



UNIVERSIDAD DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA DEL SOFTWARE

**LLEGANDO SANO Y SALVO A MI DESTINO
GETTING SAFE MY TRIP**

Realizado por
Daniel Llopis Laguna

Tutorizado por
José Luis Pastrana Brincones
Francisco Rus Mansilla

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE 2020

RESUMEN

Son muchas las personas y familias que tienen alguna persona mayor dependiente y/o enfermo, pero debido al ritmo de vida no pueden atenderles como les gustaría.

Partiendo de esta situación, se desea implementar una solución que ayude a las familias a asegurarnos que nuestros mayores o personas con cualquier minusvalía han llegado a su destino cuando hacen un trayecto ellos solos.

Para ello, se desea realizar una aplicación móvil en la cual le indiquemos nuestro destino y el medio de transporte (caminando, en transporte público o en automóvil) y calcule el tiempo estimado en llegar. Si transcurrido ese tiempo (con un margen de 15 minutos) no se han indicado que hemos llegado sanos y salvos, la aplicación mandará un aviso al usuario y caso de no ser respondido, contactará automáticamente por teléfono, WhatsApp o SMS con la persona o entidad configurada o con el 112 por defecto.

PALABRAS CLAVE

e-Health, Smartcities, GPS, asistencia a mayores, movilidad.

ABSTRACT

There are many people and families having a dependent elderly person and/or sick one, but, because of their rhythm of life they cannot help them as they would like.

Starting from that situation, we want to implement a solution to help families being sure our elders or impaired people get their target when they take a trip alone. In order to get this goal, we would like to make a mobile application where we set his/her target and the transport method he/she is using (walking, on public transport, car) and the tool will calculate the estimated time to arrive. When the arriving time is got (plus 15 minutes additional time) and it has not been confirmed he/she has arrived safe and sound, the application will send a notification to the user and in case of it is not responded, the tool will contact automatically by phone, WhatsApp or SMS with the person or entity configured, taking with the emergency number 112 by default.

KEYWORDS

e-Health, Smartcities, GPS, elderly people assistant, mobility.

Tabla de contenido

1. Introducción.....	13
1.1. Motivación y objetivos del TFG.....	13
1.2. Tecnologías y herramientas principales utilizadas	14
1.2.1. Tecnologías y herramientas utilizadas para el sistema de persistencia	14
1.2.2. Tecnologías y herramientas utilizadas para la aplicación	15
1.2.3. Otras tecnologías y herramientas usadas.....	16
1.3. Estructura de la memoria.....	16
2. Ingeniería del Software.....	17
2.1. Metodología eXtreme Programming	17
2.2. Ingeniería de Requisitos.....	18
2.2.1. Obtención de datos de usuario	18
2.2.2. Modificación de datos de usuario	19
2.2.3. Inicio de ruta.....	19
2.2.4. Adición de tiempo.....	19
2.2.5. Confirmación de llegada.....	19
2.2.6. Aviso al usuario al expirar el tiempo.....	20
2.2.7. Notificación de emergencia.....	20
2.2.8. Inicio de viaje anterior	20
2.3. Casos de Uso	21
2.3.1. Añadir datos de usuario.....	22
2.3.2. Editar datos de usuario.....	24
2.3.3. Inicio de ruta.....	26

2.3.4. Confirmación de llegada.....	28
2.3.5. Añadir tiempo al trayecto.....	29
2.3.6. Inicio de viaje anterior.....	30
2.4. Modelado y Diseño del Software.....	31
2.4.1. Usuario.....	31
2.4.2. Contacto Emergencia.....	31
2.4.3. Trayecto.....	32
3. Diseño de la aplicación.....	34
4. Desarrollo de la aplicación.....	36
4.1. Interfaz de usuario.....	36
4.2. Navegabilidad.....	36
4.3. Programación del modelo.....	38
4.4. Desarrollo de los controladores.....	40
4.4.1. Actividad Inicial.....	40
4.4.2. Actividad de Datos de Usuario.....	40
4.4.3. Actividad Principal.....	41
4.4.4. Actividad de Historial de Viajes.....	41
4.4.5. Actividad de Inicio de Viajes.....	42
4.4.6. Actividad de Viaje.....	43
4.5. Recursos.....	44
4.5.1. Idiomas.....	44
4.5.1.Array.....	44

4.5.1.Layout.....	45
4.5.1.Drawable.....	45
5. Pruebas.....	46
6. Conclusiones y trabajos futuros	48
Bibliografía	50
Apéndice.....	52
Manual de uso.....	52

1. Introducción

1.1. Motivación y objetivos del TFG

Son muchas las personas y familias que tienen alguna persona mayor dependiente y/o enfermo, pero debido al ritmo de vida no pueden atenderles como les gustaría.

En líneas generales la demencia se puede definir como la pérdida progresiva de las funciones cognitivas debida a daños o desordenes cerebrales. Esta alteración cognitiva provoca incapacidad para la realización de actividades de la vida diaria. Las demencias se caracterizan por sufrir el paciente un deterioro de las capacidades cognitivas. Realmente la demencia en sí no es una enfermedad sino una serie de síntomas originados por una enfermedad que, de alguna manera, afectan a la función cerebral.

Dicho esto, hablamos de demencia tipo Alzheimer como una demencia primaria o degenerativa cortical, con la aparición lenta de déficits amnésicos, práxicos, afásicos y cambios de personalidad.

La pérdida de memoria por edad afecta únicamente a la memoria, mientras que en la demencia se ven comprometidas otras funciones cerebrales, así como habilidades para desarrollar actividades normales de la vida diaria: dificultad para recordar nombres de personas y/u objetos familiares, facilidad para extraviar cosas, desorientación con pérdidas en rutas habituales, problemas para hacer tareas que antes resultaban sencillas, etc.

Partiendo de esta situación, se desea implementar una solución que ayude a las familias a asegurarnos que nuestros mayores o personas con cualquier minusvalía han llegado a su destino cuando hacen un trayecto ellos solos. Para ello, se desea realizar una aplicación móvil en la cual le indiquemos nuestro destino y el medio de transporte (caminando, en transporte

público o en automóvil) y calcule el tiempo estimado en llegar. Si transcurrido ese tiempo (con un margen de 15 minutos) no se han indicado que hemos llegado sanos y salvos, la aplicación mandará un aviso al usuario y caso de no ser respondido, contactará automáticamente por teléfono, WhatsApp o SMS con la persona o entidad configurada o con el 112 por defecto.

1.2. Tecnologías y herramientas principales utilizadas

Las tecnologías utilizadas para el proyecto a realizar han sido escogidas de entre las opciones que Android [1] ofrece para realizar aplicaciones móviles nativas y aquellas compatibles con dicho sistema. A continuación, se exponen ordenadas por función y con una descripción la tecnología usada junto con una breve explicación de su papel en la aplicación y el motivo de elección.

1.2.1. Tecnologías y herramientas utilizadas para el sistema de persistencia

- SQLite [2]: Base de datos de tipo Relacional, que está diseñada para todos los teléfonos móviles y la mayoría de ordenadores. Esta base de datos está usada principalmente para guardar la lista de viajes realizados anteriores y el tiempo que se tardó en realizar dicho viaje, de forma que el usuario pueda seleccionar uno de la lista y no tener que volver a realizar los cálculos de tiempo.

