto y puesto en el mercado para ser descargado por los usuarios en cualquier plataforma, se ha diseñado un modelo de comercialización donde se explica la obtención de los ingresos y el retorno de la inversión inicial y posteriores beneficios.

El modelo de comercialización de la aplicación se basa en la adquisición de puntos dentro de la aplicación para poder realizar las reservas de los puestos de aparcamiento.

Obtener una plaza de garaje tiene un coste de 40 puntos y el usuario que publica la plaza obtiene 25. Los puntos restantes son el beneficio que obtiene la aplicación.

La compra de puntos se pueden realizar a través de dos métodos:

- Introduciendo una cantidad de dinero en la aplicación para obtener los puntos equivalentes.
- Elegir uno de los paquetes ofrecidos en la aplicación, donde se obtienen más puntos que realizando la compra individualmente. Los paquetes son:
  - 100 ptos por 5€
  - 220 ptos por 10€
  - 450 ptos por 20€
  - 1200 ptos por 50€

Como apoyo económico, se prevee integrar anuncios en la aplicación a través de GoogleAds u otras plataformas.

En un futuro se estudiará la opción de implementar un sistema de suscripción donde se otorgan una serie de beneficios como podrían ser:

- Demanda ilimitada de plazas.
- Mayor radio de búsqueda.

### 6.3.1. Estudio del Retorno de la Inversión

El retorno de inversión (ROI)(16) es una medida financiera utilizada para evaluar la rentabilidad de un proyecto.

Partiendo de los datos obtenidos, el desarrollo del proyecto tendría una duración de 230 días, un coste inicial de 106.535,00€ y se estima un ingreso semanal para la aplicación por usuario de 4€ al mes.

Mes	Usuarios	Ingreso semanal	Ingreso Acumulado	Gasto acumulado	
0	0	0,00 €	0,00 €	106.534,50 €	
1	10	80,00 €	80,00 €	109.476,00 €	
2	36	400,00 €	480,00 €	112.417,50 €	
3	78	976,00 €	1.456,00 €	115.359,00 €	
4	136	1.808,00 €	3.264,00 €	118.300,50 €	
5	210	2.896,00 €	6.160,00 €	121.242,00 €	
6	300	4.240,00 €	10.400,00 €	124.183,50 €	
7	406	5.840,00 €	16.240,00 €	127.125,00 €	
8	528	7.696,00 €	23.936,00 €	130.066,50 €	
9	666	9.808,00 €	33.744,00 €	133.008,00 €	
10	820	12.176,00 €	45.920,00 €	135.949,50 €	
11	990	14.800,00 €	60.720,00 €	138.891,00 €	
12	1176	17.680,00 €	78.400,00 €	141.832,50 €	pago anual
13	1378	20.816,00 €	99.216,00 €	144.892,00 €	
14	1596	24.208,00 €	123.424,00 €	147.833,50 €	
15	1830	27.856,00 €	151.280,00 €	150.775,00 €	ROI
16	2080	31.772,00 €	183.052,00 €	153.716,50 €	
17	2346	35.920,00 €	218.972,00 €	156.658,00 €	
18	2628	40.336,00 €	259.308,00 €	159.599,50 €	

Figura 6.5: Tabla ingresos mensuales

A partir de esta tabla, se obtiene la gráfica que se despliega a continuación 6.6. En la misma se representa por un lado el gasto acumulado, que parte de 106.535,00€, observándose el aumento constante que suponen los gastos de explotación. Estos gastos abarcan: marketing, personal, mantenimiento, publicidad en redes sociales, entre otros. Y, por otro lado, se representa el ingreso acumulado que inicialmente presenta un crecimiento lento pero, conforme transcurre el tiempo, se hace exponencial.

El punto de intersección entre los gastos acumulados y los ingresos acumulados es el punto de retorno de la inversión, es decir, el ROI.

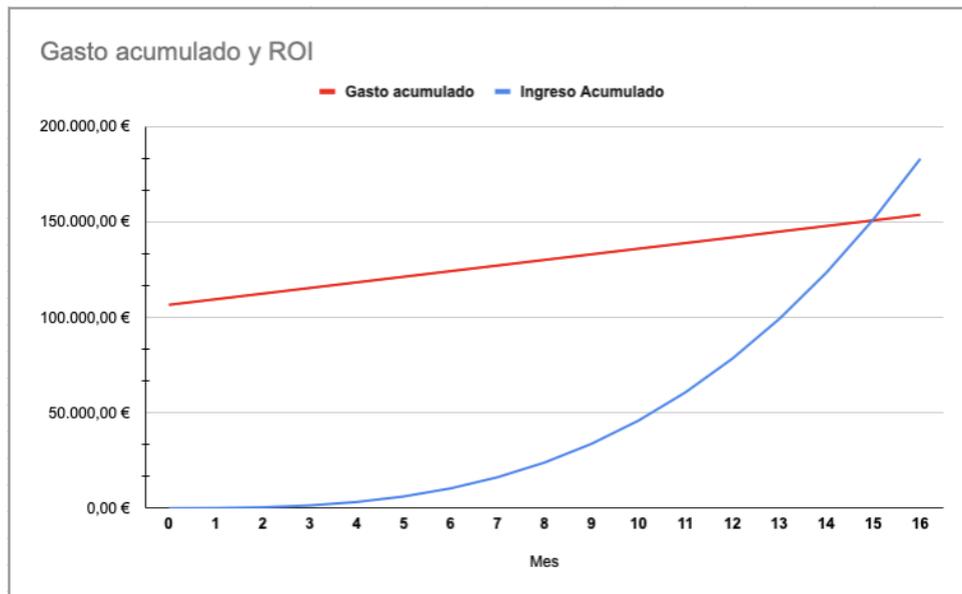


Figura 6.6: Gráfica de Gasto Acumulado y ROI

La gráfica indica que el punto de retorno de la inversión estaría en el mes 15, es decir, un año y tres meses del despliegue de la aplicación.

