

Sistema de reserva de plaza de aparcamiento.

Alumno:

Pablo Retamero Sánchez

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Aplicaciones multimedia de nueva generación

Profesor:

David García Solórzano

Enero de 2022



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Sistema de reserva de plaza de aparcamiento</i>
Nombre del autor:	<i>Pablo Retamero Sánchez</i>
Nombre del consultor/a:	David García Solórzano
Nombre del PRA:	David García Solórzano
Fecha de entrega (mm/aaaa):	01/2022
Titulación:	Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Área del Trabajo Final:	<i>Aplicaciones multimedia de nueva generación</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Parking space, booking, saving money.</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El tráfico en las ciudades supone una fuente de ruido, contaminación y gasto energético que ha ido aumentando con el tiempo. Una parte de este tráfico que colapsa las ciudades lo provocan conductores que tratan de encontrar una plaza de aparcamiento.</p> <p>El presente trabajo tiene como finalidad el desarrollo de una aplicación que facilite la reserva de una plaza de aparcamiento para que los conductores dediquen el mínimo tiempo posible a esta tarea. De este modo, se consigue reducir el tráfico en las ciudades y por tanto, la contaminación y el gasto energético.</p> <p>La particularidad del presente trabajo es la idea de la implantación de paneles luminosos en las plazas de aparcamiento. Cuando un usuario reserva una plaza desde la aplicación un panel luminoso se enciende para indicar que la plaza está reservada y así ningún otro conductor puede ocuparla.</p> <p>La aplicación se ha desarrollado para dispositivos Android, para lo cual se ha utilizado el software de desarrollo Android Studio.</p> <p>Como resultado final se ha conseguido desarrollar la aplicación que se podría utilizar para llevar a cabo la propuesta de este trabajo, con los servidores de Google Maps, Autenticación y Base de datos listos para comenzar a funcionar.</p>	

Abstract (in English, 250 words or less):

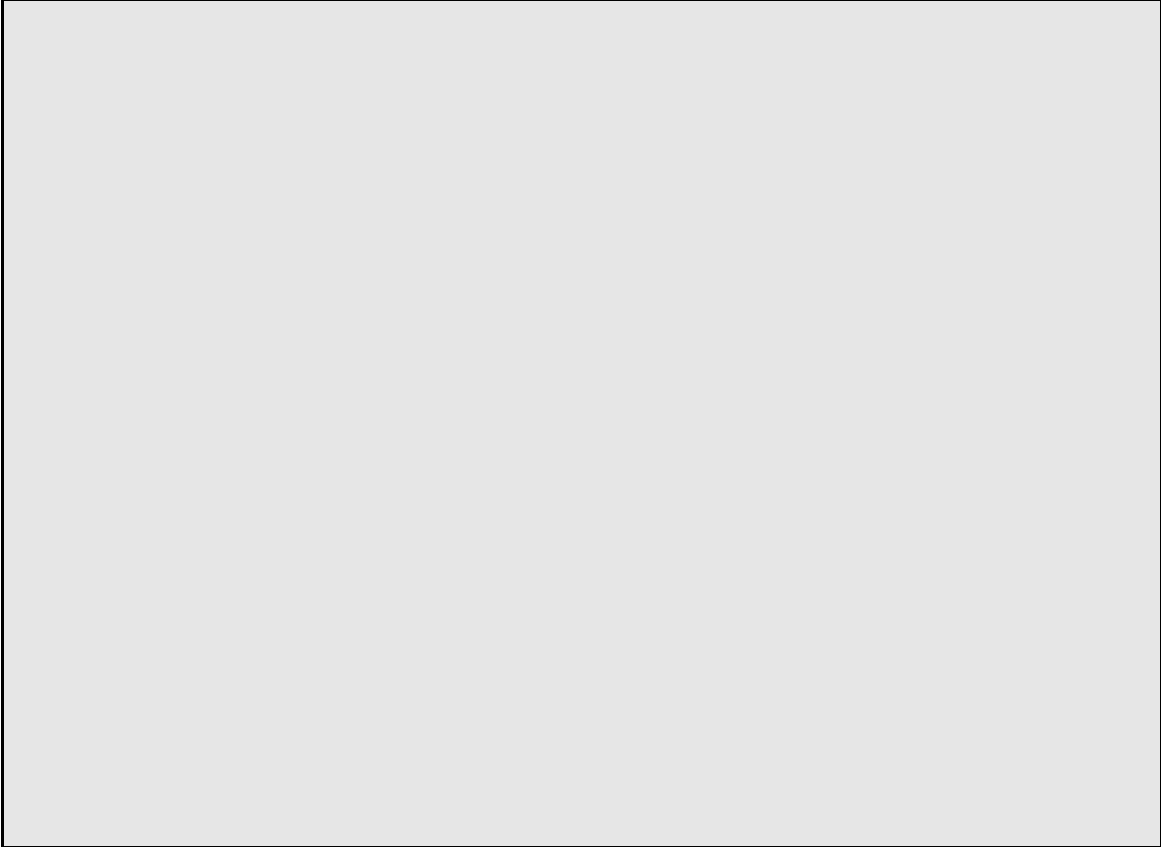
Traffic in cities is a source of noise, pollution and energy consumption that has been increasing over time. Some of this city collapsing traffic is caused by drivers trying to find a free parking space.

The purpose of this work is to develop an application that facilitates the reservation of a parking space so that drivers spend as little time as possible on this task. In this way, it is possible to reduce traffic in cities and therefore, pollution and energy consumption.

The particularity of this work is the idea of installing light panels in parking spaces. When a user reserves a seat from the application, a light panel lights up to indicate that the seat is reserved and thus no other driver can occupy it.

The application has been developed for Android devices, for which the Android Studio development software has been used.

As a final result, it has been possible to develop the application that could be used to carry out the proposal of this work, with the Google Maps, Authentication and Database servers ready to start working.



Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo	3
1.5 Breve resumen de productos obtenidos	3
1.6 Breve descripción de los otros capítulos de la memoria.....	3
2. Estado del Arte.....	5
2.1 Aplicaciones similares	5
2.2 Herramientas de desarrollo para la app	8
2.3 Base de datos	9
2.4 Hardware necesario	15
3. Diseño y desarrollo de la aplicación	16
3.1 Diseño gráfico de la aplicación	17
3.2 Integración de Google Maps	20
3.3 Activación de la base de datos.....	21
3.4 Presentación de la aplicación.....	24
3.5 Puesta a prueba	28
4. Conclusiones.....	30
5. Glosario	32
6. Bibliografía	33

Lista de figuras

Figura 1. Google play for parking	5
Figura 2. EasyPark.....	6
Figura 3. Parkopedia	6
Figura 4. Comparativa de Bases de datos	12
Figura 5. Servicios de Firebase.....	13
Figura 6. Raspberry.....	15
Figura 7. Arduino.....	15
Figura 8. ESP8266.....	16
Figura 9. Diseño autenticación	18
Figura 10. Diseño de pantalla principal	19
Figura 11. Estructura de la pantalla principal	19
Figura 12. Icono de la aplicación.....	20
Figura 13. Plantillas de Android Studio.....	20
Figura 14. Google Maps Platform.....	21
Figura 15. Identificador de Google Maps en AndroidManifest.....	21
Figura 16. Gestión de proyectos en Firebase.....	22
Figura 17. Consola de Firebase	22
Figura 18. Autenticación en Firebase.....	23
Figura 19. Base de datos de autenticación	23
Figura 20. Base de datos de Firestore	24
Figura 21. Icono de la aplicación.....	24
Figura 22. Pantalla de autenticación	25
Figura 23. Error en autenticación	26
Figura 24. Pantalla principal.....	27
Figura 25. Selección de la plaza en Google Maps	27
Figura 26. Abandonar la plaza ocupada.....	28

1. Introducción

1.1 Contexto y justificación del Trabajo

El creciente número de habitantes en zonas urbanas plantea una serie de retos de cara al futuro, de los cuales nació el concepto Smart city, o ciudad inteligente, partiendo de la necesidad de crear ciudades más inteligentes y sostenibles que puedan dar solución a esos nuevos retos sociales. De estos nuevos retos, este proyecto se centra en el problema que supone el tráfico en las ciudades: Ruido, contaminación del aire, gasto energético y pérdida de tiempo en retenciones.

La búsqueda de aparcamiento en las ciudades suele ser un problema, especialmente a ciertas horas o cuando se realizan celebraciones o eventos en determinados sitios.