1.2.2. Tecnologías y herramientas utilizadas para la aplicación

- JAVA [3]: Lenguaje de programación orientado a objetos de propósito general. Se usará para la realización de la aplicación Android. Se usa esta tecnología respecto a otras por su robustez.
- Graddle [4]: Es el gestor de paquetes y dependencias para Android por defecto. Este gestor permite descargar automáticamente todas las dependencias de las distintas librerías usadas en la aplicación. Se escoge esta herramienta por preferencia personal y porque es la que usa Android Studio.
- Android Studio [5]: Entorno de desarrollo para aplicaciones Android. Es un entorno que se usa para desarrollar y probar las aplicaciones para Android. Se escoge este por la facilidad que aporta y los emuladores que puedes configurar para probar las aplicaciones.
- XML [6]: Lenguaje de Marcas Extensible. Es un metalenguaje que sirve para estructurar documentos grandes. En Android este lenguaje se usa para definir y crear hacer las interfaces gráficas de la aplicación.

1.2.3. Otras tecnologías y herramientas usadas

- GitHub [7]: Servicio de almacenamiento de repositorios Git.
- GitKraken [8]: Cliente visual que se usa para conectar con repositorios Git. Será usado para conectar con GitHub.
- Android Emulator [9]: Emulador de Android Studio sobre el que se realizarán las pruebas de las funciones implementadas. Se puede configurar el dispositivo móvil emulado con el sistema operativo y las dimensiones del teléfono.
- Postman [10]: Entorno de desarrollo de APIs que permite probar servicios REST y que será usada para comprobar las respuestas JSON que se obtendrán como resultado del uso de Google Maps. Se escoge esta por el entorno visual y la facilidad de uso.

1.3. Estructura de la memoria

Se van a abordar los distintos aspectos ordenándolos de más general, mostrando los diagramas y requisitos del proyecto, hasta aspectos más concretos como las librerías o implementaciones usadas en el proyecto.

Se comenzará tratando la ingeniería del software, comentando la metodología empleada, así como los requisitos y el modelo obtenido y se irá profundizando conforme se traten los aspectos del desarrollo del proyecto, tomando en cuenta consideraciones y decisiones en el desarrollo de esta memoria.

2. Ingeniería del Software

2.1. Metodología eXtreme Programming

Para el desarrollo del proyecto se usa la metodología eXtreme programming, también llamada XP, puesto que su base es la simplicidad. Gracias a esta base podemos realizar el desarrollo de forma ágil y podremos mantener el código con menor esfuerzo.

Dado que el proyecto se realiza con un solo programador, usamos una simplificación que no tendrá en cuenta la programación por parejas.

De entre los valores de esta metodología se resalta la comunicación. En nuestro proyecto implica realizar simplificaciones del código de forma que sea inteligible y la realización de las distintas pruebas del proyecto.

Otro valor fundamental es la retroalimentación o feedback. Al estar incluido el cliente en el desarrollo del proyecto se pueden llegar a soluciones de problemas de forma eficaz.

El primer paso es la obtención de los requisitos del producto a realizar y se define el backlog, es decir, la lista de tareas a realizar para el desarrollo.

ID	Tarea	Prioridad	Dependencia
A	Interfaz de usuario	1	-
B	Obtención de datos de usuario	2	A
C	Inicio de viaje	3	B
D	Iniciar viajes anteriores	4	C
E	Modificar datos de usuario	5	B

Las distintas tareas estarán divididas en implementaciones menores que al ser completadas darán por finalizado la tarea. A este backlog se le incluye posteriormente la tarea de pruebas del proyecto.

Para la validación de los requisitos mostramos la aplicación en funcionamiento al cliente mostrando la implementación realizada. Cuando es satisfactorio se da por finalizada la tarea y se pasa a la siguiente por orden de prioridad.

2.2. Ingeniería de Requisitos

Antes de realizar el desarrollo del proyecto se realizó una obtención de los requisitos fundamentales del proyecto, mediante los cuales se realizaron los distintos diagramas para detallar el software a realizar.

Los requisitos principales tienen en cuenta la simplicidad en la aplicación de forma que las personas a las cuales está destinado no tengan mayores dificultades a la hora de usarla. Por ello, la aplicación tiene un diseño minimalista y con funcionalidades claras. Los requisitos tomados para aplicación son:

2.2.1. Obtención de datos de usuario

La aplicación tomará datos del usuario al inicio de la misma y solo la primera vez que se inicie. Dichos datos serán:

- Nombre del usuario.
- Teléfono de contacto para emergencias.
- Método de contacto para dicho teléfono.

2.2.2. Modificación de datos de usuario

El usuario podrá modificar los datos previamente introducidos. Los datos previos aparecerán en los distintos campos.

2.2.3. Inicio de ruta

El usuario introducirá el destino y el medio de transporte y el sistema hará un cálculo del tiempo necesario para llevar a cabo el trayecto. El origen se obtendrá directamente de la posición GSP del móvil.

2.2.4. Adición de tiempo

Se podrá añadir tiempo adicional al viaje iniciado. El tiempo estará expresado en minutos.

2.2.5. Confirmación de llegada

Se esperará a que el usuario pulse un botón con el cual confirme la llegada a su destino. La confirmación aparecerá al pasar un tiempo determinado proporcional al tiempo del trayecto. En cuanto se confirme la llegada se guardará en la base de datos el viaje realizado y el tiempo en el cual en la base de datos.

2.2.6. Aviso al usuario al expirar el tiempo

Pasado el tiempo establecido del trayecto se mostrará una notificación al usuario y el dispositivo móvil vibrará para que el usuario tenga consciencia de la notificación. El usuario podrá entrar para cancelar el viaje o reajustar el tiempo a necesidad.

2.2.7. Notificación de emergencia

Si la alarma no ha sido contestada pasado un tiempo se realizará el contacto de emergencia al teléfono seleccionado y mediante el método escogido. Si no han sido configurados se realizará una llamada al 112 por defecto.

2.2.8. Inicio de viaje anterior

El usuario podrá seleccionar un viaje anterior de la lista de viajes completados que será inicializado automáticamente sin tener que insertar datos.

2.3. Casos de Uso

Obtenidos los requisitos de la aplicación realizamos los casos de uso y obtenemos a partir de ellos los diagramas de secuencia. En el diagrama de Casos de Uso mostrado en la siguiente ilustración podemos observar al actor Usuario y al Actor Distance Matrix API, que pertenece al API de Google y que será necesario para obtener el tiempo entre el origen y destino dado el método de viaje seleccionado.

