

Santiago Hernández - DAM 2 Dual

# PUNTOS KILOMÉTRICOS APP

Autor: Jorge Nicolás Magallón Cuartero.

Tutores: Edgar Bernáldez y Pilar Lozano.



## Índice

|   |    |
|---|----|
| Abstract .....  | 3  |
| Resumen.....  | 4  |
| Datos descriptivos .....  | 6  |
| Descripción general: .....  | 6  |
| Marco teórico: .....  | 6  |
| Principios tecnológicos y normas técnicas aplicables: .....       | 7  |
| Análisis de la realidad: Entorno de la empresa colaboradora ..... | 7  |
| Análisis del sector: .....  | 7  |
| Análisis de la empresa colaboradora: .....                        | 8  |
| Análisis de la colaboración: .....                                | 9  |
| Justificación. ....   | 9  |
| Destinatarios. ....   | 11 |
| Desarrollo del proyecto .....                                     | 12 |
| Acuerdo de proyecto .....   | 12 |
| Requisitos funcionales .....                                      | 12 |
| Requisitos no funcionales .....                                   | 12 |
| Actividades y cronología .....                                    | 14 |
| Diagrama de Gantt .....   | 14 |
| Documento de análisis y diseño.....                               | 15 |
| Pruebas del sistema .....   | 30 |
| Documentación para el cliente.....                                | 32 |



|  |           |
|--|-----------|
| Metodología .....  | 33        |
| Recursos .....   | 34        |
| Presupuesto .....  | 35        |
| Gastos.....  | 35        |
| Ingresos .....   | 35        |
| Beneficio .....  | 36        |
| Resultados obtenidos y conclusiones.....                                       | 37        |
| <b>Referencias bibliográficas .....</b>  | <b>38</b> |
| ANEXO.....   | 41        |
| Migración de datos.....  | 41        |
| Pruebas de peticiones a la API de Google maps y Google roads con POSTMAN:..... | 46        |
| Test unitarios de la clase JsonServiceTest.java .....                          | 48        |

**Abstract**

This thesis presents the innovation in the development of a mobile application designed to provide geolocation and roadside assistance services, addressing a need that is not covered by existing applications in the market, including Google Maps and other similar apps. The main motivation behind this project is the lack of detailed information about kilometric points in existing applications. The developed application fills this gap by allowing users to identify and search for specific kilometric points along the roads.

To achieve this, extensive research was conducted to obtain reliable data about kilometric points and understand how to migrate and utilize this data on the Android platform, in conjunction with the services provided by Google. Reliable sources of information about kilometric points were investigated, and best practices were applied to migrate and adapt the data to the database structure used in the application.

Furthermore, Google services such as geolocation and related services were leveraged to ensure the accuracy and functionality of the application. This enabled effective identification of the user's location and display of the nearest kilometric points. Research and the application of technical knowledge were crucial in achieving successful implementation and providing an optimal user experience within the application.

In summary, this thesis highlights the innovation in the development of a mobile application that addresses the lack of information about kilometric points in existing applications. Extensive research was conducted to obtain and migrate the necessary data, and Google services were utilized to ensure the application's functionality.



Esta tesis presenta la innovación en el desarrollo de una aplicación móvil diseñada para proporcionar servicios de geolocalización y asistencia vial, abordando una necesidad que no está cubierta por las aplicaciones existentes en el mercado, incluyendo Google Maps u otras aplicaciones similares. La principal motivación detrás de este proyecto es la falta de información detallada sobre los puntos kilométricos en las aplicaciones existentes. La aplicación desarrollada llena este vacío al permitir a los usuarios conocer y buscar puntos kilométricos específicos a lo largo de las carreteras.

Para lograr esto, fue necesario realizar una investigación exhaustiva para obtener datos confiables sobre los puntos kilométricos y comprender cómo migrar y utilizar estos datos en la plataforma Android, junto con los servicios proporcionados por Google. Se investigaron fuentes confiables de información sobre los puntos kilométricos y se aplicaron las mejores prácticas para migrar y adaptar los datos a la estructura de la base de datos utilizada en la aplicación.

Además, se aprovecharon los servicios de Google, como la geolocalización y otros servicios relacionados, para garantizar la precisión y funcionalidad de la aplicación. Esto permitió identificar la ubicación del usuario y mostrar los puntos kilométricos más cercanos de manera efectiva. La investigación y la aplicación de conocimientos técnicos fueron fundamentales para lograr una implementación exitosa y proporcionar una experiencia óptima para los usuarios de la aplicación.

En resumen, esta tesis destaca la innovación en el desarrollo de una aplicación móvil que aborda la falta de información sobre puntos kilométricos en las aplicaciones existentes. Se realizó una investigación exhaustiva para obtener y migrar los datos

PUNTOS KILOMETRICOS

DAM 2 Dual – Santiago Hernández



necesarios, y se aprovecharon los servicios de Google para garantizar la funcionalidad de la aplicación.



**Descripción general:**

Esta tesis se enfoca en el desarrollo de una aplicación móvil para la geolocalización y asistencia en carretera. La aplicación permitirá a los usuarios conocer su ubicación exacta en la carretera, utilizando la API de Google Maps. Además, la aplicación también contará con la funcionalidad de llamar a emergencias y al seguro del vehículo en caso de accidente. También habrá una funcionalidad que permitirá al usuario buscar puntos kilométricos, tanto el primero, como el ultimo como uno en concreto y todos los de una carretera.

La aplicación se desarrollará para dispositivos móviles con sistema operativo Android, y contará con una base de datos local para el almacenamiento de información de los usuarios.

**Marco teórico:**

El marco teórico de esta tesis se basará en la revisión de diferentes estudios y trabajos relacionados con la API de Google Maps y su integración en aplicaciones móviles. También se revisarán conceptos relacionados con la asistencia en carretera, como la llamada a emergencias y al seguro del vehículo.

Además, se abordarán temas relacionados con el desarrollo de aplicaciones móviles, como la utilización de diferentes herramientas y lenguajes de programación para la creación de interfaces de usuario y la conexión con servicios externos.

**Principios tecnológicos y normas técnicas aplicables:**

Para el desarrollo de la aplicación, se utilizará la API de Google Maps para obtener la ubicación exacta del usuario y la dirección de la carretera en la que se encuentra. Asimismo, se utilizarán diferentes herramientas y tecnologías para la integración de la API y la conexión con servicios externos, como la realización de llamadas telefónicas.

Para la base de datos local, se utilizarán herramientas y tecnologías de almacenamiento de datos para dispositivos móviles, como SQLite.

En cuanto a las normas técnicas, se aplicarán estándares como la norma ISO/IEC 12207 para la gestión del ciclo de vida del software, la norma ISO/IEC 9126 para la evaluación de la calidad del software, y la norma ISO 26262 para la seguridad funcional en sistemas electrónicos en el automóvil.

**Análisis de la realidad: Entorno de la empresa colaboradora**

Para el desarrollo de la aplicación móvil de geolocalización y asistencia en carretera se ha establecido una colaboración con una empresa de seguros de automóviles. A continuación, se realizará un análisis del entorno de la empresa colaboradora.

**Análisis del sector:**

El sector de seguros de automóviles es un mercado altamente competitivo y dinámico. Las empresas en este sector se enfrentan a una gran presión por parte de





los consumidores para ofrecer productos y servicios de alta calidad a precios competitivos. Además, la presión regulatoria en el sector ha aumentado en los últimos años, lo que ha llevado a una mayor atención a la calidad y seguridad de los productos y servicios ofrecidos.

#### **Análisis de la empresa colaboradora:**

La empresa colaboradora es una empresa de seguros de automóviles que ha estado en el mercado durante más de 10 años. La empresa cuenta con una amplia red de oficinas y agentes en todo el país y ha desarrollado una sólida reputación en el sector.

La empresa colaboradora ha demostrado un fuerte compromiso con la innovación y la mejora de la experiencia del cliente. La colaboración en el desarrollo de una aplicación móvil para la geolocalización y asistencia en carretera es un reflejo de este compromiso.

La empresa colaboradora cuenta con una base sólida de clientes y una amplia experiencia en el sector de seguros de automóviles. Esto significa que la empresa tiene una comprensión sólida de las necesidades de los clientes y está bien posicionada para desarrollar productos y servicios que satisfagan estas necesidades.

**Análisis de la colaboración:**

La colaboración entre la empresa colaboradora y el equipo de desarrollo de la aplicación móvil tiene como objetivo desarrollar una aplicación móvil de alta calidad para la geolocalización y asistencia en carretera. La empresa colaboradora aportará su conocimiento y experiencia en el sector de seguros de automóviles, mientras que el equipo de desarrollo de la aplicación móvil aportará su experiencia en el desarrollo de aplicaciones móviles y su conocimiento de las últimas tecnologías.

La colaboración permitirá la creación de una aplicación móvil que satisfaga las necesidades de los clientes de la empresa colaboradora y proporcione una experiencia de usuario de alta calidad. Esto permitirá a la empresa colaboradora mejorar su posición en el mercado y ofrecer un producto innovador y atractivo a sus clientes.

**Justificación.**

La necesidad de una aplicación móvil que pueda geolocalizar el punto kilométrico en el que se encuentra un conductor en una carretera ha sido identificada debido a la falta de una solución efectiva en el mercado actual. A pesar de que existen aplicaciones de navegación y mapas que pueden proporcionar una ubicación general, ninguna de ellas proporciona información precisa sobre el punto kilométrico en el que se encuentra el usuario.