Existen aplicaciones capaces de proporcionar información sobre plazas de aparcamiento libres en las calles de una ciudad, de manera que se puede programar una ruta hacia esa plaza de aparcamiento, bien siguiendo las indicaciones del sistema GPS (situación actual generalizada) o bien indicando al coche autónomo (en un futuro) que queremos que estacione el vehículo en esa plaza de aparcamiento. El problema surge si durante el trayecto esa plaza de aparcamiento la ocupa otro vehículo que se nos anticipa, lo cual nos dejaría un tanto desamparados, nos habría hecho perder tiempo y energía, además, nos veríamos en la urgente necesidad de programar una nueva ruta. A veces, incluso los aparcamientos privados se encuentran completos, lo cual en ocasiones provoca que el conductor desista de su búsqueda y cambie su destino.

Durante el tiempo que un vehículo se encuentra circulando únicamente para encontrar donde aparcar se sigue generando ruido, contaminación, consumo energético y supone una pérdida de tiempo y dinero para el conductor. Existen múltiples estudios que informan sobre los grandes riesgos que suponen estos factores de cara a la salud de las personas y a la economía [1,2].

Mediante un sistema que permita reservar una plaza de aparcamiento, de forma que nadie pudiera ocuparla, se evitaría este gran problema.

Este proyecto desarrollará una aplicación para smartphone capaz de recibir la información del centro de operaciones de red para mostrar al usuario las plazas de aparcamiento que están disponibles. Desde esta aplicación se podrá reservar una de estas plazas, de manera que al hacerlo una luz led (que simulará un panel luminoso) se encienda encima de esta plaza, indicando que esa plaza ya está reservada y que nadie más puede aparcar en ella.

La reserva de una de estas plazas se podría tarificar de la misma forma que se hace actualmente en “zona azul”, de manera que la tarificación comenzara desde el momento en el que se confirmara la reserva de la plaza de aparcamiento y finalizara al abandonar esa plaza de aparcamiento.

El sistema sería totalmente compatible con la actual “zona azul”, tan solo habría que instalar los paneles luminosos entorno a cada plaza de aparcamiento, de manera que se pueda advertir claramente a otros conductores, cuando esa plaza esté reservada, que no pueden aparcar en ese espacio. Estos paneles irían interconectados con el servidor central, de forma que los indicadores luminosos puedan recibir la orden de encenderse o apagarse en función del estado de reserva, ocupación o liberación de cada plaza de aparcamiento.

Las administraciones tienen la obligación de proporcionar a la ciudadanía todos los servicios que sean necesarios para mejorar su calidad de vida y en general conseguir una mejor gestión de los recursos con los que cuenta. En este caso, los ayuntamientos, desde su autonomía tienen la posibilidad de implantar este proyecto si consideran que van a mejorar la gestión de sus recursos y la calidad de vida de sus ciudadanos.

Por todo esto, esta propuesta puede ser relevante de cara a la implantación en ciudades, ya que es un proyecto escalable que se podría iniciar como prueba con unas pocas plazas de aparcamiento para comprobar el resultado y posteriormente se podría ir incrementando el número de plazas de aparcamiento que se puedan gestionar de esta forma. El resultado final podría ser que la mayoría de las plazas de aparcamiento disponibles en las zonas más céntricas o concurridas de una ciudad se pudieran reservar y gestionar de esta forma.

1.2 Objetivos del Trabajo

Desarrollar una app para sistemas Android con las siguientes características:

- Sistema de autenticación de usuarios
- Ubicación de plazas de aparcamiento en Google Maps
- Interacción con base de datos en tiempo real
- Reservar una plaza de aparcamiento
- Liberar la plaza de aparcamiento una vez finalizada su ocupación
- Capacidad para encender el panel luminoso de la plaza reservada

1.3 Enfoque y método seguido

Las dos posibles alternativas a la hora de desarrollar una aplicación para móvil consisten en adaptar alguna otra aplicación ya existente o desarrollar una nueva partiendo de cero.

La posibilidad de adaptar una aplicación ya existente podría resultar más rápido y sencillo, sin embargo, también se encuentra la dificultad de encontrar alguna muy parecida a la que se quiere desarrollar, entender el código y saber adaptar la idea de proyecto a esa aplicación.

La posibilidad de desarrollar la aplicación desde el principio se ha considerado más apropiada para un trabajo fin de grado por varios motivos. Se ha considerado más original el desarrollo de un proyecto desde cero, pasando por todas las fases de planificación, estudio,

desarrollo e implementación. Al desarrollar la aplicación desde cero se conoce como está desarrollada toda la lógica del código, de forma que ante cualquier fallo que se detecte o cualquier modificación que se quiera añadir se pueda resolver con mayor facilidad. Se considera importante también poder dar respuesta a cualquier respuesta durante la defensa del proyecto, lo cual es más difícil cuando no se ha desarrollado la aplicación en su totalidad.

El alumno que ha desarrollado este proyecto tiene escasos conocimientos de programación, por tanto, se ha considerado más acertado desarrollar la aplicación utilizando un IDE que le ofreciera las mayores facilidades gráficas posibles, por lo que se ha utilizado el software Android Studio, el cual cuenta con algunas plantillas prediseñadas para facilitar el desarrollo.

Este software al ser tan popular cuenta con mucha documentación de ayuda, así como videotutoriales y guías. Existe también una gran comunidad para dar soporte ante problemas que puedan ir surgiendo. Por tanto, se ha considerado que la estrategia más acertada sería el desarrollo desde cero pero utilizando un software sencillo y con mucho soporte como es Android Studio.

1.4 Planificación del Trabajo

La planificación del tiempo y recursos utilizados para el desarrollo del presente proyecto se han basado principalmente en cumplir los objetivos marcados en cada una de las PEC que se han marcado por el profesor responsable de la asignatura y que han sido las siguientes:

PEC1 desde 16/09/2021 al 29/09/2021 Propuesta de proyecto.

PEC2 desde 30/09/2021 al 27/10/2021 Estado del arte.

PEC3 desde 28/10/2021 al 15/12/2021 Diseño e implementación.

PEC4 desde 16/12/2021 al 05/01/2022 Memoria.

1.5 Breve resumen de productos obtenidos

Tras todo el proceso de desarrollo del TFG se han obtenido:

- 1- Una aplicación para Android
- 2- Memoria del proyecto
- 3- Presentación del proyecto

1.6 Breve descripción de los otros capítulos de la memoria

En el siguiente capítulo 2 - Estado del arte, contiene una breve una comparación de aplicaciones similares existentes en el mercado, entornos de programación para desarrollar la aplicación, bases de datos y el hardware necesario para su implantación.

En el capítulo 3 – Diseño y desarrollo de la aplicación, se ilustra el diseño de la aplicación, incluyendo arquitectura del sistema, diseño de interfaces, diseño de base de datos, de servidor y de seguridad. Además, se enumeran las pruebas realizadas para comprobar el buen funcionamiento de la aplicación.

El capítulo 4 manifiesta unas conclusiones sobre el trabajo realizado y un posible trabajo futuro con el objetivo de plantear mejoras una vez finalizado este Trabajo de Fin de Grado.

Finalmente, los capítulos 5 y 6 relacionan el Glosario y la Bibliografía utilizada en este trabajo respectivamente.

2. Estado del Arte

2.1 Aplicaciones similares

En el mercado existen multitud de aplicaciones para teléfonos inteligentes que facilitan la búsqueda de aparcamiento. Aquí se van a enumerar algunas de las más destacadas y posteriormente se argumentará sobre lo innovador de este proyecto:

- **Google Maps:** Esta herramienta, mundialmente conocida, permite localizar zonas de parking público y privado mediante su sistema de geoposicionamiento.

Además, aunque todavía no está disponible en España, ha desarrollado un método de pago mediante Google Pay para poder realizar el pago por estacionar en las zonas conocidas como ORA (Ordenanza Reguladora de Aparcamiento). La imagen de la Figura 1 está capturada de una imagen en formato gif [1] donde se muestra el proceso de pago integrado en Google Maps.

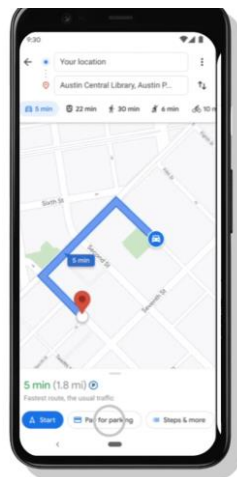


Figura 1. Google play for parking

También, desde la misma web de Google Pay, se ofrece la posibilidad de realizar el pago [2] indicando el número de zona de un aparcamiento regulado ORA.

- **EasyPark:** Es una herramienta que cuenta con app para Android e IOS, además de la página web [3] para buscar zonas de aparcamiento privado, indicando el precio de todos los que hay disponibles en el radio que se le marque. Permite realizar el pago y reserva de plaza de aparcamiento para el día y tramo horario deseado.