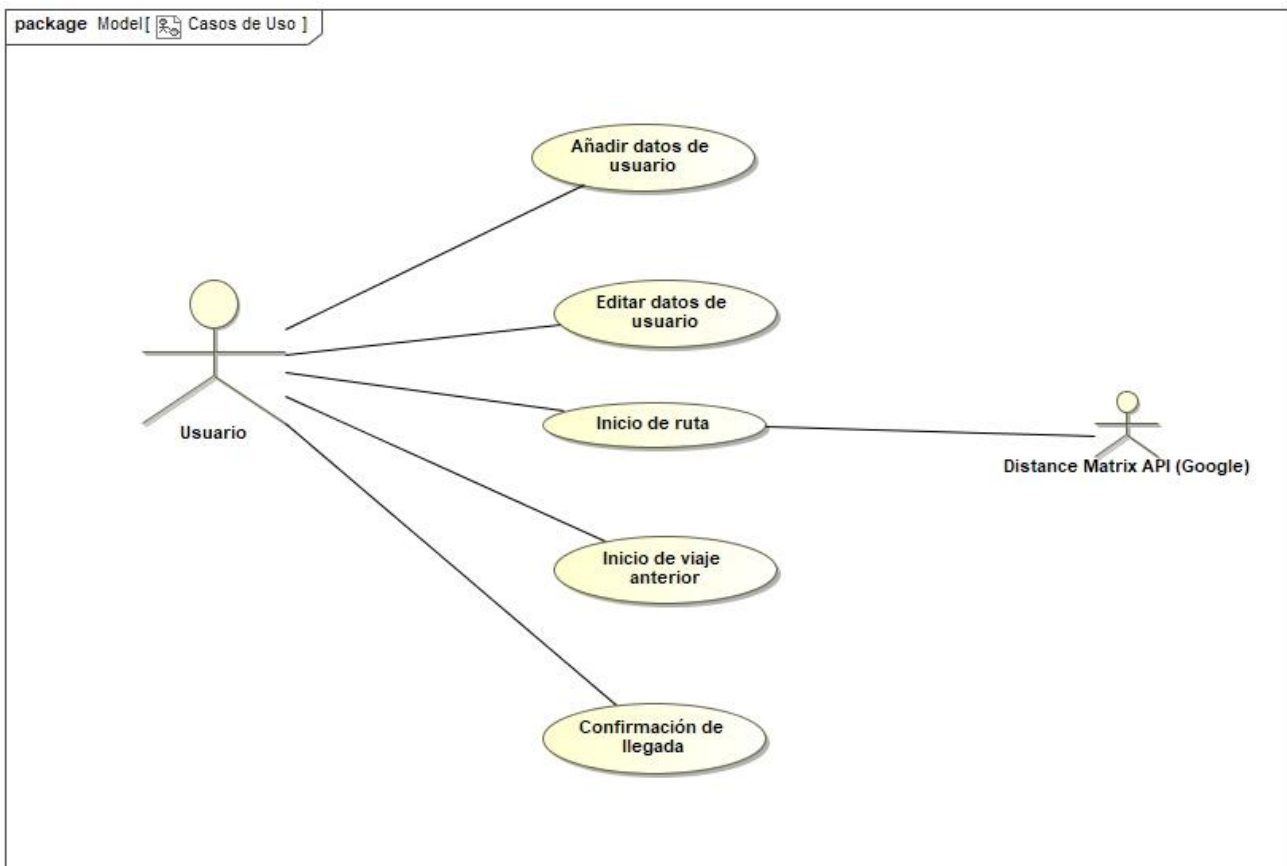


Ilustración 1: Casos de Uso

A continuación, se detallarán los casos de uso individualmente mostrando también su diagrama de secuencia asociado y una tabla que muestre las precondiciones y postcondiciones, así como el escenario principal y opcionales.

2.3.1. Añadir datos de usuario

El usuario deberá añadir sus datos. Estos datos están constituidos por el nombre, el contacto de emergencia y el método de contacto. Este caso de uso será ejecutado únicamente al iniciarse por primera vez la aplicación.

Título	Añadir datos de usuario
Descripción	El usuario podrá añadir datos a la aplicación para ser guardado.
Pre-Condición	El usuario debe haber iniciado el sistema por primera vez.
Post-Condición	Los datos insertados quedarán registrados en el sistema, salvo que se cancele la operación.
Prioridad	Imprescindible
Escenario principal	
<ol style="list-style-type: none">1. El usuario abre la aplicación por primera vez.2. El sistema pide a Android permisos del dispositivo móvil.3. Android pide al usuario los permisos.4. El usuario podrá aceptar los permisos.5. El sistema muestra el formulario de datos de usuario.6. El usuario inserta el nombre.7. El usuario pulsa el botón de guardado.8. El sistema almacena los datos9. El sistema muestra la pantalla principal.	
Escenario alternativo	
<ol style="list-style-type: none">6.a.1. El usuario pulsa el botón de contacto de emergencia.6.a.2. El sistema pide a Android la agenda de contactos.6.a.3. Android muestra la agenda a la espera de que seleccione un contacto.6.a.4. El usuario elige un contacto.6.a.5. Android envía al sistema el contacto seleccionado. 6.b.1. El usuario pulsa el desplegable de método de contacto.6.b.2. El sistema muestra las opciones.6.b.3. El usuario elige una opción.	

Interaction Añadir datos de usuario [Añadir datos de usuario]