Esta información es importante para los conductores, especialmente en situaciones de emergencia en la carretera, ya que puede ser crítica para pedir



una situación de emergencia.

La innovación propuesta por la aplicación móvil de geolocalización y asistencia en carretera es su capacidad para proporcionar información precisa sobre el punto kilométrico en el que se encuentra el usuario, lo que la convierte en una solución única en el mercado. Además, la integración de la API de Google Maps y la capacidad de llamar a emergencias y al seguro del vehículo en caso de accidente proporcionan una experiencia de usuario completa y un mayor nivel de seguridad en la carretera.



La aplicación de geolocalización y asistencia en carretera no solo es útil para los conductores en la carretera, sino que también puede ser beneficiosa para otros usuarios. En particular, la aplicación puede ser una herramienta útil para la Guardia Civil de Tráfico, los servicios de emergencias y los geógrafos.

En primer lugar, la aplicación puede ser de gran ayuda para la Guardia Civil de Tráfico en la elaboración de informes de accidentes en carretera. Al proporcionar información precisa sobre la ubicación del accidente, incluyendo el punto kilométrico exacto, la aplicación puede ayudar a la Guardia Civil de Tráfico a determinar las causas del accidente y tomar medidas para evitar futuros accidentes en la misma zona.

En segundo lugar, la aplicación también puede ser útil para los servicios de emergencias, como los bomberos o las ambulancias, que necesitan localizar rápidamente la ubicación exacta de un accidente o incidente en la carretera. Al proporcionar información precisa sobre la ubicación, la aplicación puede ayudar a los servicios de emergencia a llegar al lugar del incidente rápidamente y proporcionar asistencia vital a las personas involucradas.

Por último, la aplicación también puede ser beneficiosa para los geógrafos que trabajan en la definición y mapeo de los terrenos colindantes a las carreteras. Al proporcionar información precisa sobre la ubicación, la aplicación puede ayudar a los geógrafos a validar y verificar sus datos, lo que puede ser crucial para el desarrollo de mapas precisos y actualizados.



## Acuerdo de proyecto

### Requisitos funcionales

**Geolocalización:** La aplicación permitirá a los usuarios conocer su ubicación exacta en la carretera utilizando la API de Google Maps. Los usuarios podrán visualizar su posición en el mapa dentro de la aplicación móvil.

**Asistencia en carretera:** La aplicación proporcionará funcionalidad para llamar a emergencias en caso de accidente. Los usuarios podrán contactar al seguro del vehículo a través de la aplicación en caso de accidente.

**Búsqueda de puntos kilométricos:** La aplicación permitirá al usuario buscar el punto kilométrico inicial de una carretera. La aplicación permitirá al usuario buscar el punto kilométrico final de una carretera. Los usuarios podrán buscar un punto kilométrico específico en una carretera. La aplicación ofrecerá la opción de mostrar todos los puntos kilométricos de una carretera.

### Requisitos no funcionales

La aplicación debe utilizar la API de Google Maps para permitir a los usuarios conocer su ubicación exacta en la carretera. Se deben solicitar los permisos de ubicación al usuario para acceder a esta funcionalidad. La aplicación debe utilizar la dependencia `com.google.android.gms:play-services-maps:18.1.0` para integrar la API de Google Maps. La aplicación debe ofrecer la funcionalidad de llamar a emergencias en caso de accidente. También se debe incluir la opción de llamar al seguro del vehículo en situaciones de emergencia. Se requiere acceso a internet para realizar estas llamadas. Se puede utilizar la dependencia `com.karumi:dexter:4.2.0` para solicitar permisos en tiempo de ejecución. La aplicación debe permitir al usuario buscar puntos kilométricos, incluyendo el punto kilométrico inicial, final, todos y uno



específico. Se utilizará una base de datos SQLite que contendrá todos los puntos

kilométricos. Se puede utilizar la dependencia

`com.readystatesoftware.sqliteasset:sqliteassethelper;2.0.1` para trabajar con la base de datos SQLite.

La aplicación se desarrollará para dispositivos móviles con sistema operativo Android. Se utilizará el SDK con nivel mínimo 25 y máximo 33. La aplicación está escrita en Java 8. Se utilizará Gradle 7.4 como herramienta de construcción. El desarrollo se realizó utilizando un emulador Pixel 2 XL con la API 25.

Se utilizará la dependencia `com.google.android.gms:play-services-location:21.0.1` para la funcionalidad de ubicación. Se utilizará la dependencia `org.apache.commons:commons-lang3:3.12.0` de Apache Commons Lang.

Para utilizar la API de Google Maps, la aplicación necesitará una clave de API válida asociada a una cuenta de Google. La clave de API proporcionará acceso a los servicios de Google Maps utilizados en la aplicación, como la geolocalización y la visualización de mapas. Se utilizará la dependencia `com.android.volley:1.2.1` para realizar las solicitudes a la api deGoogle.