Figura 2. EasyPark

Por tanto, mediante esta herramienta, se puede garantizar que cuando el vehículo llegue a su destino tendrá su aparcamiento reservado. Además, cuenta con la función “Find & Park” para ayudar a encontrar aparcamiento en la calle, ya sea de zonas ORA o libre.

- **Parkopedia:** Esta aplicación [4] es una de las mayores del mundo para búsqueda de aparcamiento. Mediante colaboración de usuarios se informa sobre plazas de aparcamiento libres en la calle, y además, la ubicación de parkings públicos y privados. Se premia con puntos para aparcar a los usuarios que colaboran, como se aprecia en la figura 3.



Figura 3. Parkopedia

Se muestran los precios de los parkings y zonas ORA, además también permite la reserva de plaza de aparcamiento en los parkings privados.

- **Aplicación Smart City en Quart de Poblet:** Aunque está limitada al municipio de Quart de Poblet, en Valencia, se destaca esta aplicación porque utiliza hardware externo, de manera que son varios dispositivos los que interaccionan entre sí. Distintas cámaras instaladas en el municipio proporcionan información sobre la ocupación o las plazas libres en distintas zonas, esta información es tratada con inteligencia artificial y recogida en una base de datos que interactúa con la aplicación.

La instalación de este sistema, contratado por el ayuntamiento de un municipio a la plataforma ETRA [5] pone de manifiesto la posibilidad de que otros ayuntamientos puedan mostrar interés por la aplicación que se va a desarrollar en este TFG, así como la viabilidad del desarrollo de una aplicación interconectada con otros elementos que se encuentran repartidos por el municipio.

En conclusión: Existen multitud de aplicaciones tanto conocidas y distribuidas a gran escala como desarrolladas en pequeños municipios que proporcionan la posibilidad de pagar los parquímetros (zonas ORA) y de reservar plazas en aparcamientos privados, pero no se ha encontrado ninguna desarrollada para reservar plaza de aparcamiento en zonas ORA. Este aspecto innovador parte desde la idea de reducir el tráfico en las ciudades, en concreto la de los conductores que circulan buscando aparcamiento en zonas ORA, las cuales son más económicas que los aparcamientos privados y que hasta ahora no tienen la posibilidad de reservar.

2.2 Herramientas de desarrollo para la app

En cuanto al desarrollo de la aplicación para dispositivos Android, existen distintos IDE de programación, entre las que se destacan las siguientes:

- **Android Studio:**



Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan la productividad cuando se desarrollan apps para Android.

- **ECLIPSE:**



Eclipse es un entorno de desarrollo software multi-lenguaje construido alrededor de un workspace al que pueden incluirse un gran número de plug-ins que proporcionan funcionalidades concretas relacionadas con lenguajes específicos o con la interacción con otras herramientas implicadas en el desarrollo de una aplicación. Pese a ser un entorno multi-lenguaje, está desarrollado en Java, siendo el desarrollo en este lenguaje su aplicación principal.

- **NetBeans:**



NetBeans es un entorno de desarrollo integrado libre, orientado principalmente al desarrollo de aplicaciones Java. La plataforma NetBeans permite el desarrollo de aplicaciones estructuradas

mediante un conjunto de componentes denominados “módulos”. Cada uno de estos módulos sería un archivo Java conteniendo un conjunto de clases que interactuarán con las APIs de NetBeans. El objetivo de esta arquitectura es favorecer el desarrollo de funcionalidades de forma independiente y la reutilización de componentes.

- JDeveloper:



JDeveloper es un entorno de desarrollo integrado (IDE) desarrollado por Oracle y disponible de forma gratuita desde 2005. Se trata de un entorno que cubre todo el ciclo de vida del desarrollo software, proporcionando funcionalidad para las fases de diseño, codificación, depuración y despliegue, mediante una aproximación visual y declarativa.

Según las características necesarias para este proyecto, se pueden destacar a estos cuatro IDE, como los más relevantes [6].

Y entre estos cuatro, se considera Android Studio como el más acertado para este proyecto en concreto, ya que cumple los requisitos que se necesitan y además es el que cuenta con el respaldo del propio Google para desarrollar aplicaciones en Android y para la integración con Google Maps.

2.3 Base de datos

Para el almacenamiento de las plazas disponibles, usuarios, sensores, etc. Se necesitará una base de datos dinámica y adaptable a las características que pueda requerir cada ayuntamiento o administración. Además, deberá ser compatible con el entorno de programación elegido para el desarrollo de la aplicación, el cual sería Android Studio, según se ha determinado en el punto anterior.

A continuación, se va a analizar la posibilidad de utilizar distintos sistemas de almacenamiento en bases de datos:

Oracle:



Oracle es una herramienta para la gestión de bases de datos, usada principalmente por las grandes empresas para que estas puedan controlar y gestionar una gran cantidad de contenidos desde un solo archivo.

De esta manera se consigue, además de reducir costes, minimizar el riesgo de pérdida de información.

Entre las acciones de Oracle se pueden destacar las siguientes: soporte de transacciones, estabilidad y soporte multiplataforma de Windows, Linux, Mac OS X, Unix y BDS

MySQL:



MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,¹² y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

SQL Server:



Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

PostgreSQL:



PostgreSQL es un gestor de bases de datos relacional y orientado a objetos. Su licencia y desarrollo es de código abierto, siendo mantenida por una comunidad de desarrolladores, colaboradores y organizaciones comerciales de forma libre y desinteresadamente. Esta comunidad es denominada PDGD (PostgreSQL Global Development Group, por sus siglas en inglés).

Es reconocido actualmente como uno de los sistemas gestores de bases de datos relacionales más potentes del mercado. Presenta fácil accesibilidad, es multiplataforma y está disponible para su utilización en casi todos los sistemas operativos utilizados en la actualidad sin disminuir su rendimiento.

A continuación, se muestra una tabla comparativa con las principales ventajas y desventajas que presentan unos sobre otros:

				
EMPRESA	Oracle Corporation	Sun Microsystems	Microsoft	PostgreSQL Global Development Group
LICENCIA	Privada.	Libre a nivel de usuario, pero para las empresas que quieran incorporarlo en productos privativos deben comprar la licencia.	Privada	Libre
VENTAJAS	<ul style="list-style-type: none"> ✓ Oracle es el motor de base de datos relacional más usado a nivel mundial. ✓ Puede ejecutarse en todas las plataformas, desde un PC hasta una supercomputadora. ✓ Oracle es la Base de datos con mas orientación hacia INTERNET ✓ Soporte de transacciones ✓ Estabilidad ✓ Soporte multiplataforma 	<ul style="list-style-type: none"> ✓ No necesita mucha memoria ram. ✓ Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema ✓ Fácil configuración e instalación. ✓ Múltiples motores de almacenamiento. ✓ Agrupación de transacciones. ✓ Replicación segura. ✓ Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está ✓ Planificación de eventos. ✓ Conectividad segura. ✓ Búsqueda e indagación de datos. ✓ Buenas utilidades de administración. ✓ Buena integración con php. 	<ul style="list-style-type: none"> ✓ Gran facilidad de configuración e instalación. ✓ Utiliza una extensión al SQL estándar, que se denomina TransactSQL ✓ Seguridad: SQL permite administrar permisos a todo. ✓ Permisos a nivel de servidor, seguridad entablas, permitronolectura, etc ✓ Ofrece una potente forma de unir SQL e Internet. ✓ Consultas jerárquicas con selectfrom 	<ul style="list-style-type: none"> ✓ Sistema gratis ✓ Conexión estable. ✓ Permisos a nivel de columna. ✓ Consultas complejas ✓ Integridad transaccional Controlde ✓ Concurrencia (multiversión) ✓ El código fuente está disponible para todos.
DESVENTAJAS	<ul style="list-style-type: none"> * El mayor inconveniente es el precio del producto y de su licencia. * Un Oracle mal configurado es potencialmente lento 	<ul style="list-style-type: none"> * No tiene soporte. * No permite el modo de autenticación local * No sincroniza los datos con otras bases de datos réplicas. 	<ul style="list-style-type: none"> * Requiere una enorme cantidad de memoria RAM para la instalación y utilización del software. * La relación calidad-precio esta muy debajo comparado con Oracle. 	<ul style="list-style-type: none"> * Requiere administradores capacitados. * Lento en comparación con mysql o sql server

Figura 4. Comparativa de Bases de datos

Finalmente, se analiza la base de datos que se ha elegido para el desarrollo de este proyecto, **Firestore**.

¿Qué es Firestore?

Firestore entra dentro de la categoría de los últimamente tan populares SaaS (Software as a Service) que facilitan el desarrollo de software ofreciendo ciertas funcionalidades ya preparadas para ser consumidas por nuestros productos.

Esta herramienta se centra en ofrecer servicios a diferentes niveles.

Desarrollo, Crecimiento y Ganancias constituyen las principales áreas de Firestore y para cada una de ellas ofrece múltiples funcionalidades.