Cabe mencionar que esta gráfica no contempla ingresos por subvenciones, colaboraciones o algún tipo de publicidad externa como Google AdSense.

# Capítulo 7

## Conclusiones y líneas futuras

Para el desarrollo del proyecto se ha realizado un estudio detallado de todos los requisitos y necesidades que debe cumplimentar la aplicación, así como las tecnologías y herramientas disponibles en el mercado para su consecución.

Gracias a la estructura del código, la aplicación puede adaptarse a prácticamente cualquier entorno, ya que es una aplicación que funciona a través de geolocalización, por lo que la escalabilidad es elevada.

A pesar de las limitaciones económicas y temporales a la hora de su desarrollo, la aplicación es totalmente funcional. Sin embargo, como está especificado en el estudio de viabilidad económica<sup>6</sup>, hay una serie de mejoras a implementar que aportan gran valor a la aplicación.

Para concluir, se debe tener presente que el éxito de la aplicación requiere de la voluntad activa de los usuarios, solo ellos dan valor efectivo a la aplicación, y para ello se necesita que las comunidades sean conscientes de las mejoras en términos de calidad de vida que el uso de estas alternativas les puede aportar.

# Capítulo 8

## Summary and Conclusions

For the development of the project, a detailed study was made of all the requirements and needs to be covered by the application, as well as the technologies and tools available on the market for its implementation.

Thanks to the structure of the code, the application can be adapted to practically any environment, since it is an application that works through geolocation, so the scalability is high.

Despite the economic and time constraints at the time of its development, the application is fully functional. However, as specified in the economic feasibility study<sup>6</sup>, there are a number of improvements to be implemented that will add great value to the application.

In conclusion, it should be remembered that the success of the application requires the active will of the users, only they will give effective value to the application, and for this it is necessary that the communities are aware of the improvements in terms of quality of life that the use of these alternatives can bring them.

# Capítulo 9

## Presupuesto

Además de realizar el estudio de viabilidad económica, se ha llevado a cabo un estudio sobre el tiempo dedicado para realizar el Trabajo de Fin de Grado y el coste que supone.

<b>Tipos</b>	<b>Descripción</b>	<b>Coste</b>
Investigación y selección de tecnologías	60 horas	15€
Estudio previo y análisis de requisitos	45 horas	20€
Desarrollo aplicación	160 horas	15€
Documentación	40 horas	15€
Total	305 horas	4800€

Tabla 9.1: Tabla de Costes asociados a la elaboración del proyecto

Finalmente, este Trabajo de Fin de Grado se ha llegado a cabo en 305 horas, con un coste total de 4800€.

# Bibliografía

- [1] Castro, M. D., Martín-Vide, J., & Alonso, S. (2005). El clima de España: pasado, presente y escenarios de clima para el siglo XXI.
- [2] de Cambio Climático, O. E. (2006). Plan nacional de adaptación al cambio climático. Oficina Española de Cambio Climático. Torreguil, España.
- [3] De Motes, J. M. (2008). El crecimiento moderno de la población de España de 1850 a 2001: una serie homogénea anual. *Investigaciones de Historia Económica*, 4(10), 129-162.
- [4] de España, G. (1999). Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. *Bol Del Estado*, 298(2), 43088-43099.
- [5] de España, G. (1999). Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. *Bol Del Estado*, 298(2), 43088-43099.
- [6] Gutierrez, D. (2011). Casos de uso Diagramas de Casos de Uso. *Gutierrez, Demián*, 1, 45.
- [7] Lizama, O., Kindley, G., Morales, J. J., & Gonzales, A. (2016). *Redes de computadores Arquitectura Cliente-Servidor*. Universidad Tecnica Federico Santa Maria, 1-8.
- [8] Native, R. (2020). *React Native*. línea]. Disponible en: <https://reactnative.dev/>. [Último acceso: 2 de noviembre 2019].
- [9] Collaguazo, M. L. R. Q., Venegas, M. M. S. P., Guerrero, A. A. A., Freire, M. N. M., & Beltrán, M. S. H. C. (2022). Desarrollo híbrido con flutter. *Ciencia Latina Revista Científica Multidisciplinar*, 6(4), 4594-4609.
- [10] Macias Vera, E. V. (2021). Estudio comparativo de los frameworks del desarrollo móvil nativo "FlutterzReact Native"(Bachelor's thesis, BABAHOYO: UTB, 2021).
- [11] Bryant, M. (2017, December). GraphQL for archival metadata: An overview of the EHRI GraphQL API. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 2225-2230). IEEE.
- [12] Sivasubramanian, S. (2012, May). Amazon dynamoDB: a seamlessly scalable non-relational database service. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (pp. 729-730).
- [13] Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2014, May). The promises and perils of mining github. In *Proceedings of the 11th working conference on mining software repositories* (pp. 92-101).

- [14] Lessa, H. (2019). Production-grade full-stack apps with AWS Amplify. In AWS Re: Invent 2019.
- [15] Hinojosa, M. A. (2003). Diagrama de gantt. Producción, procesos y operaciones, 48.
- [16] Cuevas Villegas, C. F. (2001). Medición del desempeño: retorno sobre inversión, ROI; ingreso residual, IR; valor económico agregado, EVA; análisis comparado. Estudios Gerenciales, 17(79), 13-22.