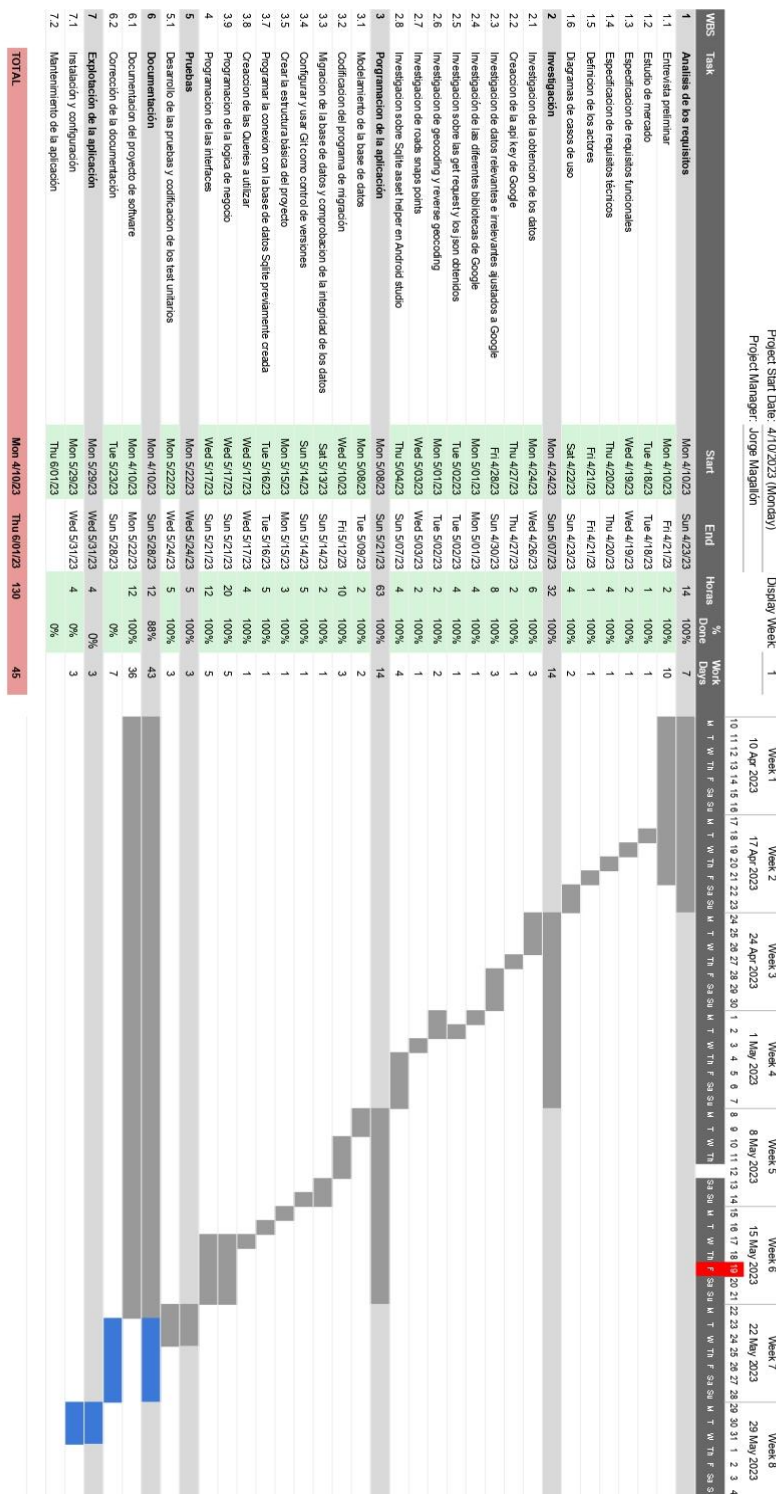


# PUNTOS KILOMETRICOS

## Actividades y cronología

DAM 2 Dual – Santiago Hernández

### Diagrama de Gantt

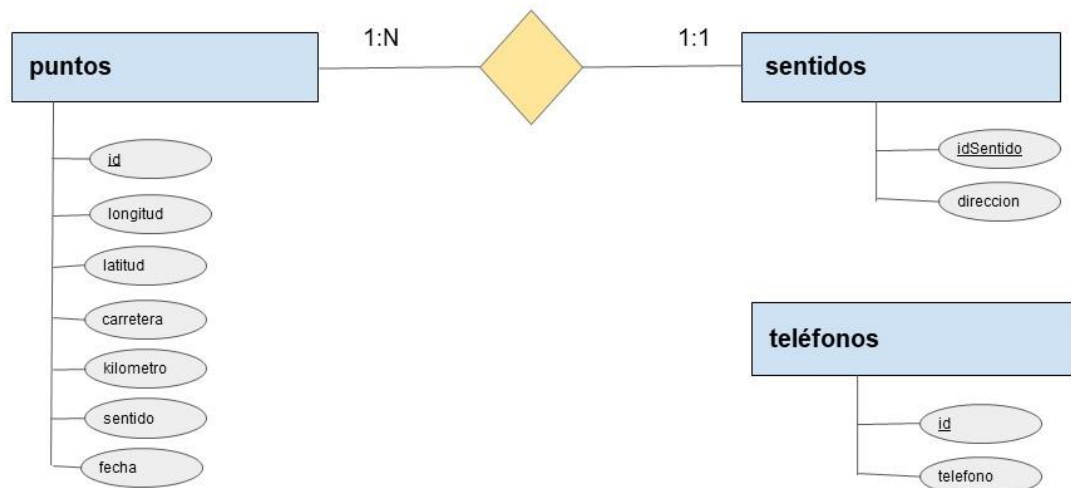




### *Base de datos*

Para la base de datos, se llevó a cabo la obtención de datos mediante la plataforma web de Esri España quienes son una empresa tecnológica, innovadora, orientada al desarrollo de soluciones y servicios en el marco de los Sistemas de Información Geográfica (SIG).

A través de su Open Data Source, se adquirió un fichero CSV que contenía todos los puntos kilométricos de España, actualizados a fecha del año 18 de mayo de 2022. Para convertir el fichero CSV en una base de datos, primero, se creó un diagrama entidad relación y un diagrama relacional:







Para la gestión de estos datos, se creó un script de base de datos que incluía

los campos y tablas necesarios:

```

CREATE TABLE sentidos (
    idSentido INTEGER PRIMARY KEY,
    direccion VARCHAR(20)
);

CREATE TABLE puntos (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    longitud REAL(6) NOT NULL,
    latitud REAL(6) NOT NULL,
    carretera VARCHAR(150) NOT NULL,
    kilometro INTEGER NOT NULL CHECK(kilometro >= 0),
    sentido INTEGER NOT NULL,
    fecha DATE NOT NULL,
    FOREIGN KEY (sentido) REFERENCES sentidos(idSentido)
);

CREATE TABLE telefonos (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    telefono char(9)
);

INSERT INTO sentidos (idSentido, direccion) VALUES (1, 'Creciente');
INSERT INTO sentidos (idSentido, direccion) VALUES (2, 'Decreciente');
INSERT INTO sentidos (idSentido, direccion) VALUES (3, 'Ambos sentidos');
  
```

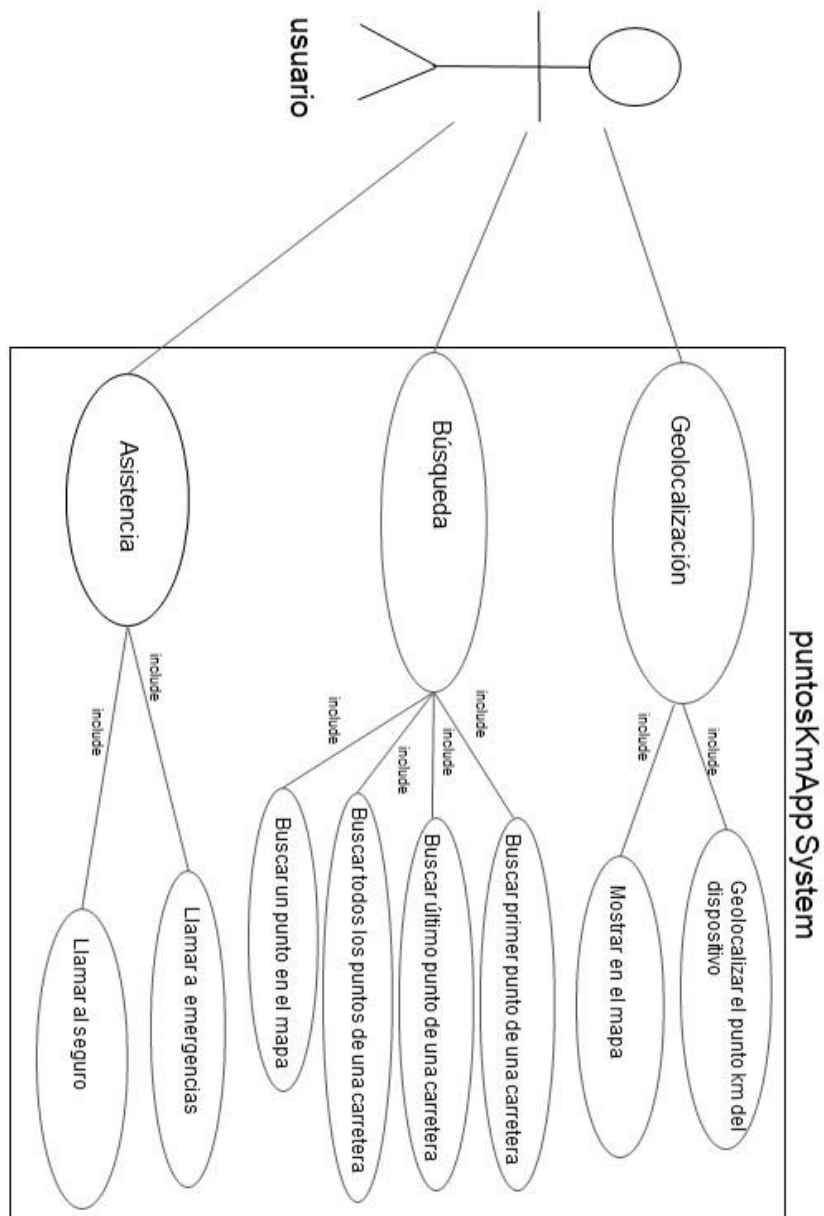


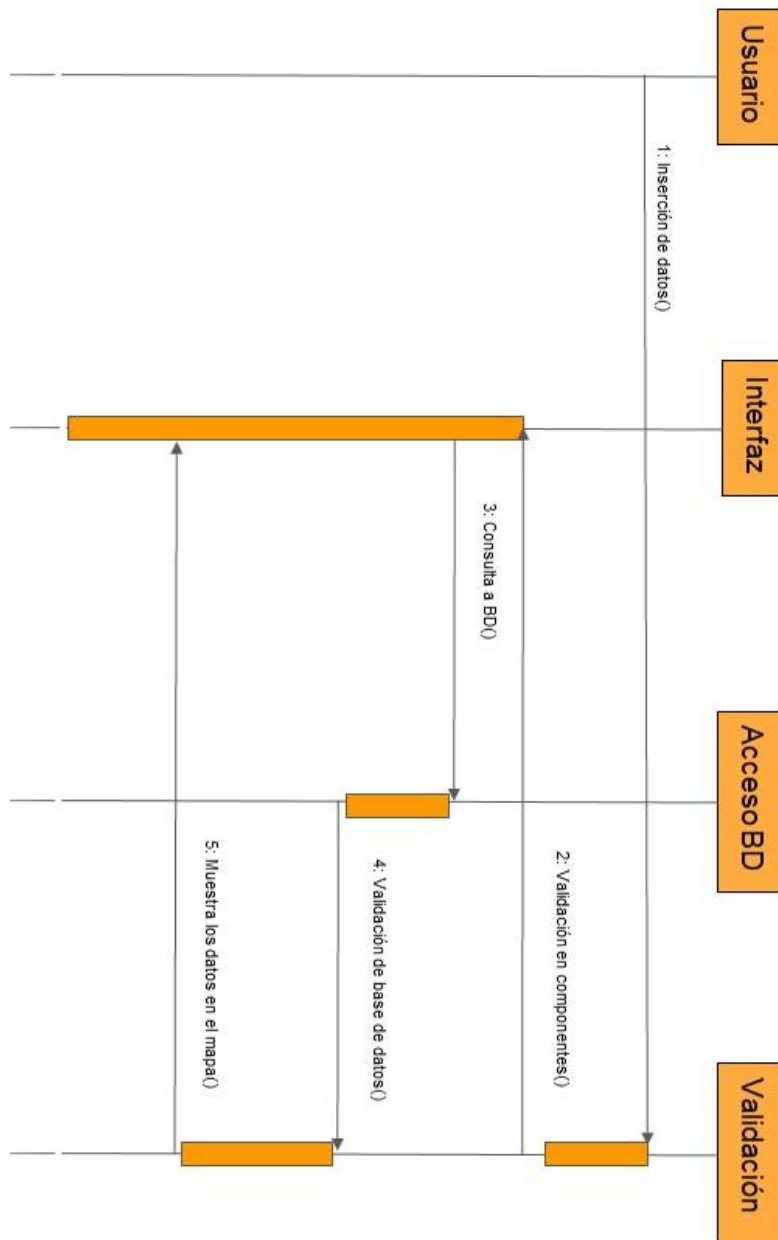
Mediante una migración de datos que se adjunta en el anexo, se procedió a importar los datos del fichero CSV a la base de datos SQLite desarrollada específicamente para esta aplicación, llevando a cabo de 191169 registros del fichero csv, 191062 migrados y 107 que no ya que no disponían del nombre de carretera, requisito fundamental para este proyecto.