➤ Las funcionalidades ofrecidas dentro de la categoría de desarrollo están enfocadas a facilitar el desarrollo rápido de apps manteniendo una alta calidad.

➤ Las funcionalidades ofrecidas dentro la categoría de crecimiento tienen como objetivo permitir aumentar la base de usuarios sin esfuerzo ni dificultad, nos aportan escalabilidad.

➤ Las funcionalidades ofrecidas dentro de la categoría de ganancias están orientadas a monetizar la app, a obtener beneficios a través de la misma.



Figura 5. Servicios de Firebase

¿Por qué Firebase en este proyecto?

Uno de los pilares básicos sobre los que construir aplicaciones móviles hoy en día es la escalabilidad, a menudo se pueden ver ejemplos de aplicaciones que se viralizan y terminan rehaciendo su aplicación o presentando otra versión de la misma debido a que la inicial no era escalable y no es capaz de afrontar la demanda de usuarios.

Encontrar el éxito y la viralización de una aplicación hoy en día es un tremendo reto en un mercado tan saturado, pero si encima ésta no es escalable, la aplicación habrá muerto antes de ver la luz.

Firebase pertenece a Google, no obstante, no sólo está disponible para Android sino que también para iOS, web y hoy en día también está disponible para plataformas como Unity. Además, la herramienta ofrece la confianza y madurez necesaria contando con el apoyo de un gigante como Google y siendo

integrada en multitud de aplicaciones móviles globalmente utilizadas y líderes de sus mercados.

Permite construir aplicaciones de manera rápida sin tener que preocuparte por la infraestructura, pues utiliza las de Google y además podrás gestionar desde una misma consola todos los servicios que se han implementado en la aplicación. Aunque Google ya ofrecía un servicio de base de datos en tiempo real a través de Firebase, llamado Realtime Database, ahora tiene un nuevo servicio, es una nueva versión del mismo que facilita el escalado global y facilita la sincronización entre dispositivos online y offline, se llama Cloud Firestore y es el que emplearemos en este proyecto.

Herramientas de Firebase que se usan en el proyecto:

- Cloud Firestore: Permite utilizar Firebase como BaaS (Backend as a service) en la aplicación para tener en la nube los datos de manera que se pueda acceder a ellos desde cualquier dispositivo. En definitiva, se emplea como servidor en la nube.
- Google analytics: Ayuda a tomar en el futuro decisiones inteligentes basadas en el comportamiento de los usuarios.
- Gestión de usuarios: Se utiliza el módulo de autenticación para gestionar la misma, editar las plantillas de correos enviados a los usuarios.

Además cuenta con otras herramientas fácilmente implementables como son:

- Notificaciones Push: Que facilita la posibilidad de enviar notificaciones push a grupos de usuarios.
- Integración con servicios de redes sociales: Esta integración con servicios de redes sociales en combinación con la gestión de usuarios para entre otras cosas ofrecer autenticación a través de grandes plataformas como Facebook o Google.

2.4 Hardware necesario

En cuanto al hardware de interacción con las aplicaciones, también se pueden encontrar algunas opciones en el mercado [7], pero las principales, sin duda son:

- **Raspberry Pi:** Se puede considerar como un mini ordenador, desarrollado por la empresa Raspberry Pi Foundation. Cuenta con conectividad WiFi y Ethernet integrada, su SO es de código abierto aunque permite instalar cualquier SO.

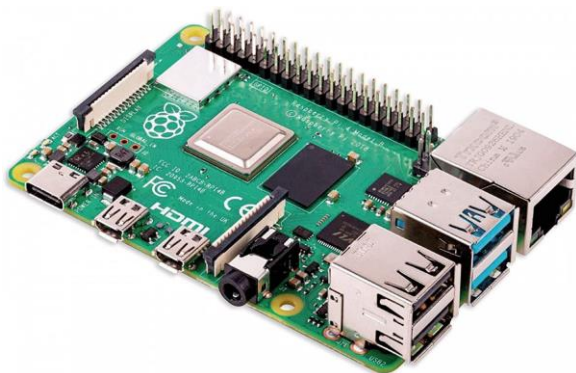


Figura 6. Raspberry

- **Arduino:** Se trata de hardware y software libre, es muy versátil y sencillo de programar aunque carece de potencia de computación. No dispone de conectividad WiFi integrada aunque se puede añadir una expansión.



Figura 7. Arduino

Entre estas dos, se considera Arduino más acertado debido a que cumple perfectamente los requisitos necesarios y hay que tener en cuenta que cada una de las plazas de aparcamiento necesitará una de estas pequeñas placas y aunque en principio, para pequeñas pruebas piloto no supongan un gran reembolso, sí que supondría una gran cantidad de dinero si se instalan de forma masiva.

Otras placas base:

Además de las conocidas Raspberry y Arduino, se pueden encontrar en el mercado otras placas mucho más económicas que podrían cumplir con las necesidades del proyecto pero a un costo muy inferior, como es el caso de la placa EL0112 NodeMCU ESP8266 V3 CH340.

Se trata de un kit de desarrollo de código abierto basado en el popular chip ESP8266 (ESP-12E), que utiliza el lenguaje de programación Lua, no obstante puede programarse con el ID Arduino.

Al integrar el ESP8266 ofrece una solución completa y autónoma de redes Wi-Fi, lo que le permite alojar la aplicación o servir como puente entre Internet y un microcontrolador.



Figura 8. ESP8266

3. Diseño y desarrollo de la aplicación

Tras haber estudiado y profundizado en el marco en el que se va a desarrollar la aplicación, es decir, en los puntos tratados dentro del estado del arte: aplicaciones similares que se encuentran en el mercado, Android Studio como herramienta principal sobre la que se va a trabajar y las distintas bases de datos más usadas y cuál ha sido seleccionada para guardar los datos de la aplicación. Se procede a describir detalladamente a lo largo de este punto.

3.1 Diseño gráfico de la aplicación

Para el diseño de la aplicación se han utilizado los entornos gráficos predefinidos por Android Studio, utilizando las herramientas de edición y diseño con las que cuenta se han diseñado las dos pantallas que utiliza la aplicación, dotándolas de los campos visuales y los botones necesarios para conseguir los objetivos marcados en este proyecto:

Pantalla de Autenticación:

Se muestra el campo "Email" y un campo "Contraseña" donde el usuario deberá introducir su email y contraseña con los que se quiera registrar y pulsar posteriormente en el botón "Registrar" o en el caso de que ya esté registrado pulsando el botón "Acceder".

Es un modelo sencillo ya que el propósito de esta aplicación se centra en otros aspectos pero al haberse desarrollado utilizando Firebase es relativamente sencillo introducir otros métodos de identificación como pueden ser mediante el nombre de usuario y contraseña de Facebook o Google.

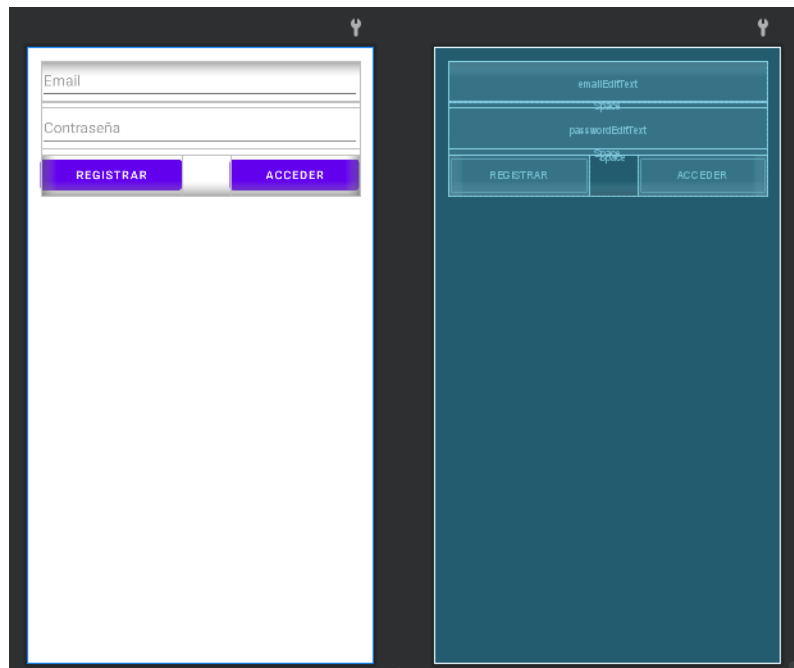


Figura 9. Diseño autenticación

Pantalla principal:

Una vez identificado correctamente el usuario accede a la pantalla principal, desde la cual se muestra un mapa de Google Map en la parte superior de la pantalla para que muestre la ubicación de las plazas de aparcamiento disponibles, debajo un botón para refrescar esta pantalla, llamado “Refrescar” y a continuación debajo un scrollview en el que se muestran las plazas disponibles. Para proceder al listado de las plazas se ha insertado un linearlayout vertical y para cada plaza un linearlayout horizontal con varios campos: 2 de texto para mostrar información de la plaza y un botón para reservar esa plaza de aparcamiento.