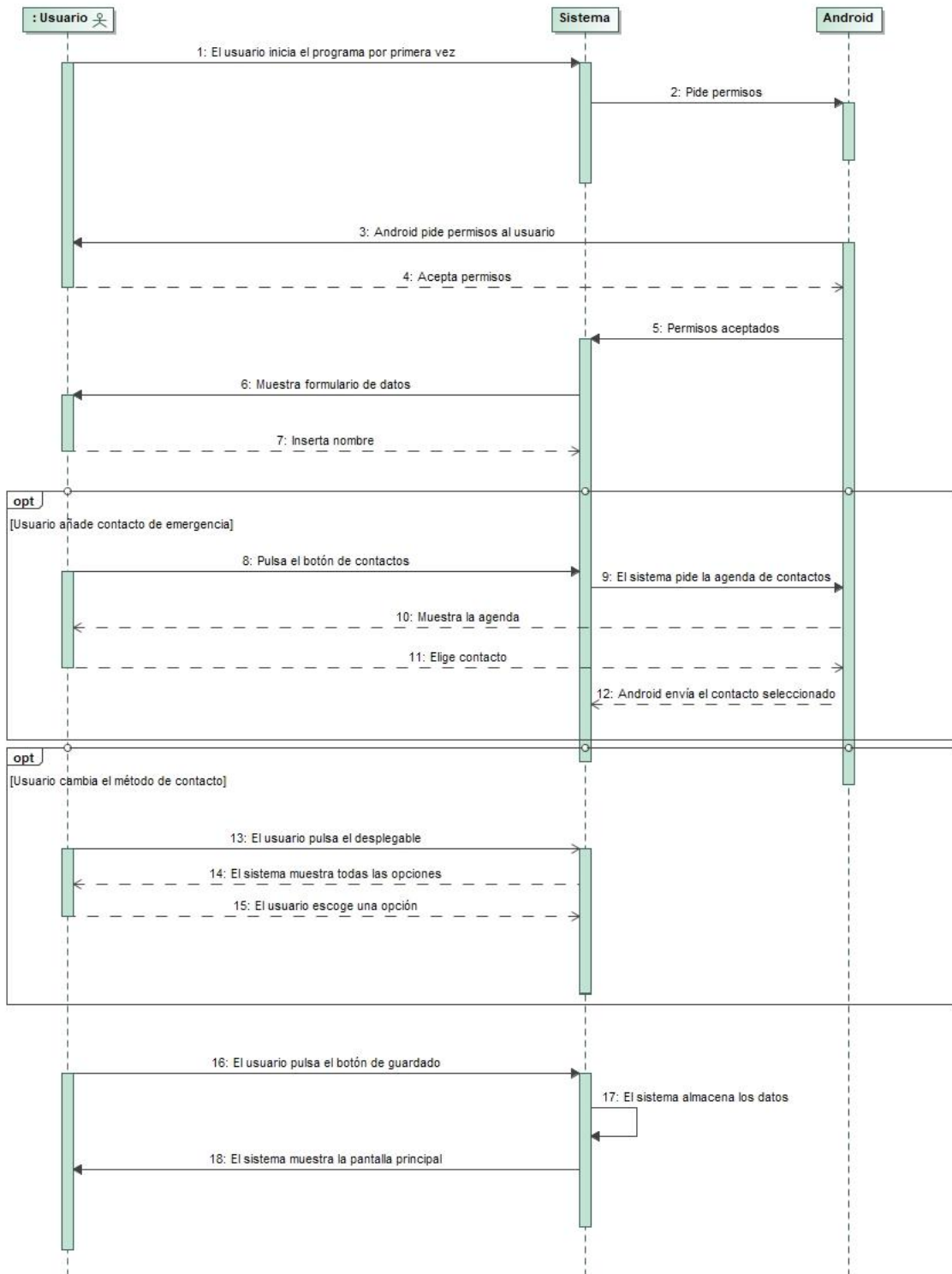


Ilustración 2: Diagrama de secuencia Añadir datos de Usuario

2.3.2. Editar datos de usuario

El usuario podrá editar en cualquier momento los datos de usuario pulsando el botón de editar en la pantalla principal. Los datos actuales aparecerán en los campos del formulario.

Título	Editar datos de usuario
Descripción	El usuario podrá editar los datos ya existentes de manera que la aplicación pueda actualizarlos.
Pre-Condición	El usuario debe tener datos.
Post-Condición	Las modificaciones se actualizarán en el sistema o se mantendrán iguales.
Prioridad	Imprescindible
Escenario principal	
<ol style="list-style-type: none">1. El usuario pulsa el botón de editar datos2. El sistema muestra el formulario de datos de usuario.3. El usuario modifica el nombre.4. El usuario pulsa el botón de guardado.5. El sistema almacena los datos6. El sistema muestra la pantalla principal.	
Escenario alternativo	
<ol style="list-style-type: none">3.a.1. El usuario pulsa el botón de contacto de emergencia.3.a.2. El sistema pide a Android la agenda de contactos.3.a.3. Android muestra la agenda a la espera de que seleccione un contacto.3.a.4. El usuario elige un contacto.3.a.5. Android envía al sistema el contacto seleccionado. 3.b.1. El usuario pulsa el desplegable de método de contacto.3.b.2. El sistema muestra las opciones.3.b.3. El usuario elige una opción.	

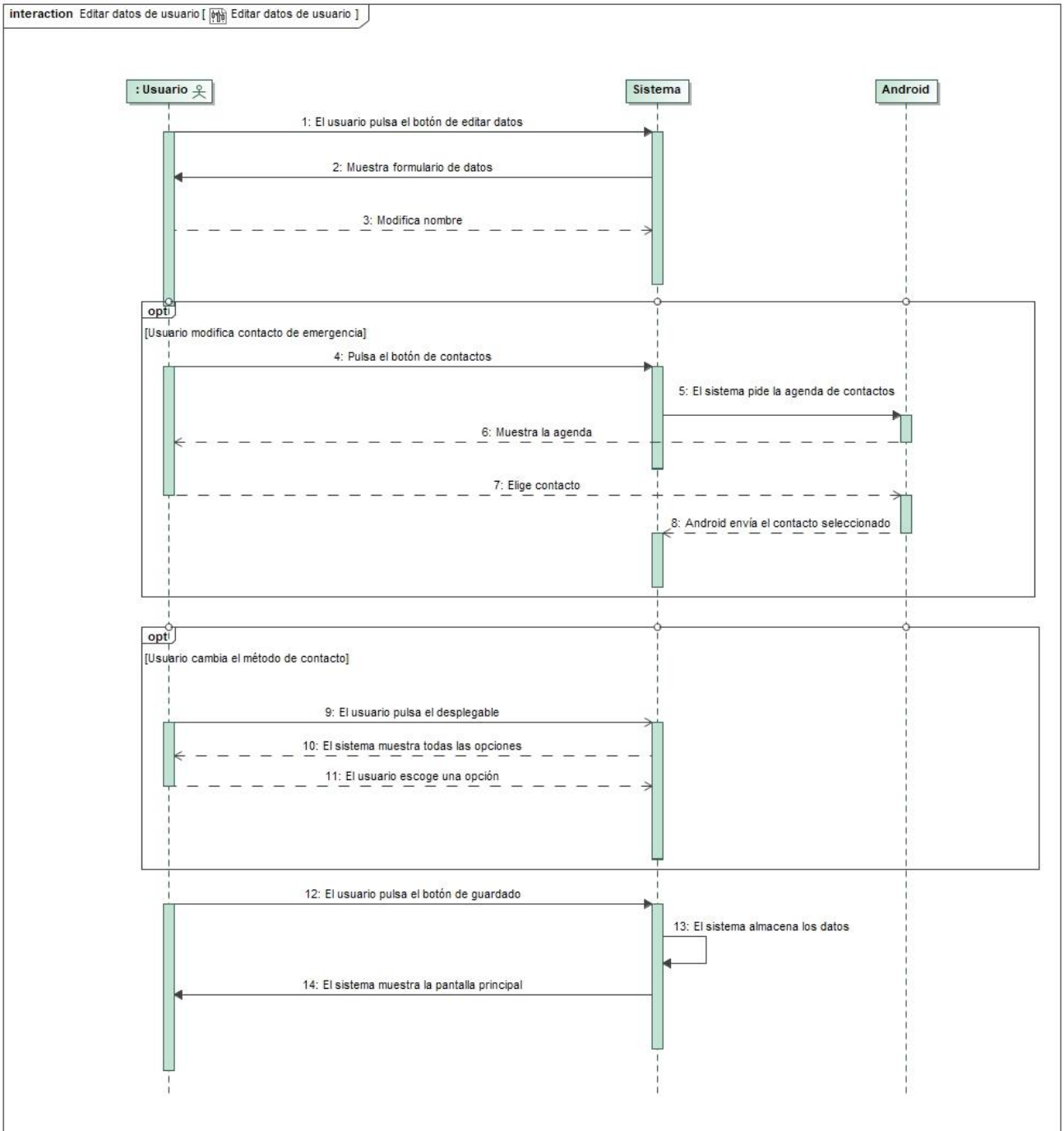


Ilustración 3: Diagrama de secuencia Editar datos de usuario

2.3.3. Inicio de ruta

El usuario pulsará el botón de inicio de ruta y podrá insertar el destino y modificar la forma de viaje. Se calculará el tiempo y una vez que el usuario confirme, el sistema iniciará el viaje con los datos obtenidos del usuario.

Título	Inicio de ruta
Descripción	El usuario introducirá el destino y podrá modificar el método de viaje.
Pre-Condición	El usuario debe haber pulsado el botón de inicio de ruta.
Post-Condición	El sistema iniciará el contador con el tiempo de ruta y mostrará la ventana de viaje iniciado.
Prioridad	Imprescindible
Escenario principal	
<ol style="list-style-type: none">1. El usuario pulsa el botón inicio de trayecto.2. El sistema muestra el formulario.3. El usuario inserta el destino.4. El sistema pide al API el tiempo con los datos introducidos.5. El API devuelve un JSON al sistema con el resultado.6. El sistema muestra el tiempo en la interfaz.7. El usuario pulsa el botón para iniciar el viaje.8. El sistema inicia y muestra la pantalla del viaje.	
Escenario alternativo	
<ol style="list-style-type: none">6.a.1. El usuario pulsa el desplegable.6.a.2. El sistema muestra las opciones del desplegable.6.a.3. El sistema pide al API el tiempo con los datos introducidos.6.a.4. El usuario elige un contacto.6.a.5. El API devuelve un JSON al sistema con el resultado.6.a.6. El sistema actualiza el tiempo.	