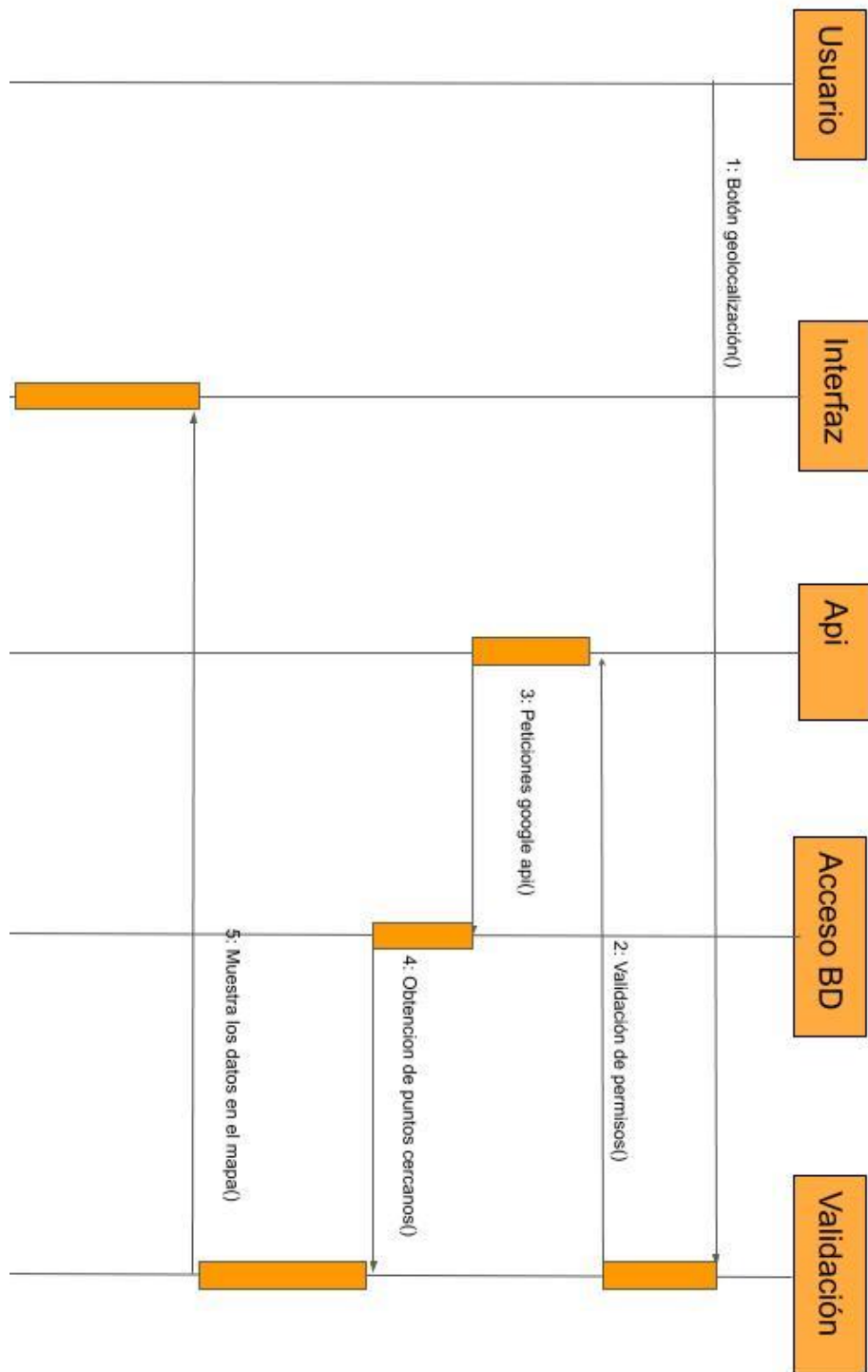
No se migró toda la información sobre los puntos kilométricos, solo aquella que se considera fundamental para el buen uso de la base de datos.

Este proceso permitió contar con una base de datos local que almacena de manera eficiente y actualizada los puntos kilométricos necesarios para el funcionamiento de la aplicación móvil.

Esta base de datos se almacena en la carpeta Assets de Android studio y utiliza la dependencia `com.readystatesoftware.sqliteasset:sqliteassethelper:2.0.1` para trabajar con esta base de datos.









Se ha utilizado los colores para el diseño de la aplicación, siendo el blanco y el negro los predominantes y el rosa el toque acentuado.

#000000  #FFFFFF  #FF4081 

En cuanto a la fuente, se ha utilizado INTER REGULAR, una fuente descargada de Google.

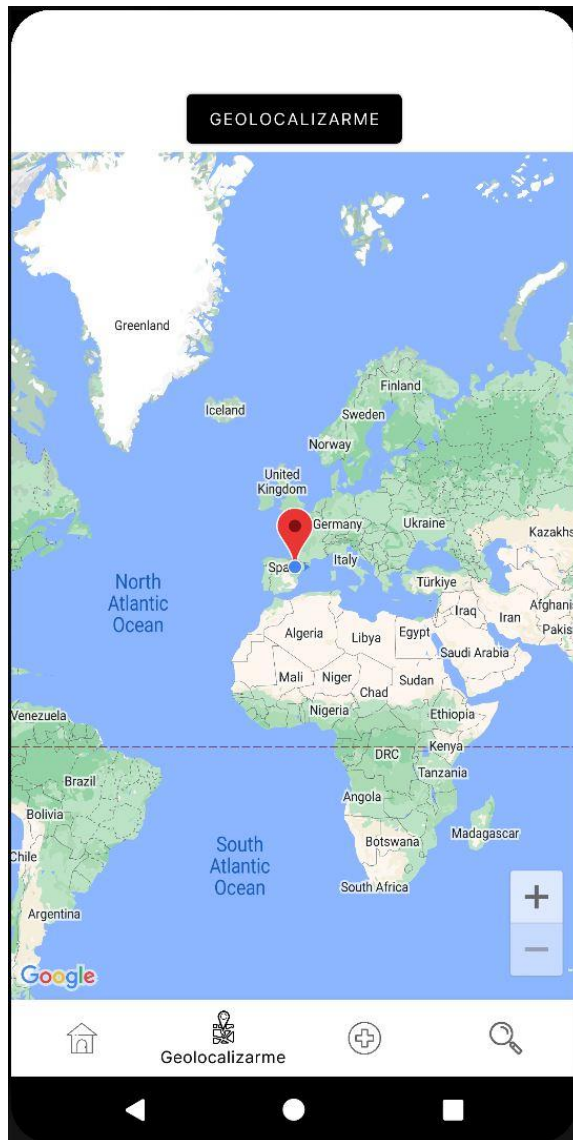


La interfaz nada mas abrir la app es esta, Contiene dos TextView uno para el titulo y otro para el párrafo de bienvenida y explicación de la app y la leyenda.





La interfaz de geolocalización es esta, contiene un Button y un mapa de Google.







La interfaz de asistencia es esta, contiene dos TextView y dos ImageButton





La interfaz para guardar y cambiar los números de los teléfonos es esta, contiene dos TextViex , dos EditText para insertar o actualizar los teléfonos y dos Buttons.

The screenshot shows a mobile application interface with a white background and rounded corners. It contains two sections for entering phone numbers. The first section is labeled 'Inserte aqui el numero del seguro :' and has a corresponding text input field below it. The second section is labeled 'Inserte aqui el numero de emergencias :' and also has a corresponding text input field below it. At the bottom of the form, there are two black buttons with white text: 'GUARDAR' and 'CANCELAR'. Below the buttons is a navigation bar with four icons: a house, a person, a plus sign in a circle, and a magnifying glass. The text 'Asistencia' is centered below the plus sign icon. The entire interface is framed by a black border, and the bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

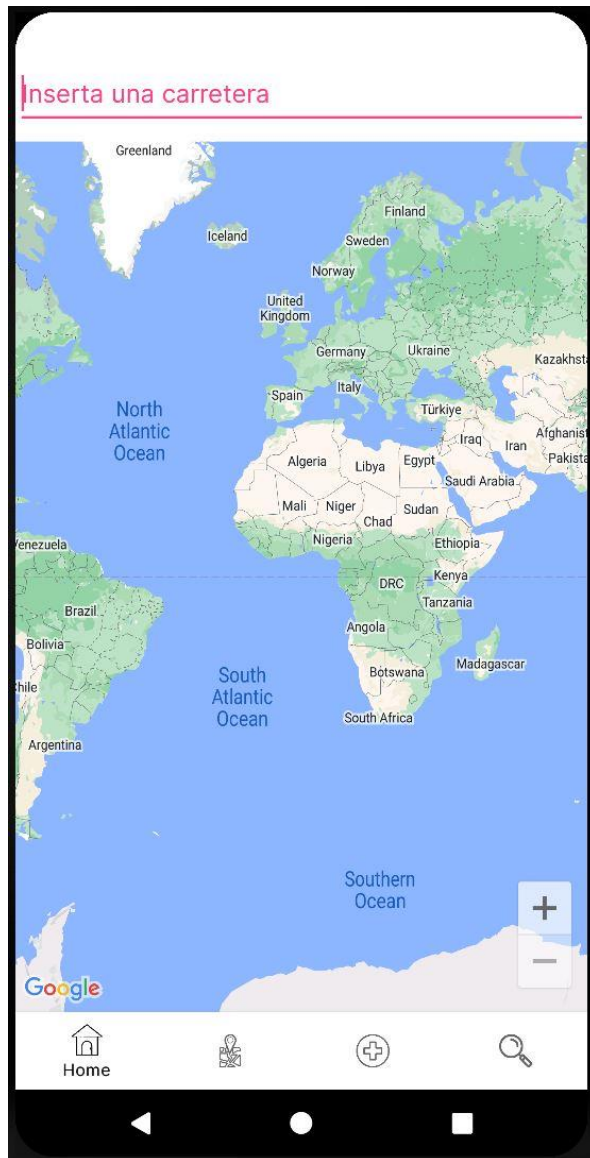


La interfaz de búsqueda es esta, contiene un TableRow con 4 ImageView:



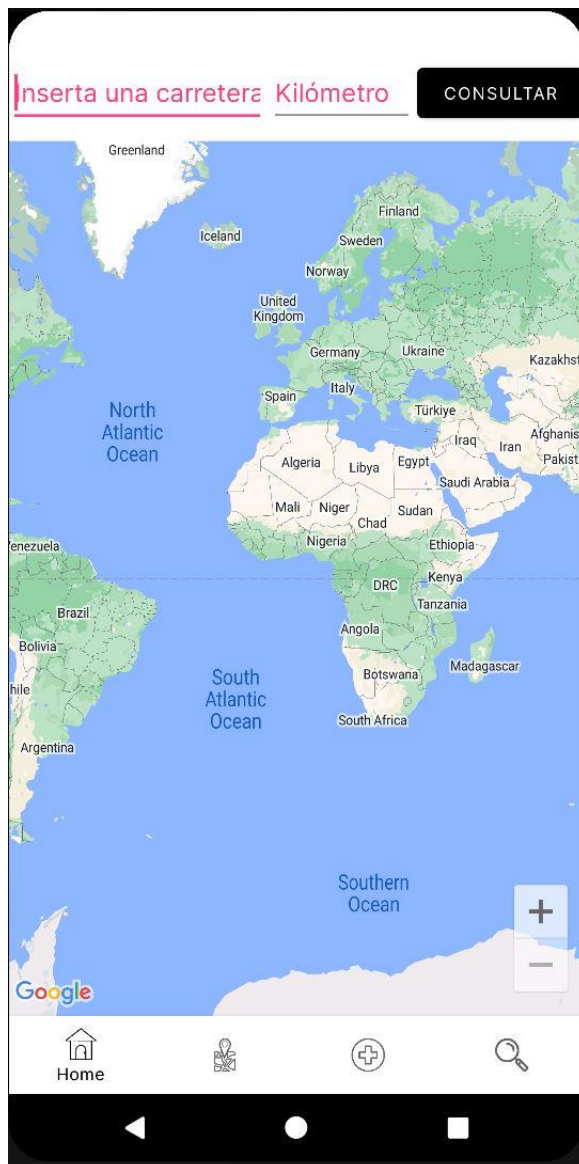


La interfaz de buscar el primer punto kilométrico y el último punto kilométrico son iguales y es esta, contiene un EditText y un mapa de Google:



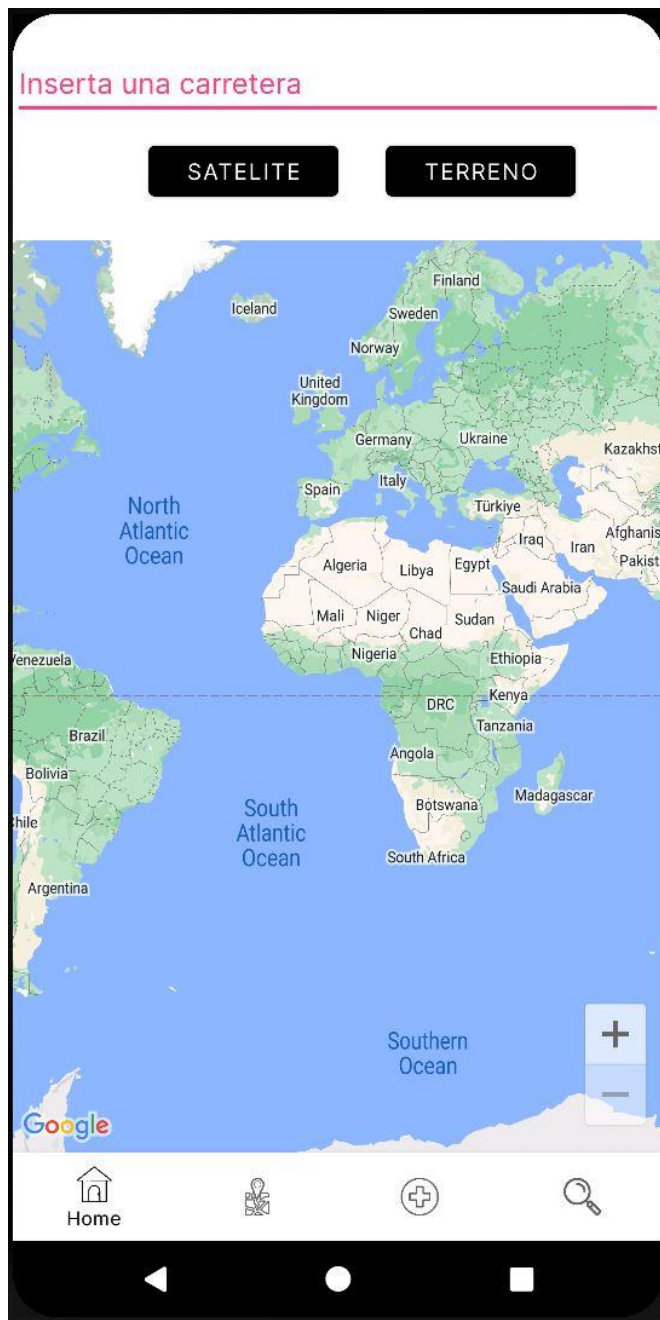


La interfaz para buscar un punto kilométrico concreto es esta, contiene dos EditText y un Button además del mapa de Google:





La interfaz para buscar todos los puntos kilométricos de una carretera es esta, contiene un EditText, un mapa de Google y dos Button para cambiar el tipo de vista del mapa:





### **Pruebas del sistema**

Las pruebas de sistema de software son una parte crucial del proceso de desarrollo, ya que permiten evaluar el funcionamiento y la integridad del sistema en su conjunto. En el caso específico de un sistema que involucra operaciones con una base de datos y procesamiento de archivos JSON, se han llevado a cabo pruebas unitarias exhaustivas para garantizar su correcto funcionamiento. Estas pruebas se han centrado en los métodos responsables de las operaciones con la base de datos, como la inserción y actualización de registros, asegurando que se realicen de manera adecuada y que los datos se manejen correctamente. Asimismo, se han realizado pruebas en los métodos encargados de procesar los archivos JSON, verificando que la lectura de datos se realiza sin errores y siguiendo las especificaciones establecidas. Estas pruebas de sistema se han hecho con test unitarios con la librería junit. Y para no tener que mockear los test, se ha añadido la dependencia `org.robolectric:robolectric:4.10.3`

Al usar robolectric permite ejecutar pruebas unitarias en el entorno de JVM en lugar de ejecutarlas en un dispositivo o emulador real. Están añadidas en el anexo.

Las pruebas de integración son una etapa esencial en el proceso de desarrollo de software, donde se verifica la interacción correcta y fluida entre los diferentes componentes del sistema. Estas pruebas se centran en probar la comunicación y la colaboración entre los módulos, subsistemas o servicios del software, asegurándose de que funcionen correctamente juntos. Se simulan escenarios de integración realistas y se prueban las interfaces y dependencias entre los componentes para identificar posibles problemas de interoperabilidad o fallos en la comunicación. El objetivo principal de las pruebas de integración es garantizar que el sistema funcione

PUNTOS KILOMETRICOS

DAM 2 Dual – Santiago Hernández



como un todo coherente y que cumpla con los requisitos establecidos, brindando confianza en su rendimiento y estabilidad.



**Documentación para el cliente****Manual de Usuario - App de Geolocalización y Asistencia Vial**

¡Bienvenido/a a nuestra aplicación de Geolocalización y Asistencia Vial! A continuación, te proporcionamos una guía rápida para aprovechar al máximo todas las características y funcionalidades que ofrece nuestra aplicación.

**Navigation Button Menu:** En la parte inferior de la pantalla, encontrarás un menú con cuatro opciones principales. Puedes acceder a cada una de ellas tocando los iconos correspondientes:

a) Home: En esta sección, encontrarás información general sobre la aplicación, así como una leyenda explicativa sobre los iconos utilizados en los mapas.

b) Geolocalización: Al seleccionar esta opción, la aplicación utilizará la tecnología de geolocalización de tu dispositivo para mostrar tu posición actual en el mapa. Además, podrás encontrar los puntos kilométricos más cercanos de la carretera en la que te encuentras. Si deseas actualizar tu ubicación, simplemente pulsa el botón "Geolocalizar" para mostrar los puntos kilométricos.

c) Asistencia: En esta sección, podrás guardar hasta dos números de teléfono de contacto para casos de emergencia o asistencia vial. También encontrarás dos botones de llamada rápida que te permitirán llamar a los números guardados con un solo toque.

d) Búsqueda de Puntos Kilométricos: Esta opción te brinda varias formas de buscar puntos kilométricos en las carreteras:

**Primer punto kilométrico:** Te muestra el primer punto kilométrico de una carretera específica.

**Último punto kilométrico:** Muestra el último punto kilométrico disponible en una carretera en particular.



Todos los puntos kilométricos: Esta función te permite visualizar todos los puntos kilométricos de una carretera específica.