Se ha elegido el color verde de fondo para cada plaza para remarcar el estado de disponibilidad de la plaza para poder reservarla. El color de los botones se ha seleccionado en azul por su relación con la “zona azul”.

Cuando una plaza se reserva desaparecen todas las demás de la pantalla y se muestra sólo la plaza que se ha reservado junto a un botón “abandonar” en color rojo, el cual vuelve a dejar libre la plaza ocupada y devuelve la pantalla a la vista anterior en la que se muestran todas las plazas libres.

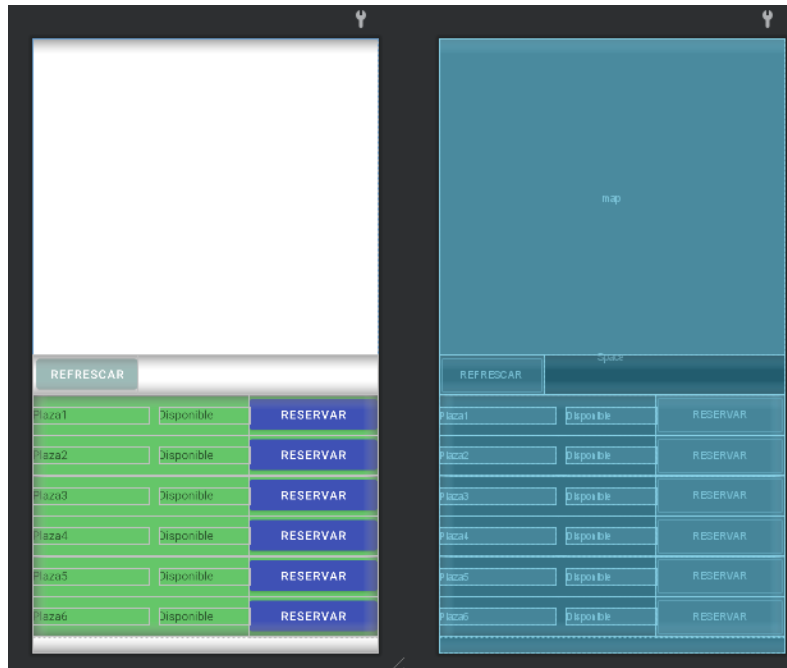


Figura 10. Diseño de pantalla principal

En la siguiente imagen se puede apreciar la estructura utilizada para el diseño de esta pantalla principal.

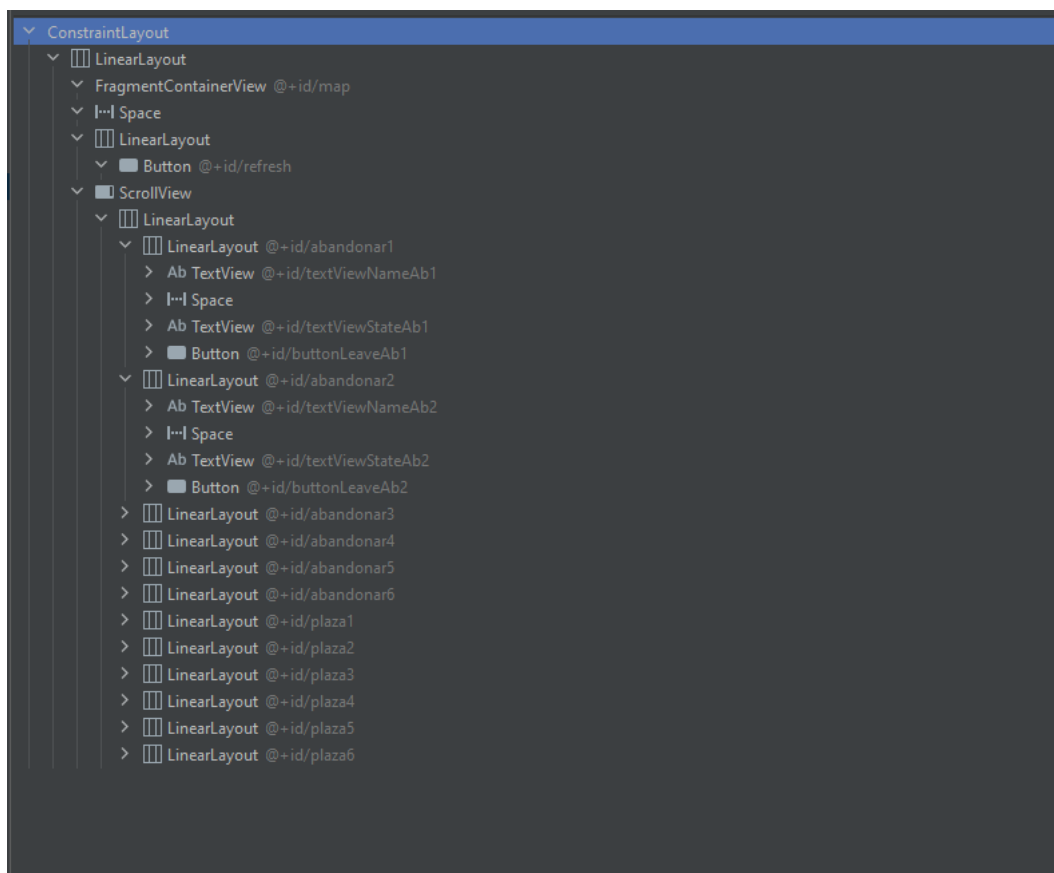


Figura 11. Estructura de la pantalla principal

Finalmente para personalizar la aplicación se ha utilizado un marcador azul con una “P” similar a la que se utiliza en los Parking como icono de la aplicación.

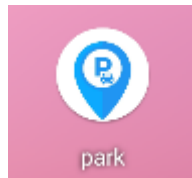


Figura 12. Icono de la aplicación

3.2 Integración de Google Maps

Para poder integra Google Maps en la aplicación se ha utilizado una de las plantillas de Android Studio.

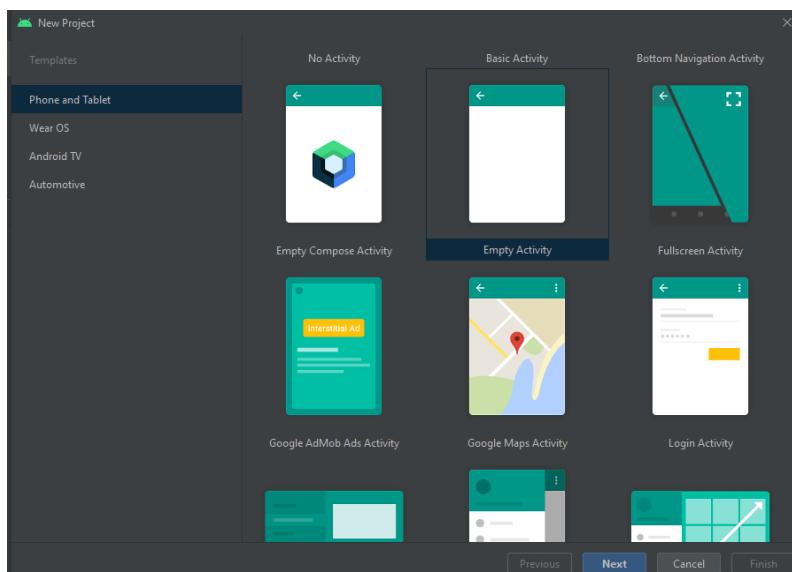


Figura 13. Plantillas de Android Studio

Y se han seguido las instrucciones que aparecen en la guía de la plataforma de Google Map.