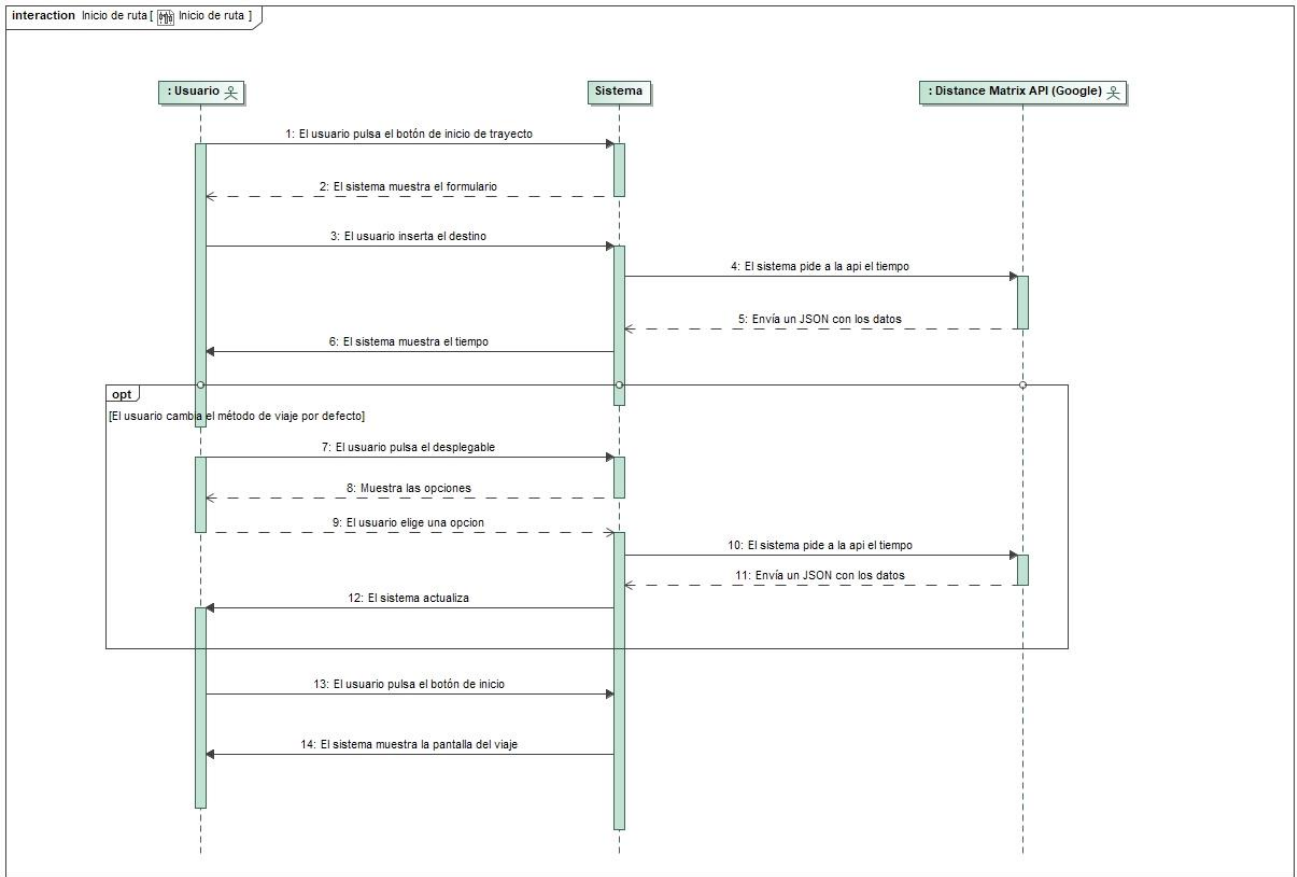


Ilustración 4: Diagrama secuencia Inicio de ruta

2.3.4. Confirmación de llegada

El usuario podrá confirmar su llegada al destino mediante un botón. El sistema guardará el viaje completado en la base de datos de los viajes anteriores.

Título	Confirmación de llegada
Descripción	El usuario podrá confirmar su llegada al destino mediante un botón. El sistema añadirá ese viaje a la base de datos.
Pre-Condición	El usuario debe estar en la ventana del viaje iniciado.
Post-Condición	El sistema guardará el viaje a la base de datos y mostrará la pantalla principal.
Prioridad	Imprescindible
Escenario principal	<ol style="list-style-type: none">1. El usuario pulsa el botón de llegada.2. El sistema guarda la información del viaje en la base de datos.3. El sistema devuelve al usuario a la pantalla principal.

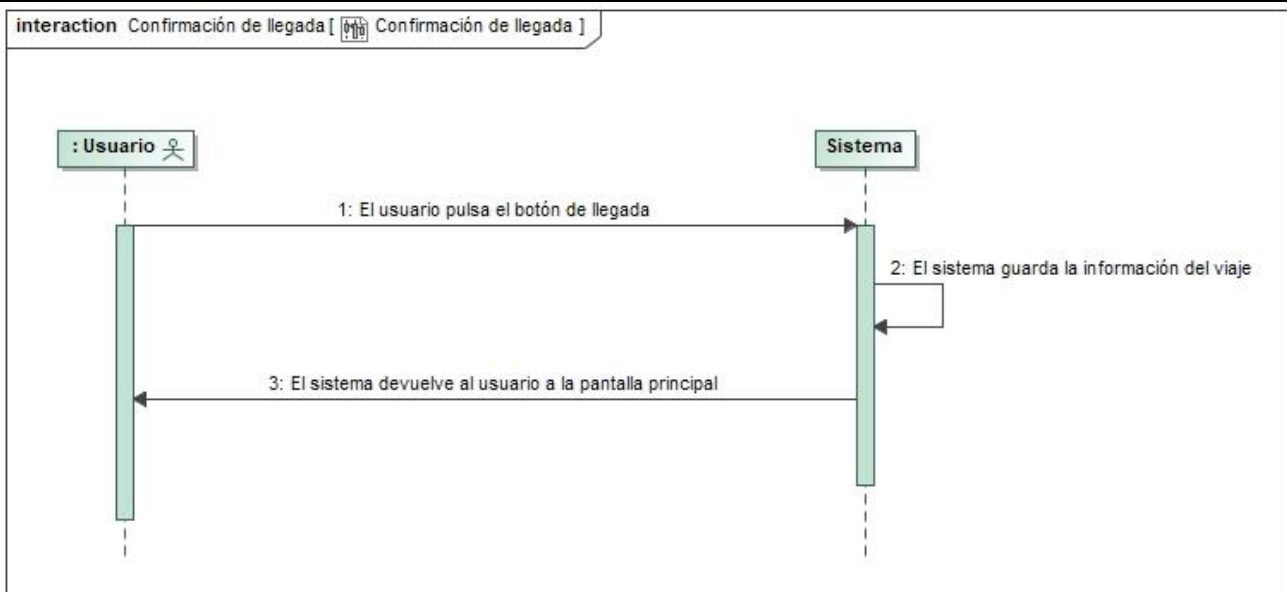


Ilustración 5: Diagrama de secuencia Confirmación de llegada

2.3.5. Añadir tiempo al trayecto

El usuario podrá añadir tiempo extra, representado en minutos, al viaje iniciado. El sistema mostrará por pantalla el contador con el tiempo actualizado.