Punto kilométrico concreto: Permite buscar y mostrar un punto kilométrico específico en una carretera determinada.

Recuerda que nuestra aplicación está diseñada para brindarte información precisa y herramientas útiles para tu experiencia vial. ¡Esperamos que disfrutes utilizando nuestra aplicación de Geolocalización y Asistencia Vial! Si tienes alguna pregunta o necesitas asistencia adicional, no dudes en contactarnos a través de nuestros canales de soporte.

Por favor, ten en cuenta que nuestra aplicación requiere ciertos permisos para funcionar correctamente. Recuerda que, si no otorgas los permisos necesarios, es posible que algunas funcionalidades de la aplicación no estén disponibles o no funcionen correctamente.

## **Metodología**

En el desarrollo de esta tesis, se ha seguido la metodología en cascada para llevar a cabo el proceso de investigación y desarrollo de manera estructurada y secuencial. La metodología en cascada, también conocida como el modelo de desarrollo clásico, se basa en una secuencia lineal y ordenada de fases, donde cada etapa se realiza de forma secuencial y no se avanza a la siguiente hasta que la anterior se haya completado. Esta metodología ha permitido una planificación rigurosa de las actividades, una definición clara de los objetivos y una división adecuada de las tareas a lo largo del tiempo. Además, ha facilitado la identificación temprana de posibles problemas o desviaciones, permitiendo una pronta corrección y asegurando la calidad del trabajo realizado. La metodología en cascada ha sido



fundamental para lograr una estructura lógica y coherente en el desarrollo de esta tesis.

## **Recursos**

Para llevar a cabo el desarrollo de esta aplicación, se han utilizado diversos recursos que han sido fundamentales en el proceso. Se ha empleado un ordenador HP con un procesador Intel Core i5 de 9ª generación, 500 GB de almacenamiento SSD y 8 GB de memoria RAM, brindando la potencia y capacidad necesarias para el desarrollo y pruebas del software. Como entorno de desarrollo, se ha utilizado Android Studio para la creación de la aplicación en sí, mientras que Eclipse ha sido utilizado para facilitar la migración de componentes. Para documentar y realizar pruebas de las peticiones a la API de Google, se ha empleado Postman, una herramienta que ha permitido validar y verificar el correcto funcionamiento de las interacciones con la API, se adjuntan pruebas en el anexo. Además, se ha utilizado una cuenta de Google con una clave para acceder a los servicios y funcionalidades necesarios. Por último, para la creación del esquema de la base de datos, se ha utilizado DB Browser SQLite, una herramienta que ha facilitado la definición y diseño del DDL (Lenguaje de Definición de Datos). Estos recursos han sido indispensables en el proceso de desarrollo de la aplicación, brindando las herramientas necesarias para asegurar su calidad y funcionalidad.



### **Gastos**

Se ha considerado el gasto asociado con el acceso a servicios y recursos externos, como la API de Google y los costos asociados con una cuenta de Google, necesaria para acceder a ciertas funcionalidades. El precio relativo varía en función del número de usuarios y la cantidad de peticiones, siendo gratuito los primeros 200 euros.

La electricidad consumida durante las horas de trabajo que equivalen a unos 100 euros.

Además de los gastos mencionados anteriormente, es importante considerar el costo del recurso humano involucrado en el desarrollo de la aplicación. Durante las 130 horas de desarrollo, se ha requerido el trabajo de un programador, cuyo salario relativo debe ser considerado. El salario del convenio del sector TIC del grupo E nivel 1 es de 15860 euros brutos anuales, 130 horas equivalen a un mes de trabajo a jornada completa. Y unas 40 horas anuales de salario durante los 3 próximos años en mantenimiento.

### **Ingresos**

Los ingresos generados por este proyecto provienen del cliente que nos ha contratado para el desarrollo de la aplicación. El monto específico de los ingresos depende del acuerdo y las condiciones negociadas con el cliente. Dicha negociación puede incluir diversos factores, como el alcance del proyecto, la complejidad de las funcionalidades requeridas y el tiempo estimado para completar el desarrollo. Es importante destacar que los ingresos generados por el proyecto pueden variar y se basan en el contrato específico establecido con el cliente.



Para este proyecto en concreto los ingresos son por desarrollo de la app entera con un valor de 130 horas de trabajo actual más 40 horas anuales de mantenimiento anuales los próximos 3 años hace un total 250 horas.

El precio hora que se acuerda con el cliente son 20 euros.

Esto hace unos ingresos totales de 5000 euros.

### **Beneficio**

Ingresos: 5000 euros

Gastos de electricidad: 100 euros

Gastos de salario: 1,250 euros

Beneficio neto:  $5000 - 1,250 - 100 = 3650$  euros

Después de restar los gastos de electricidad y salario, el beneficio neto resultante sería de 3650 euros.



Durante el desarrollo de la aplicación, se lograron implementar exitosamente todas las funcionalidades planificadas, incluyendo la capacidad de conocer el punto kilométrico exacto en el que se encuentra el usuario y realizar búsquedas específicas de puntos kilométricos en una carretera determinada. Esto ha permitido a los usuarios tener un mayor control y precisión al utilizar la aplicación. La geolocalización ha demostrado ser precisa y confiable, brindando información actualizada sobre la ubicación del usuario y mostrando de manera efectiva los puntos kilométricos más cercanos en el mapa. Además, la funcionalidad de asistencia ha sido bien recibida por los usuarios, al permitirles guardar contactos de emergencia y realizar llamadas rápidas desde la aplicación. Estos resultados positivos demuestran la efectividad de la implementación y validan la utilidad de la aplicación en situaciones reales. En conclusión, el desarrollo de esta aplicación ha cumplido con éxito los objetivos propuestos, brindando a los usuarios una herramienta confiable y funcional para la geolocalización y búsqueda de puntos kilométricos en carreteras.



Referencias sobre implementaciones específicas (api, geolocalización, asistencia y búsqueda de puntos kilométricos, base de datos, pedir permisos, requests):

Maplink Global. (2023). Cómo obtener Google Maps API Key. Recuperado de <https://maplink.global/blog/es/como-obtener-google-maps-api-key/>

Google. (2023). Google Maps Platform - Documentation - Roads API - Nearest Roads. Recuperado de <https://developers.google.com/maps/documentation/roads/nearest?hl=es-419>

Google. (2023). Google Maps Platform - Documentation - Places API for Android - Place IDs. Recuperado de <https://developers.google.com/maps/documentation/places/android-sdk/place-id?hl=es-419>

Develou. (2023). Google Maps Android API v2 - Tutorial Completo en Español. Recuperado de <https://www.develou.com/google-maps-android-api-v2/#insertar-mapa>

Google. (2023). Google Maps. Recuperado de <https://mvnrepository.com/artifact/com.google.android.gms/play-services-maps>

Google. (2023). Google Location. Recuperado de <https://mvnrepository.com/artifact/com.google.android.gms/play-services-location>

Karumi. (2023). Dexter. Recuperado de <https://mvnrepository.com/artifact/com.karumi/dexter>

Readystate Software. (2023). SQLiteAssetHelper. Recuperado de <https://mvnrepository.com/artifact/com.readystatesoftware.sqliteasset/sqliteassethelper>

Apache Software Foundation. (2023). Apache Commons Lang. Recuperado de <https://mvnrepository.com/artifact/org.apache.commons/commons-lang3>

<https://mvnrepository.com/artifact/com.squareup.okhttp/okhttp>

Referencias de documentación de Android Developer:

Android Developers. (2023). Buttons. Recuperado de

<https://developer.android.com/guide/topics/ui/controls/button>

Android Developers. (2023). TextViews. Recuperado de

<https://developer.android.com/reference/android/widget/TextView>

Android Developers. (2023). API Level. Recuperado de

<https://developer.android.com/guide/topics/manifest/uses-sdk-element>

Android Developers. (2023). Saving Data Using SQLite. Recuperado de

<https://developer.android.com/training/data-storage/sqlite>

Android Developers. (2023). Working with the AssetManager. Recuperado de

<https://developer.android.com/guide/topics/resources/providing-resources#Assets>

Android Developers. (2023). Making Network Requests. Recuperado de

<https://developer.android.com/training/volley>

Android Developers. (2023). HttpURLConnection. Recuperado de

<https://developer.android.com/reference/java/net/HttpURLConnection>

Android Developers. (2023). Requesting Permissions at Run Time. Recuperado de

<https://developer.android.com/training/permissions/requesting>

Android Developers. (2023). Permissions. Recuperado de

<https://developer.android.com/guide/topics/permissions/overview>





<https://developers.google.com/android/reference/com/google/android/gms/maps/MapFragment>

Android Developers. (2023). MapStyleOptions. Recuperado de

<https://developers.google.com/android/reference/com/google/android/gms/maps/Mode/MapStyleOptions>