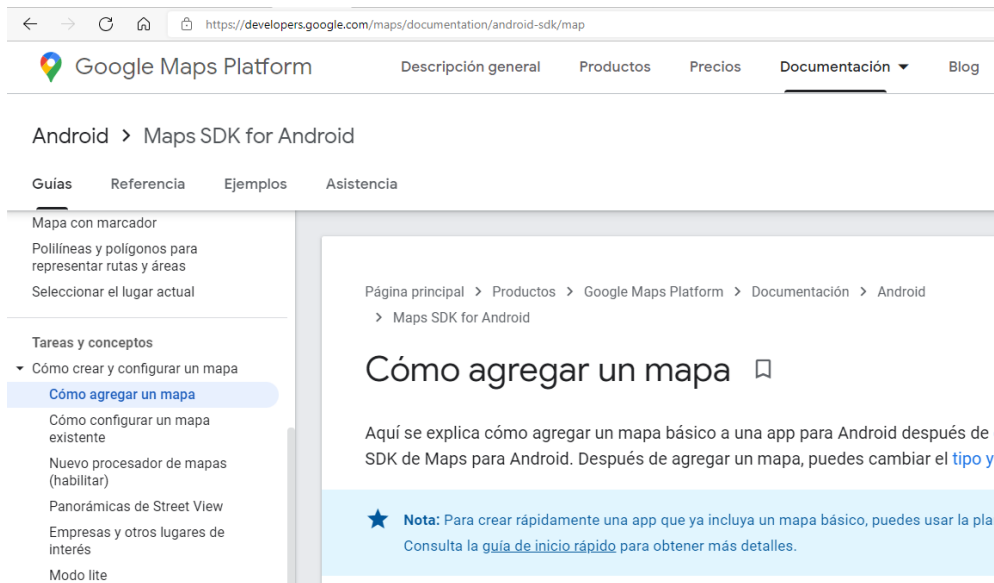


Figura 14. Google Maps Platform

De esta forma se obtiene un identificador único en Google para integrar en la aplicación. Un identificador que debe situarse en el código.

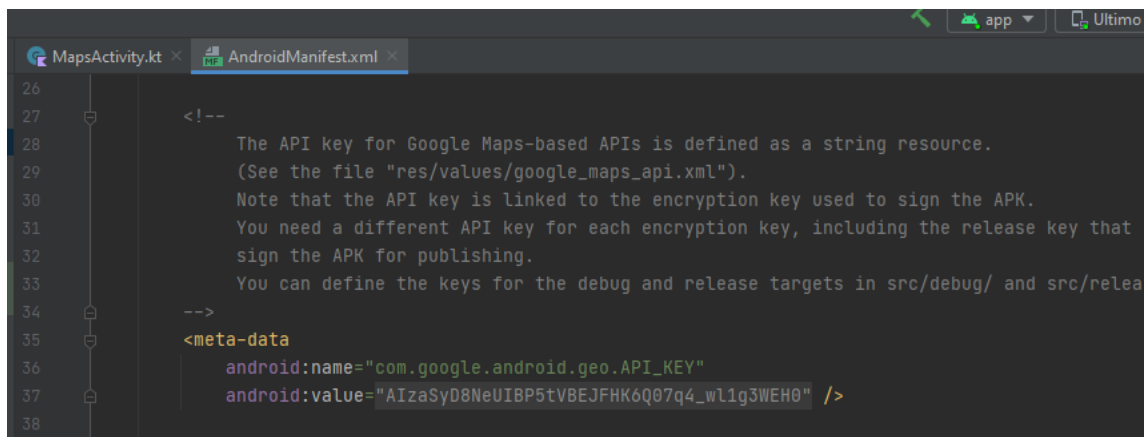


Figura 15. Identificador de Google Maps en AndroidManifest

3.3 Activación de la base de datos

Al tratarse de un producto de Google, para activar los servicios de Firebase tan solo hay que acceder a su web utilizando las credenciales que se utilizan para utilizar cualquier servicio de Google (Google Drive, Gmail...)



Figura 16. Gestión de proyectos en Firebase

Aquí se puede crear un proyecto nuevo y configurar los servicios que se quieren utilizar.

Desde la consola de Firebase se tiene acceso a todas las herramientas disponibles, tal y como se aprecia en la siguiente figura.

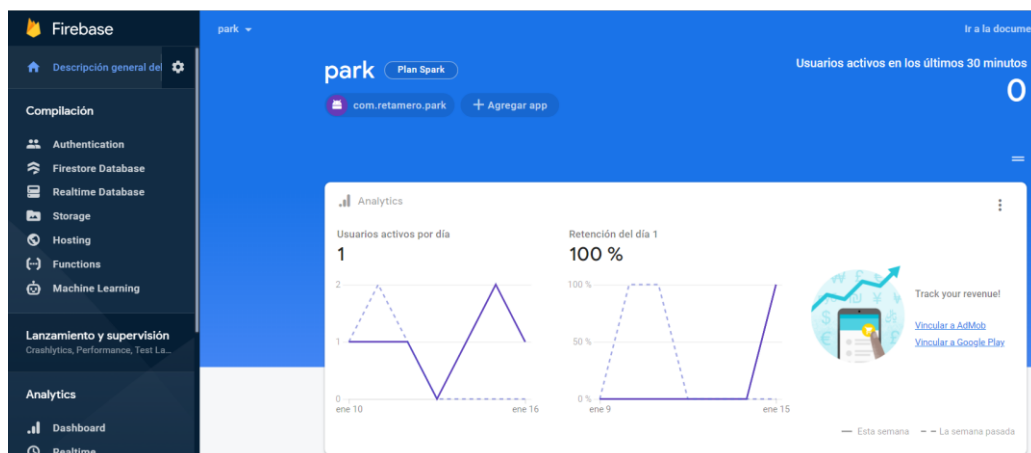


Figura 17. Consola de Firebase

Una de las herramientas que se ha utilizado ha sido Google Analytics, tal y como se muestra en la figura anterior, que se ha integrado en la aplicación para visualizar la información de acceso de los usuarios y que puede ser importante para el futuro desarrollo de la aplicación.

Authentication se ha utilizado para realizar el proceso de autenticación, aunque tan solo se ha utilizado una identificación sencilla mediante email y contraseña, la herramienta permite múltiples configuraciones para configurar la forma de acceso con la que se quiere dotar a la aplicación.

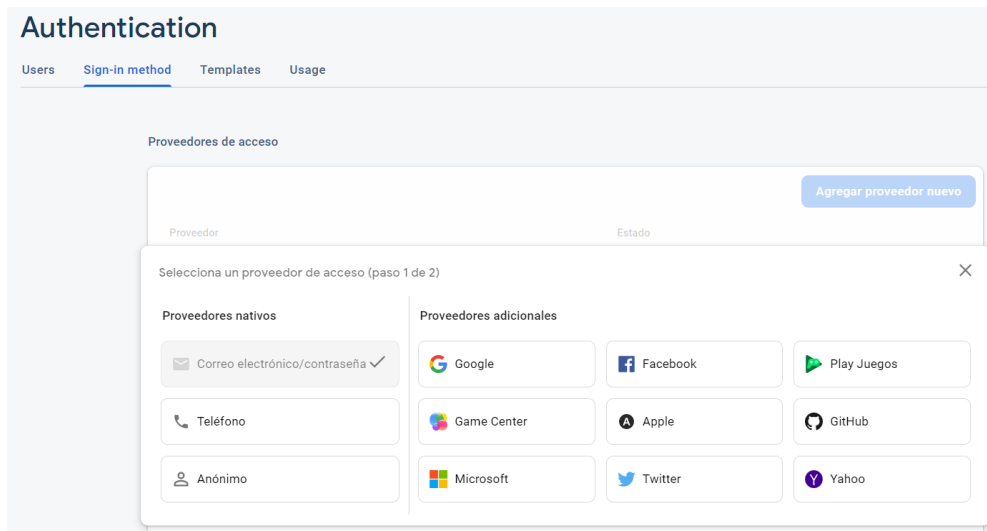


Figura 18. Autenticación en Firebase

En la siguiente figura se muestran los usuarios que se han registrado en la aplicación durante las pruebas.

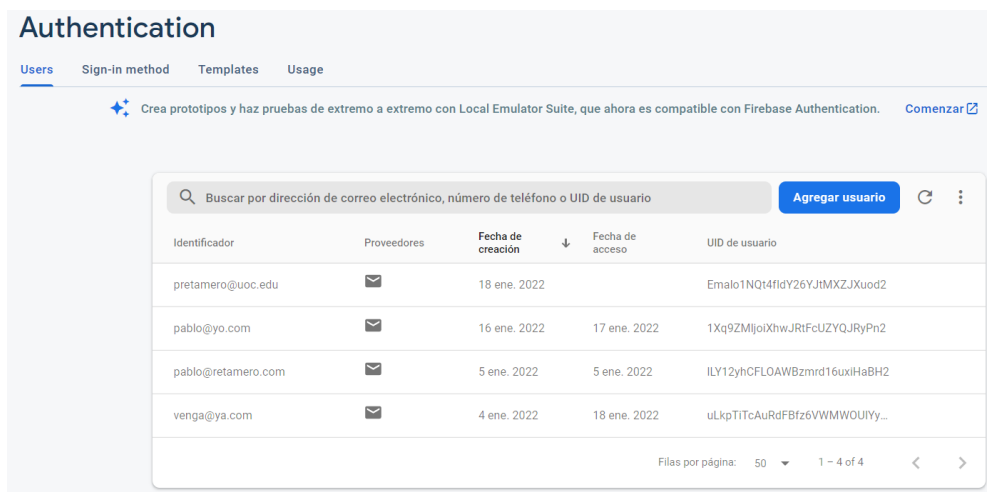


Figura 19. Base de datos de autenticación

Finalmente se muestra el diseño de la base de datos en tiempo real mediante Firestore. Una herramienta sencilla e intuitiva que permite gestionar fácilmente y de forma eficiente la base de datos.