Título	Añadir tiempo al trayecto
Descripción	El usuario podrá seleccionar el tiempo (máximo 10 minutos) que añadirá y el sistema actualizará el tiempo de viaje.
Pre-Condición	El usuario debe estar en la ventana del viaje iniciado.
Post-Condición	El sistema actualizará el tiempo que se muestra en el contador de la ventana del viaje.
Prioridad	Imprescindible
Escenario principal	
	<ol style="list-style-type: none">1. El usuario selecciona el tiempo que añadirá.2. El usuario pulsa el botón de añadir minutos.3. El sistema actualiza el tiempo del contador.

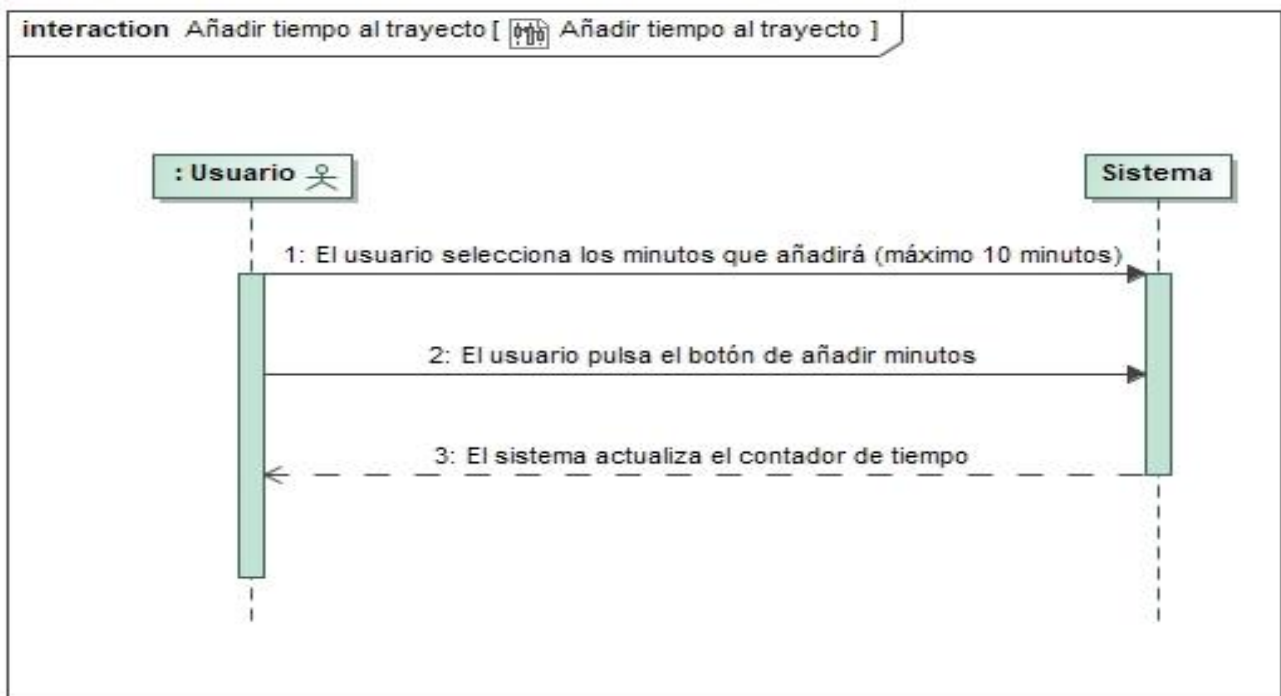


Ilustración 6: Diagrama de secuencia Añadir tiempo al trayecto

2.3.6. Inicio de viaje anterior

El usuario podrá seleccionar un viaje anterior de la lista de viajes realizados. El sistema iniciará el viaje sin necesidad de insertar datos.

Título	Inicio de viaje anterior
Descripción	El usuario podrá seleccionar un viaje que aparece en la lista de viajes completados y el sistema lo iniciará.
Pre-Condición	El usuario debe tener viajes en la base de datos para poder ser seleccionados.
Post-Condición	El sistema iniciará el viaje con los datos del viaje de la base de datos.
Prioridad	Imprescindible
Escenario principal	<ol style="list-style-type: none">1. El usuario pulsa el botón de viajes anteriores.2. El sistema consulta la base de datos y obtiene la lista de viajes.3. El sistema muestra la lista de viajes anteriores4. El usuario pulsa un viaje de la lista.5. El sistema inicia el viaje y muestra la pantalla del viaje.

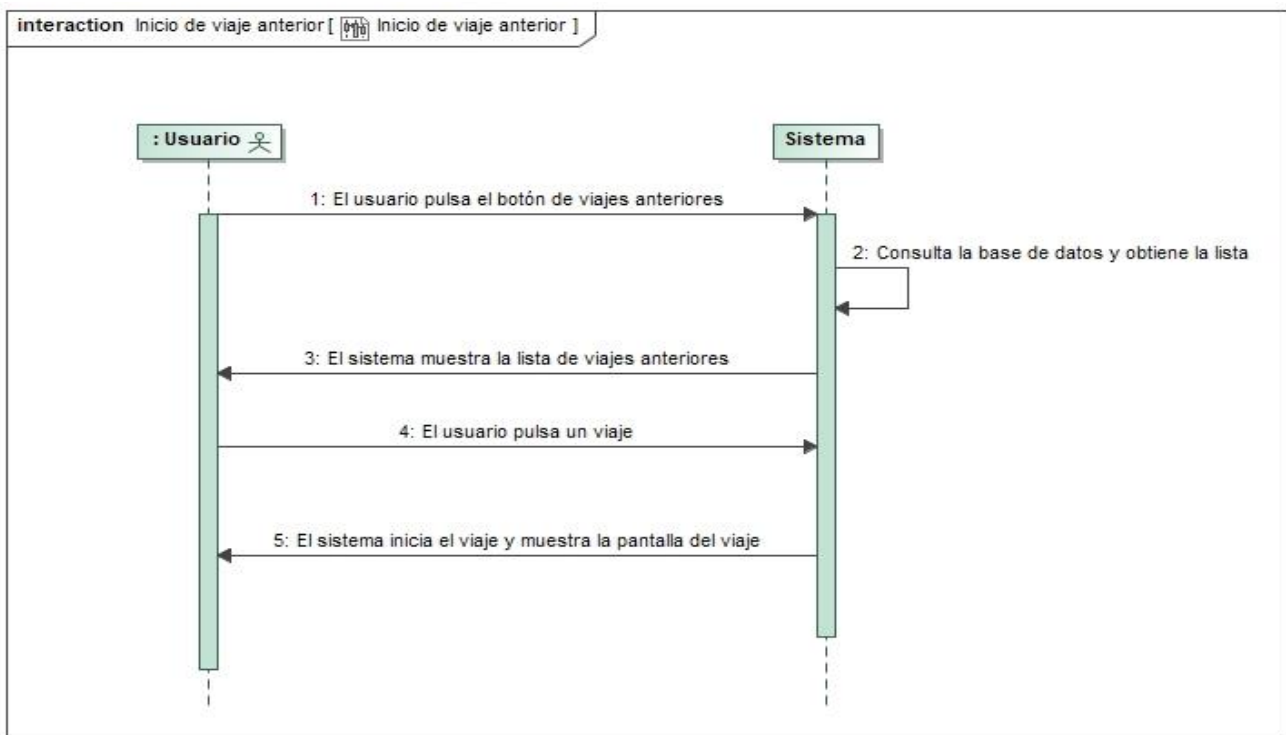


Ilustración 7: Diagrama de secuencia Inicio de viaje anterior

2.4. Modelado y Diseño del Software

Una vez obtenidos los requisitos principales y los casos de uso de la aplicación principales, procedemos a realizar al diagrama de clases en el cual representamos las distintas conexiones entre las clases.