Android Developers. (2023). TableLayout. Recuperado de

<https://developer.android.com/reference/android/widget/TableLayout>

Android Developers. (2023). ConstraintLayout. Recuperado de

<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>

Android Developers. (2023). RelativeLayout. Recuperado de

<https://developer.android.com/reference/android/widget/RelativeLayout>

Android Developers. (2023). Create a Navigation Drawer. Recuperado de

<https://developer.android.com/guide/navigation/navigation-ui>

Android Developers. (2023). Bottom Navigation. Recuperado de

<https://developer.android.com/guide/navigation/bottom-navigation>

### Referencias de Esri España:

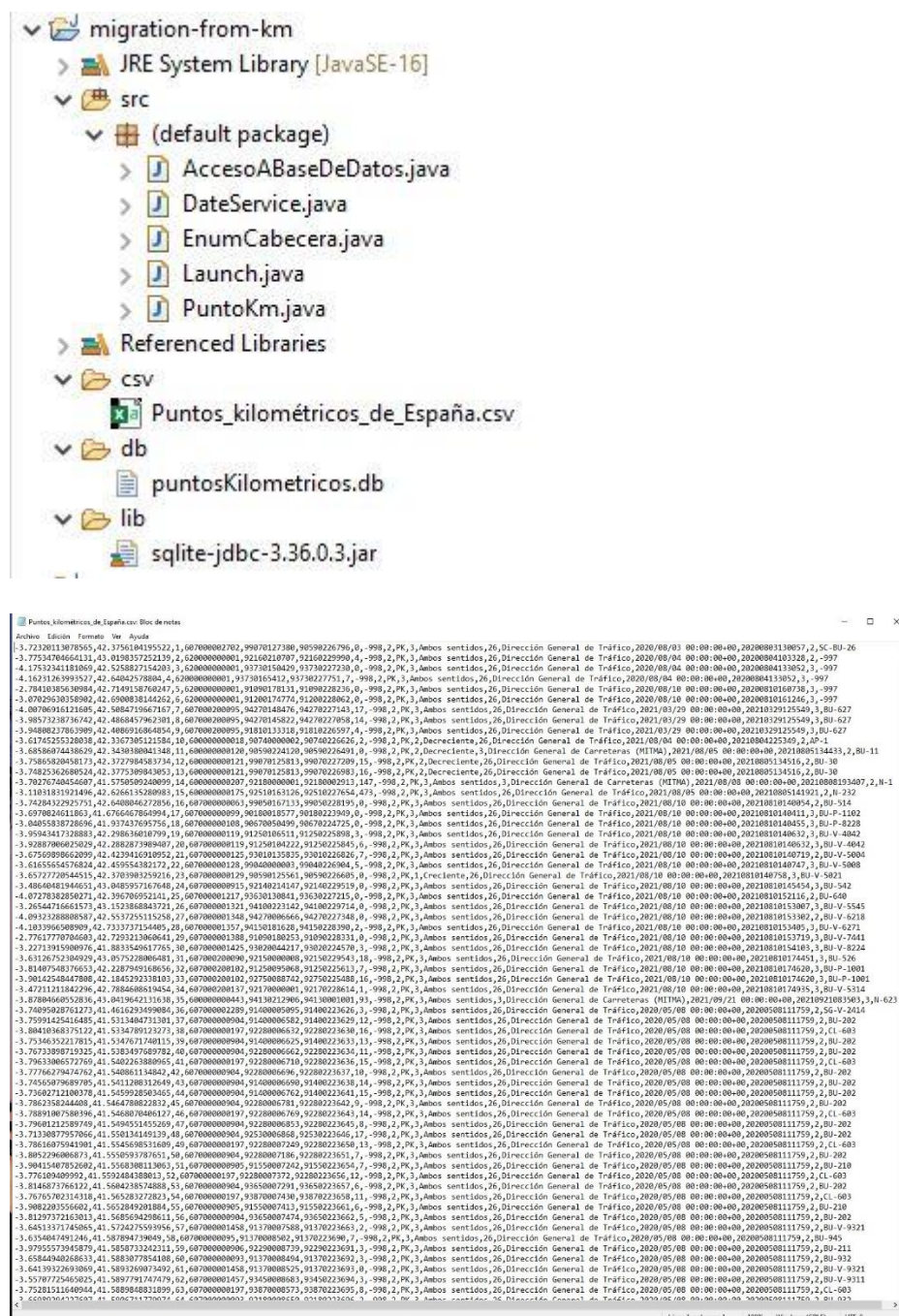
Esri España. (2023). Open Data de Esri España. Recuperado de

[https://opendata.esri.es/datasets/d8854f26fd5c4baab08337ca0f3aff6f\\_0/explore?location=53.105880%2C-17.428683%2C6.35](https://opendata.esri.es/datasets/d8854f26fd5c4baab08337ca0f3aff6f_0/explore?location=53.105880%2C-17.428683%2C6.35)



## Migración de datos

Para migrar los datos, utilice Eclipse, java 16, el fichero .db de base de datos con las tablas ya creadas y el fichero CSV con los puntos kilométricos a migrar. Además, utilice los drivers de SQLite para poder hacer operaciones de inserción con la base de datos. Aquí muestro las clases java de la aplicación:



```
1 | AccesoABaseDatos.java
2 |
3 | import java.sql.Connection;
4 |
5 |
6 |
7 |
8 |
9 |
10 |
11 |
12 | public class AccesoABaseDatos {
13 |
14 |     private final static String NOMBRE_CONTROLADOR_SQLITE = "org.sqlite.JDBC";
15 |
16 |     private final static String URL_SQLITE_DB_PERSONAL = "jdbc:sqlite:db\\puntosKilometricos.db";
17 |
18 |     private final static String QUERY_INSERTAR_PUNTOS = " INSERT INTO puntos (longitud, latitud, carretera, kilometro, sentido, fecha) VALUES(?, ?, ?, ?, ?, ?) ";
19 |
20 |
21 |     public static Connection crearConexion() throws ClassNotFoundException, SQLException {
22 |         Connection conexion = null;
23 |         Class.forName(NOMBRE_CONTROLADOR_SQLITE);
24 |         SQLiteConfig config = new SQLiteConfig();
25 |         config.enforceForeignKeys(true);
26 |         conexion = DriverManager.getConnection(URL_SQLITE_DB_PERSONAL, config.toProperties());
27 |         return conexion;
28 |     }
29 |
30 |     public static boolean insertarPuntoKm(final Connection conexion, final PuntoKm puntoKm) throws ClassNotFoundException, SQLException{
31 |
32 |         try(PreparedStatement sentencia = conexion.prepareStatement(QUERY_INSERTAR_PUNTOS)){
33 |             sentencia.setString(1, puntoKm.getLongitud());
34 |             sentencia.setString(2, puntoKm.getLatitud());
35 |             sentencia.setString(3, puntoKm.getNombre());
36 |             sentencia.setString(4, puntoKm.getNumero());
37 |             sentencia.setString(5, puntoKm.getSentido());
38 |             sentencia.setString(6, puntoKm.getFechaAlta());
39 |
40 |             return sentencia.executeUpdate() == 1;
41 |         }
42 |     }
43 | }
```

## PUNTOS KILOMETRICOS

```

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```





```
1 PuntoKm.java
2
3 public class PuntoKm {
4
5     private String longitud;
6     private String latitud;
7     private String nombre;
8     private String numero;
9     private String sentido;
10    private String fechaAlta;
11
12
13
14    public PuntoKm(String longitud, String latitud, String nombre, String numero, String sentido, String fechaAlta) {
15        this.longitud = longitud.replace(',', '.');
16        this.latitud = latitud.replace(',', '.');
17        this.nombre = nombre;
18        this.numero = numero;
19        this.sentido = sentido;
20        this.fechaAlta = DateService.convertirFecha(fechaAlta);
21    }
22
23
24    public String getLongitud() {
25        return longitud;
26    }
27
28
29    public String getLatitud() {
30        return latitud;
31    }
32
33
34    public String getNombre() {
35        return nombre;
36    }
37
38
39    public String getNumero() {
40        return numero;
41    }
42
43
44    public String getSentido() {
45        return sentido;
46    }
47
48
49    public String getFechaAlta() {
50        return fechaAlta;
51    }
52
53 }
54
```

```
1 EnumCabecera.java
2
3 public class EnumCabecera {
4
5     public final static int LONGITUD = 0;
6     public final static int LATITUD = 1;
7     public final static int KILOMETRO = 6;
8     public final static int SENTIDO = 10;
9     public final static int FECHA = 14;
10    public final static int CARRETERA = 17;
11
12 }
```



```
1 |
2
3
4 public class DateService {
5
6     public static String convertirFecha(String fechaString) {
7         if(fechaString == null || fechaString.isBlank()) {
8             return null;
9         }
10        return fechaString.substring(0, 10);
11    }
12 }
13
```



The screenshot shows the Postman interface with a GET request to the Google Maps API. The URL is `https://maps.googleapis.com/maps/api/place/details/json?place_id=CHJ-Tb37ZBWQORNM-Q2wKHM&key=AIzaSD6EB3NjW9UPTfRnH43UgpyBh_8Rv8`. The request is configured with a 'key' parameter and a 'place\_id' value. The response is shown in JSON format, displaying details for 'Autovía del Ebro'.

**Request:**