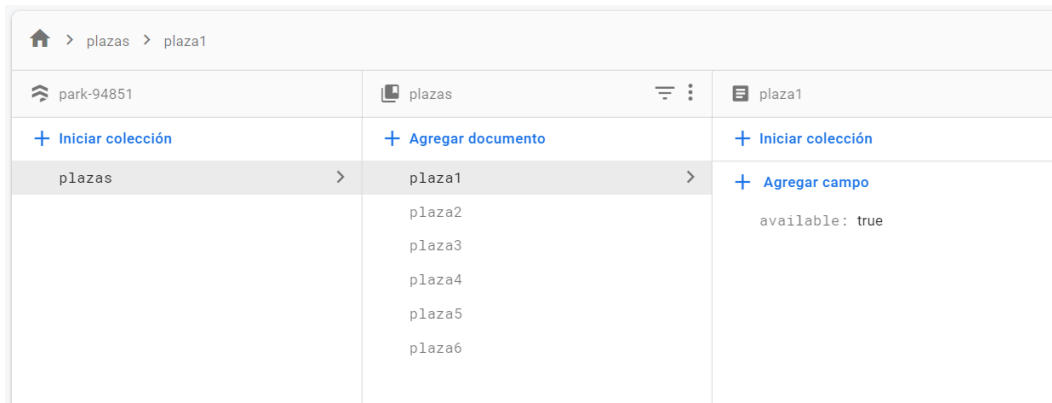


Figura 20. Base de datos de Firestore

Se ha creado la colección “plazas” que contiene cada una de las plazas que se ofrecen en la aplicación, aquí llamadas “documento” y cada una dispone del campo “available” de tipo boolean y que se va cambiando a “true” o “false” en función de si está libre u ocupada respectivamente.

3.4 Presentación de la aplicación

Para empezar a utilizar la aplicación se debe pulsar sobre el icono que se ha descrito en el apartado anterior y que se muestra en el centro de la siguiente figura.

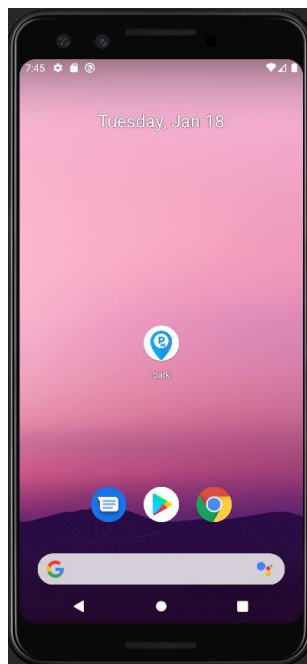


Figura 21. Icono de la aplicación

Al abrir la aplicación se visualiza una pantalla en la que el usuario puede introducir sus credenciales en el caso de tener una cuenta creada previamente o registrarse para poder acceder a la aplicación. Si el usuario tiene una cuenta, deberá introducir su correo electrónico y contraseña y pulsar sobre ACCEDER. Si no tiene cuenta y pulsa sobre REGISTRAR, se mostrará una pantalla que requiere datos para crear una cuenta.

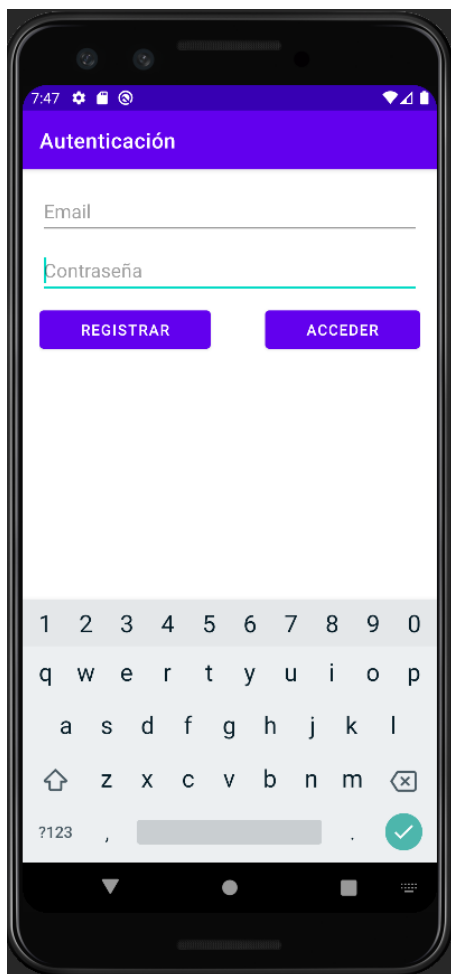


Figura 22. Pantalla de autenticación

Si el usuario no se autentifica correctamente aparecerá un error indicando que el nombre de usuario o contraseña son incorrectos, junto con un botón para aceptar el mensaje y volver a intentarlo.



Figura 23. Error en autenticación

Una vez que el usuario se registra o se identifica correctamente accede a la pantalla principal.

La pantalla principal se ha titulado “Plazas libres en Almería” ya que las plazas de aparcamiento que se han utilizado en este proyecto se han situado en dicha ciudad.

El visor de Google Map aparece centrado en la ubicación de dichas plazas, las cuales aparecen marcadas con marcadores de color verde.

El visor de Google Map es interactivo, de forma que se puede hacer zoom sobre la pantalla para obtener una visión más exacta del lugar en el que se encuentra ubicada cada plaza de aparcamiento.

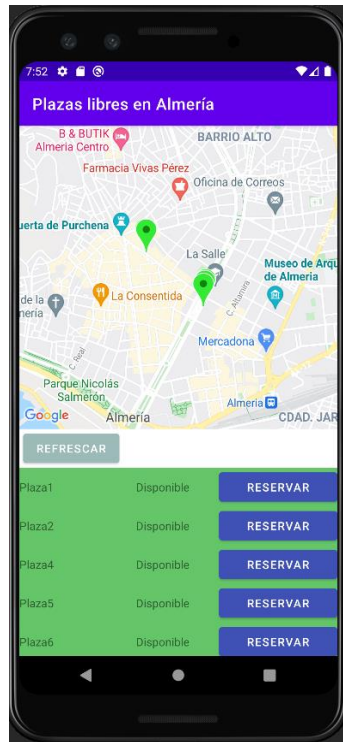


Figura 24. Pantalla principal

Al seleccionar el marcador de cada plaza de aparcamiento aparece un mensaje encima indicando el nombre de la misma así como su disponibilidad.

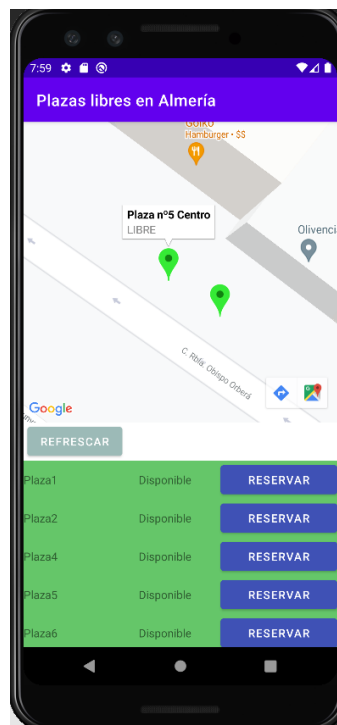


Figura 25. Selección de la plaza en Google Maps

Al pulsar la plaza de aparcamiento que se quiere reservar desaparecen el resto de las plazas de aparcamiento, de forma que no se puede reservar ninguna más y aparece el botón de “abandonar” que se deberá pulsar cuando se deje de ocupar esa plaza de aparcamiento, así volverá a quedar libre para que otro usuario pueda reservarla.