2.4.1. Usuario

Esta clase se ocupa de guardar los datos del usuario. Aunque esta información sea simple es necesaria para poder realizar las tareas básicas del contacto de emergencia de tipo texto.

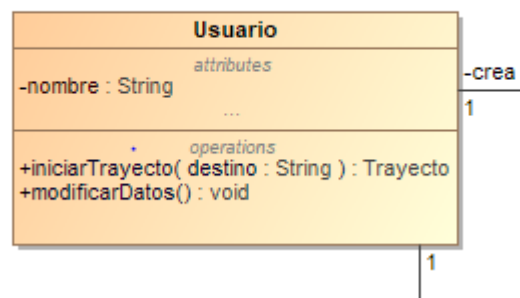


Ilustración 8: Usuario

Atributos

- **Nombre:** Este atributo será necesario para los avisos de emergencia de la aplicación por vía escrita.

2.4.2. Contacto Emergencia

El contacto de emergencia será usado para cuando sea necesario mandar el aviso. En dicho contacto guardaremos principalmente el número de contacto. Por defecto el número que aparecerá será el 112, puesto que es el teléfono de emergencia de España.

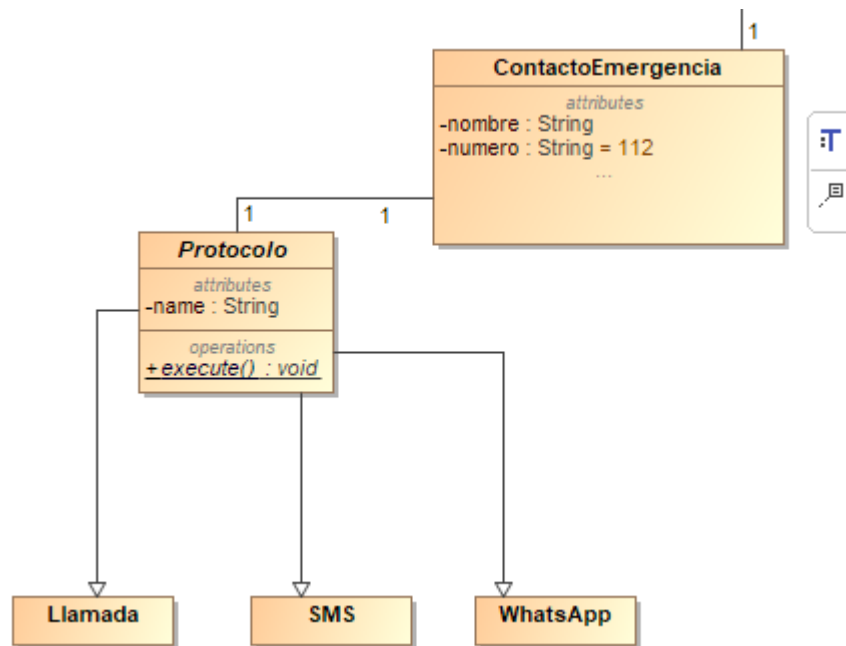


Ilustración 9: Contacto Emergencia

2.4.3. Trayecto

Esta clase será la encargada de obtener los datos de los trayectos realizados del usuario.

Dichos trayectos tomarán como origen la posición actual del usuario del GPS.

Los trayectos son almacenados en una base de datos para tener así un fácil acceso a viajes realizados anteriormente sin tener que insertar datos ni calcular el tiempo.

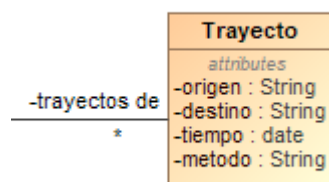


Ilustración 10: Trayecto

Atributos

- **Origen:** El origen desde el que se realiza el trayecto. Este origen es obtenido vía GPS del móvil del usuario y no podrá ser modificado.

- **Destino:** El destino será un dato a introducir cuando se realice el trayecto con este dato se realizará el cálculo para el siguiente atributo
- **Tiempo:** El tiempo se tomará basado en el cálculo de la distancia entre el origen y el destino y el método de viaje seleccionado.
- **Método:** Este atributo se obtendrá de entre una serie de opciones que permitirán seleccionar la forma de viaje (a pie, en coche, transporte público).

La relación de esta clase es la de trayectos del usuario. El usuario siempre tendrá una lista de trayectos realizados anteriormente de forma que sea sencillo volver a realizar uno. Debido a que un trayecto es almacenado en la base de datos una vez se ha confirmado la llegada a su destino, no serán necesarios los cálculos de tiempo ni obtener la posición inicial del GPS.

3. Diseño de la aplicación

Tomando en cuenta el público objetivo se ha realizado una fase de diseño de aplicación de forma que las interfaces de usuario sean lo más limpias y claras posibles reduciendo de esta forma la posibilidad de errores y/o de dudas.

Las interfaces de usuario son las siguientes:

- **Actividad de inicio:** Esta interfaz tendrá el propósito de mostrar brevemente una pantalla de carga en la cual se pedirán los permisos del móvil necesarios al usuario en caso de no haber sido usada anteriormente y comprobará si existen datos de usuario. En caso de no existir dichos datos se redirigirá a la actividad de datos de usuario. En caso contrario, se enviará directamente a la actividad principal.
- **Actividad Principal:** La actividad principal sirve de puente entre las distintas actividades de la aplicación. Desde ella podemos acceder a la actividad de inicio de trayecto, a la de viajes anteriores y la de datos de usuario. Aparecerá, además, un texto de bienvenida con el nombre del usuario.
- **Actividad de Datos de Usuario:** Esta actividad estará encargada de recopilar los datos del usuario necesarios. Estos datos son:
 - **Nombre:** El nombre del usuario. Aparecerá en blanco si no existen datos previos del usuario.
 - **Teléfono de contacto:** El teléfono al cual se contactará en caso de emergencia. Por defecto estará seleccionado el 112.
 - **Método de contacto:** La forma en la cual se contactará con el teléfono de emergencia. Por defecto estará seleccionado la llamada.

- Actividad de Viajes Anteriores: Esta actividad mostrará una lista con los viajes guardados en la base de datos. Solo estarán aquellos viajes que tuvieron una confirmación de llegada al destino. Los viajes dentro de la lista podrán ser pulsados para acceder directamente a la actividad del Viaje con los datos ya introducidos para realizar el viaje en el tiempo que se realizó la última vez.
- Actividad de Inicio de Trayecto: Esta interfaz obtendrá el destino y el método de viaje y mostrará el tiempo una vez introducido ambos datos. El origen será obtenido mediante la conexión GPS tomando la localización del dispositivo móvil.
- Actividad de Viaje: Esta actividad se accederá cuando los datos del inicio del trayecto sean introducidos y se pulse el botón de inicio. Existirá un contador de tiempo que indicará el tiempo restante del trayecto. Se podrán añadir minutos mediante el botón y la rueda de números incluidos en la actividad. En caso de que cualquiera de los botones sea pulsado, se detendrá el contador de emergencia en caso de que estuviera activado. Se podrá cancelar el viaje o indicar que se llegó al destino.