```
GET https://maps.googleapis.com/maps/api/place/details/json?place_id=CHJ-Tb37ZBWQORNM-Q2wKHM&key=AIzaSD6EB3NjW9UPTfRnH43UgpyBh_8Rv8
```

**Response:**

```
{
  "html_attributions": [],
  "result": {
    "address_components": [
      {
        "long_name": "Autovía del Ebro",
        "short_name": "A-68",
        "types": [
          "route"
        ]
      },
      {
        "long_name": "Zaragoza",
        "short_name": "Zaragoza",
        "types": [
          "locality",
          "political"
        ]
      },
      {
        "long_name": "Zaragoza",
        "short_name": "Z",
        "types": [
          "region"
        ]
      }
    ],
    "types": [
      "route"
    ]
  }
}
```



File Edit View Help

Home Workspaces Explore

Search Postman

Sign In Create Account

Scratch Pad

New Import

Working locally in Scratch Pad. Switch to a Workspace

https://roads.googleapis.com/v1/snapToRoads?path=41.733927,-1.061426&interpolate=true&key=AIzaSyD6EE3hYw9UPT4RuH43UlpSyBh\_gRY8 - My Workspace

GET get a file from xi GET http://0.0.0.0:8 POST get a file from DE http://0.0.0.0:8 DE http://0.0.0.0:8 PUT post un file from GET http://0.0.0.0:8 GET https://roads.g GET https://maps.g + No Environment Save Bulk Edit

https://roads.googleapis.com/v1/snapToRoads?path=41.733927,-1.061426&interpolate=true&key=AIzaSyD6EE3hYw9UPT4RuH43UlpSyBh\_gRY8

GET https://roads.googleapis.com/v1/snapToRoads?path=41.733927,-1.061426&interpolate=true&key=AIzaSyD6EE3hYw9UPT4RuH43UlpSyBh\_gRY8

Params Authorization Headers Body Pre-request Script Tests Settings

Query Params

|             |                                      |             |           |
|-------------|--------------------------------------|-------------|-----------|
| key         | Value                                | Description | Bulk Edit |
| path        | 41.733927,-1.061426                  |             |           |
| interpolate | true                                 |             |           |
| key         | AIzaSyD6EE3hYw9UPT4RuH43UlpSyBh_gRY8 |             |           |
| key         | Value                                | Description |           |

Send

Body Cookies Headers (1) Test Results

Raw Preview Visualize JSON

1 {  
2 "snappedPoints": [  
3 {  
4 "location": {  
5 "latitude": 41.7339272961342564,  
6 "longitude": -1.06142678955455  
7 },  
8 "originalIndex": 0,  
9 "placeId": "CH13-b372B1M0GBMv-Q2xvWM"  
10 },  
11 ]  
12 }

Status: 200 OK Time: 60 ms Size: 508 B Save Response

Find and Replace Console

Runner Trash





## Test unitarios de la clase JsonServiceTest.java

```
@RunWith(RobolectricTestRunner.class)
public class JsonServiceTest {

    /**
     * Se comprueba que obtiene un Optional con el placeId si lo contiene el json que se le pasa por parámetro
     * {@link JsonService#getPlaceId(String)}
     */
    @Test
    public void testGetPlaceId() {
        //Pasandole un json que contiene un placeId
        final String json = "{\n" +
            "    \"snappedPoints\": [\n" +
            "        {\n" +
            "            \"location\": {\n" +
            "                \"latitude\": 41.64638726094654,\n" +
            "                \"longitude\": -0.89676673331771484\n" +
            "            },\n" +
            "            \"originalIndex\": 0,\n" +
            "            \"placeId\": \"ChIJ_7JBa9AUWQ0RJBsZBVuRspM\"\n" +
            "        },\n" +
            "    ]\n" +
            "}";

        final Optional<String> placeIdOptional = JsonService.getInstance().getPlaceId(json);
        //Se comprueba que contiene un placeId
        Assert.assertTrue(placeIdOptional.isPresent());
        //Se comprueba que el placeId es el mismo que el del json
        Assert.assertEquals("Debe ser el mismo placeId", placeIdOptional.get(), "ChIJ_7JBa9AUWQ0RJBsZBVuRspM");

        //Pasandole un json que no contiene un placeId
        final String jsonSinPlaceId = "{}";
        try {
            JsonService.getInstance().getPlaceId(jsonSinPlaceId);
            Assert.fail("El metodo deberia haber lanzado una excepcion");
        } catch (Exception e) {
            Assert.assertEquals("Debe ser el mismo tipo de excepcion", e.getClass(), JsonDataException.class);
        }

        //Pasandole un json vacio
        try {
            JsonService.getInstance().getPlaceId(json: "");
            Assert.fail("El metodo deberia haber lanzado una excepcion");
        } catch (Exception e) {
            Assert.assertEquals("Debe ser el mismo tipo de excepcion", e.getClass(), JsonDataException.class);
        }

        //Pasandole un null
        try {
            JsonService.getInstance().getPlaceId(json: null);
            Assert.fail("El metodo deberia haber lanzado una excepcion");
        } catch (Exception e) {
            Assert.assertEquals("Debe ser el mismo tipo de excepcion", e.getClass(), JsonDataException.class);
        }
    }
}
```



```
/**
 * Obtiene un optional con el nombre de la carretera si lo encuentra en el json
 * {@link JsonService#getRoadName(String)}
 */
@Test
public void getRoadName() {
    //Pasandole un json que contiene un roadName
    final String json = "{\n" +
        "    \"html_attributions\": [],\n" +
        "    \"result\": {\n" +
        "        \"address_components\": [\n" +
        "            {\n" +
        "                \"long_name\": \"Autovía del Ebro\",\n" +
        "                \"short_name\": \"A-68\",\n" +
        "                \"types\": [\n" +
        "                    \"route\"\n" +
        "                ]\n" +
        "            },\n" +
        "            {\n" +
        "                \"long_name\": \"Zaragoza\",\n" +
        "                \"short_name\": \"Zaragoza\",\n" +
        "                \"types\": [\n" +
        "                    \"locality\",\n" +
        "                    \"political\"\n" +
        "                ]\n" +
        "            },\n" +
        "            {\n" +
        "                \"long_name\": \"Zaragoza\",\n" +
        "                \"short_name\": \"Z\",\n" +
        "                \"types\": [\n" +
        "                    \"administrative_area_level_2\",\n" +
        "                    \"political\"\n" +
        "                ]\n" +
        "            },\n" +
        "            {\n" +
        "                \"long_name\": \"Aragón\",\n" +
        "                \"short_name\": \"AR\",\n" +
        "                \"types\": [\n" +
        "                    \"administrative_area_level_1\",\n" +
        "                    \"political\"\n" +
        "                ]\n" +
        "            },\n" +
        "            {\n" +
        "                \"long_name\": \"Spain\",\n" +
        "                \"short_name\": \"ES\",\n" +
        "                \"types\": [\n" +
        "                    \"country\",\n" +
        "                    \"political\"\n" +
        "                ]\n" +
        "            },\n" +
        "        ]\n" +
        "    }\n" +
        "}"
```



```

        },\n" +
        "\icon\":" + "https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/geocode-71.png",\n" +
        "\icon_background_color\":" + "#79EB0",\n" +
        "\icon_mask_base_uri\":" + "https://maps.gstatic.com/mapfiles/place_api/icons/v2/generic_pinlet",\n" +
        "\name\":" + "Autovía del Ebro",\n" +
        "\place_id\":" + "ChIJ-Tb3728lWQQRhMV-02avKHM",\n" +
        "\reference\":" + "ChIJ-Tb3728lWQQRhMV-02avKHM",\n" +
        "\types\":" + [\n" +
        "\route",\n" +
        "],\n" +
        "\url\":" + "https://maps.google.com/?q=Autov%C3%ADa+del+Ebro,+50620+Zaragoza,+Spain&ftid=0xd596560eff736f9:0x7328af66307ec584",\n" +
        "\utc_offset\":" + 120,\n" +
        "\vicinity\":" + "Zaragoza",\n" +
        "},\n" +
        "\status\":" + "OK",\n" +
        "};

final Optional<String> placeIdOptional = JsonService.getInstance().getRoadName(json);
//Se comprueba que contiene un roadName
Assert.assertTrue(placeIdOptional.isPresent());
//Se comprueba que el roadName es el mismo que el del json
Assert.assertEquals( message: "Debe ser el mismo placeId", placeIdOptional.get(), actual: "A-68");

//Pasandole un json que no contiene un placeId
final String jsonSinPlaceId = "{}";
try {
    JsonService.getInstance().getRoadName(jsonSinPlaceId);
    Assert.fail("El metodo deberia haber lanzado una excepcion");
} catch (Exception e) {
    Assert.assertEquals( message: "Debe ser el mismo tipo de excepcion", e.getClass(), JsonDataException.class);
}

//Pasandole un json vacio
try {
    JsonService.getInstance().getRoadName( json: "");
    Assert.fail("El metodo deberia haber lanzado una excepcion");
} catch (Exception e) {
    Assert.assertEquals( message: "Debe ser el mismo tipo de excepcion", e.getClass(), JsonDataException.class);
}

//Pasandole un null
try {
    JsonService.getInstance().getRoadName( json: null);
    Assert.fail("El metodo deberia haber lanzado una excepcion");
} catch (Exception e) {
    Assert.assertEquals( message: "Debe ser el mismo tipo de excepcion", e.getClass(), JsonDataException.class);
}
}

```