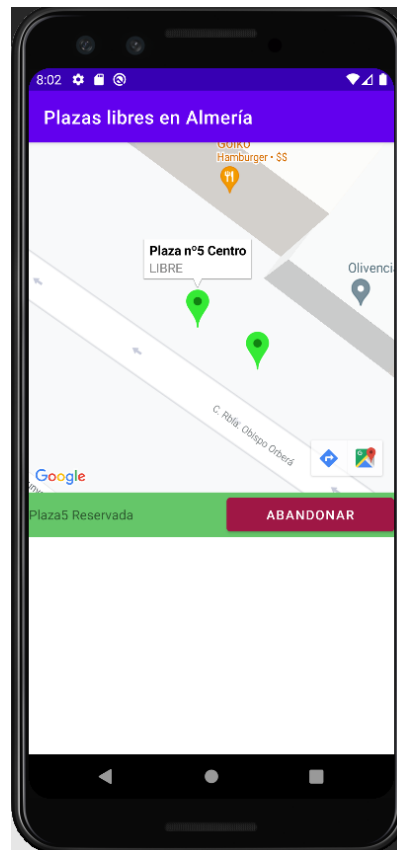


Figura 26. Abandonar la plaza ocupada

3.5 Puesta a prueba

Para poder comprobar el buen funcionamiento de la aplicación, así como su correcta interacción con la base de datos se han realizado pruebas con dos terminales virtuales utilizando la aplicación al mismo tiempo, para comprobar su buen funcionamiento de modo simultáneo.

Test realizados:

1. Autenticación:

Se ha comprobado que los usuarios pueden registrarse correctamente y acceder a la aplicación de forma satisfactoria utilizando su email y contraseña.

2. Interacción con GoogleMap:

Se ha comprobado que se puede hacer zoom en la aplicación y clicar sobre las plazas de aparcamiento para que muestre la información de la misma.

Además, aparece la opción de que se inicie la aplicación GoogleMap que viene instalada por defecto en los dispositivos Android, la cual permite iniciar el navegador para indicar la ruta hasta esa plaza de aparcamiento.

3. Actualización de plazas disponibles cuando se reserva una plaza desde uno de los terminales:

Cuando desde uno de los terminales se reserva una plaza se produce una actualización en tiempo real del estado de la plaza, de forma que la plaza queda en estado de no disponible en la base de datos (Available -> false). Así cuando otro usuario desde otro terminal accede a la aplicación no se le muestra dicha plaza.

De la misma forma, se comprueba que pulsar el botón “abandonar” para liberar esa plaza de aparcamiento, la plaza vuelve a mostrarse a los otros usuarios que estén utilizando la aplicación en ese momento.

Realizadas las pruebas, se concluye que la aplicación en todos los casos se muestra estable y responde satisfactoriamente en todos los casos.

4. Conclusiones

El desarrollo de aplicaciones multimedia cada vez cuenta con herramientas más sencillas de utilizar para facilitar la creación de aplicaciones, sin embargo, resulta imprescindible contar con conocimientos de programación cuando se quiere implementar cualquier aspecto que se encuentre fuera de las plantillas prediseñadas.

Aunque existen múltiples manuales que describen detalladamente cada librería es imposible llegar a estudiar tales materiales tan extensos durante el desarrollo del TFG por lo que se hace imprescindible encontrar tutoriales o videotutoriales en YouTube que vayan enfocados a lo que se quiere conseguir directamente.

Gran parte de los objetivos planteados inicialmente se han conseguido, principalmente se ha conseguido finalizar la aplicación y que funcione, aunque ha faltado la parte de la integración con Arduino para hacer que se encendiera o apagara la luz cuando la plaza se reservara o se quedara libre. De todas formas, esta parte tendría que estudiarse más en profundidad con la administración que estuviera dispuesta a implementar este proyecto en su ayuntamiento, ya que habría que adaptar el tipo de hardware utilizado al presupuesto de dicha administración.

Se ha intentado seguir la planificación del producto aunque se han producido retrasos en algunos puntos debido a las complicaciones surgidas con la programación de la aplicación.

La línea de trabajo futuro sería la implantación de otros métodos de identificación en la plataforma, una optimización de la aplicación para que interactúe más con la base de datos, el desarrollo de una pantalla de administración para agregar nuevas plazas de aparcamiento y finalmente la integración del hardware que interactúe con la aplicación para encender y apagar las luces de las plazas de aparcamiento.

El hardware que se instale en las plazas de aparcamiento dependerá del presupuesto disponible para ejecutar el proyecto, pero principalmente consistirá en:

- 1- Un panel luminoso para indiciar claramente a otros conductores que esa plaza ya está reservada y no pueden aparcar en ella. Este panel se encenderá en el momento que se pulse el botón "Reservar" en la aplicación, se apagará cuando el usuario que la ha reservado la ocupe y se pondrá en verde cuando la plaza esté libre de nuevo.

2- Un sensor para reconocimiento de matrícula mediante OCR para detectar la presencia de un vehículo en la plaza y poder identificarlo. Mediante el reconocimiento de matrícula se podría verificar que el usuario que ha aparcado es realmente el que ha reservado la plaza. Si otro usuario aparcase en ella el panel se mantendría encendido o incluso se podría configurar la luz para que luciera de forma intermitente, para que de esta forma el conductor se diera cuenta de que no puede ocupar esa plaza.

Cuando el usuario que ha reservado la plaza la abandone, el sistema OCR detectaría que el vehículo ya no se encuentra ahí, de forma que mandaría una orden a la base de datos para volver a dejar esa plaza disponible.

Para la instalación de este hardware se necesitaría una toma de electricidad de 230 V para alimentación y una estructura que proteja las luces y los sensores que se instalen.

La conectividad de este hardware con la aplicación y la base de datos se conseguiría mediante módulos de conexión wifi en la placa base del dispositivo que se instale. En un lugar próximo se instalaría un router con una tarjeta SIM para conseguir conexión con Internet. Así, cada placa base (a la que van conectadas las luces y sensores) se conectaría por wifi con el router y éste con la base de datos a través de Internet.

5. Glosario

- Smartphone: Anglicismo. Teléfono móvil construido sobre una plataforma informática móvil, con capacidad para almacenar datos y realizar actividades asemejándose a una minicomputadora.
- Android: Sistema operativo basado en linux para dispositivos móviles.
- Linux: Combinación de kernel libe con sistema GNU.
- GNU: Sistema operativo desarrollado por GNU, formado en su totalidad por software libre.
- API: Siglas de Interfaz de Programación de Aplicaciones. Es un conjunto de funciones, métodos o procedimientos que ofrece una biblioteca para ser utilizado por otro software como una capa de abstracción.
- Servicio Web: Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
- Recurso: Datos (arrays, imágenes, cadenas, números, estilos) que se almacenan independientemente del código de aplicación para mejorar la organización del proyecto y permitir su adaptación a cambios.
- Cliente: Aplicación informática que consume un servicio remoto de otro ordenador conocido como servidor.
- Servidor: Nodo que forma parte de una red y provee de servicios a nodos denominados clientes.
- Script: Archivo de órdenes, archivo de procesamiento por lotes o guión es un programa usualmente simple, que por lo regular se almacena en archivos de texto plano.
- Multiplataforma: Válido para varios sistemas

6. Bibliografía

[1] J.I. Levy, J.J. Buonocore y K. von Stackelberg, "Evaluation of the public health impacts of traffic congestion: a health risk assessment", *Environ Health*, nº 9, 2010. [En línea].

Disponible en: <https://doi.org/10.1186/1476-069X-9-65> Acceso: septiembre2021.

[2] K.Zhang y S. Batterman, "Air pollution and health risks due to vehicle traffic", *Science of The Total Environment*, vol. 450–451, pp. 307-316, 201. [En línea].

Disponible en: <https://doi.org/10.1016/j.scitotenv.2013.01.074> Acceso: septiembre2021.

[3] J.M. Moreira Robles y S.E. Pusedá Chulde, "Metodología de selección de IDE para el desarrollo de aplicaciones Android", *Revista Ibérica De Sistemas e Tecnologías De Informação*, nºE15, pp. 214-220, abr. 2018.

[4] R.P.P. Dinkar, "A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board", *International Journal of Advanced Research in Computer Science*, vol. 8, nº 5, pp. 2350-2352, 2017. [En línea].

Disponible en: <http://dx.doi.org/biblioteca-uoc.idm.oclc.org/10.26483/ijarcs.v8i5.3959>

Acceso: septiembre2021.

Custom Activity transition when up button is pressed - Stack Overflow. (n.d.).

Retrieved January 7, 2022, from

<https://stackoverflow.com/questions/40882733/custom-activity-transition-when-up-button-is-pressed>

android-guidelines: Architecture and code guidelines we use at ribot when developing for Android. (2022). ribot. Retrieved from <https://github.com/ribot/android-guidelines> (Original work published 2014)

Announcing Architecture Components 1.0 Stable. (n.d.). Acceso: noviembre2021.

From <https://android-developers.googleblog.com/2017/11/announcing-architecture-components-10.html>

Arquitectura Clean de Uncle Bob. (2016, January 26). Retrieved January 5, 2022, from <http://imperezramos.net/programacion/arquitectura-clean-de-uncle-bob/>

Bauer, V., & Bauer, V. (n.d.). RecyclerViewTemplate | Android-Arsenal.com. Acceso diciembre2021, from <https://android-arsenal.com/details/1/6204>