4. Desarrollo de la aplicación

La aplicación se desarrolló por tareas, separando funcionalidades. Para ello, se utilizó el lenguaje de programación JAVA que es uno de las dos opciones de Android.

Para el desarrollo de la aplicación se escogió como SDK mínima la versión 23 puesto que posee una compatibilidad elevada de dispositivos Android en los cuales la aplicación pueda ser ejecutada.

Se abarcará el desarrollo en las distintas fases realizadas.

4.1. Interfaz de usuario

El primer desarrollo fue implementar las distintas interfaces previamente diseñadas y realizar los ajustes necesarios para Android. Para las implementaciones fue utilizado XML puesto que es el lenguaje empleado para las interfaces de Android, además de ser empleado en distintos recursos. En cada interfaz se usan recursos de Android para tener un idioma en los textos empleados distinto en función del idioma al cual esté el dispositivo móvil configurado.

Además de tener los distintos XML para las distintas actividades se crearon otros recursos visuales para complementar las interfaces, como fondos degradados y estilos de botones.

Se usó una librería de Android llamada CardView para la creación de determinados elementos visuales como, por ejemplo, los botones.

4.2. Navegabilidad

Una vez realizadas las interfaces de usuario se realizó la navegabilidad básica entre las interfaces básicas de la aplicación. Se comenzó con añadir “listeners” a los distintos botones en las interfaces para navegar de una interfaz a otra.

La pantalla de inicio no cuenta con estos botones puesto que es solo una interfaz de carga. En dicha interfaz se buscan datos almacenados anteriormente en el dispositivo del usuario y en caso de encontrarlos o no se redirige a una interfaz u otra. En concreto, al iniciarse por primera vez la aplicación se piden determinados permisos para el correcto funcionamiento de la aplicación, además de redirigir a la interfaz de datos de usuario para poder tener constancia del usuario que usará la aplicación.

En la interfaz de usuario como mínimo solo se debe insertar el nombre puesto que el teléfono de emergencia es el 112 por defecto y la forma de contacto será la llamada. Estos datos pueden ser cambiados en cualquier momento entrando a la interfaz de datos desde la pantalla principal. Al pulsar el botón de selección de contacto de emergencia se redirigirá al usuario a la libreta de contactos del móvil para que seleccione uno de los que tiene guardados. De esta forma, evitamos que el usuario tenga que escribir a mano el contacto.

La selección del método será implementada mediante un “dropdown” que mostrará una lista de opciones cuando sea pulsado, teniendo el usuario que seleccionar una.

Finalmente, para guardar los datos el usuario tendrá que pulsar el botón de guardado abajo y será todo actualizado.

La interfaz principal cuenta únicamente con botones que permiten acceder a tres de las distintas interfaces principales. Será también la pantalla a la que se acceda directamente si existen datos cuando el usuario se encuentre en la pantalla de inicio. El primer botón será usado para acceder al inicio de viajes nuevos. El segundo será para acceder a la lista de viajes anteriores que fueron completados. Por último, el botón de acceso a los datos de usuario que será para acceder a la interfaz anteriormente comentada.

Al pulsar el botón de inicio de viajes se accede a la interfaz de inicio de viaje que posee dos botones para acceder a dos interfaces. El primero de ellos solo redirigirá cuando el campo

destino esté relleno. El segundo botón nos devuelve a la interfaz principal puesto que es el botón de cancelar.

En el segundo botón de la interfaz principal se accede a la lista de viajes anteriores. En esta lista aparecerán los viajes en los cuales el usuario declaró que llegó a su destino. Se podrán pulsar para ir a la ventana del viaje e iniciarlo directamente con los datos proporcionados. El otro botón solo es para volver atrás.

Volviendo a la interfaz de inicio de viajes cuando pulsemos el botón de inicio de viaje accederemos a la ventana del viaje iniciado. En ella tendremos tres botones de los cuales solo dos de ellos nos redirigen a otras interfaces. En el primer botón no se redirigirá a ninguna interfaz, pero será comentado cuando se comente dicha interfaz puesto que posee gran importancia. El segundo botón sirve para notificar que se ha llegado al destino y la aplicación guardará los datos del trayecto en la base de datos además de llevarnos a la interfaz principal. El último botón cumple el mismo objetivo, pero sin guardar el trayecto puesto que es el de cancelar.

4.3. Programación del modelo

Gracias a que Android permite guardar determinados datos en la memoria del teléfono móvil las clases del Usuario y del Contacto de emergencia pueden ser guardadas en los archivos de almacenamiento de la aplicación. Al ser una cantidad de datos baja y simple podemos ahorrarnos el tener que crear las clases. Para ello utilizamos una clase llamada Preferencias. En ella, inicializamos las SharedPreferences de Android y accedemos a los datos del archivo guardado en memoria.

Sin embargo, para la base de datos y la lista de viajes anteriores se necesita crear una clase para los trayectos para que la base de datos pueda guardar los datos de los objetos y

obtenerlos. La clase Trayecto tiene “getters” y “setters” y un constructor para la clase de forma que sea fácilmente manipulable cuando los datos se cambien, por ejemplo, si se añaden minutos extras al trayecto que se está realizado.

Para la base de datos tenemos cuatro columnas:

- ID: El ID servirá para guardar con un identificador propio cada viaje.
- Origen
- Destino
- Tiempo: El tiempo es el tiempo exacto que fue usado para llegar al destino desde el origen contando el tiempo que se añade extra.

Los métodos que posee la clase son para poder añadir y eliminar un trayecto y de obtener la lista completa de viajes de la base de datos. No es necesario buscar un trayecto en concreto debido a que todos serán mostrados en la interfaz de viajes anteriores. Para añadir los viajes se usa la estructura de ContentValues para añadir los valores y se obtiene una base de datos en modo de escritura de modo que se puedan insertar en la base de datos.

En la eliminación de un viaje se envía el objeto trayecto que quiere ser eliminado de la base de datos. Se obtiene el ID del trayecto y se envía a la base de datos que lo buscará mediante una cláusula de WHERE y lo eliminará.

En la obtención de los viajes usamos un Cursor de Android que permita recorrer toda la base de datos y los devuelva en una lista de objetos Trayecto. Esa lista será usada rellenar la interfaz de viajes anteriores. Para ello, usamos una instancia de la base de datos en modo de lectura.

4.4. Desarrollo de los controladores

Para explicar los controladores se seguirá un orden por actividad, de manera que en cada sección se hablará de la actividad en cuestión y que se usa para cumplir el objetivo de esa actividad profundizando en los aspectos de la programación y con un lenguaje técnico mayor.

4.4.1. Actividad Inicial

En la creación de la actividad se inicializa las preferencias que comentamos antes. Después se piden los permisos de Android necesarios al usuario. No se pedirán los permisos las siguientes veces que se entre en la aplicación si no han sido aceptados.

Posterior a los permisos se hace la comprobación de los datos almacenados en memoria. Si no existen se procederá a la interfaz de datos de usuario. En caso contrario, se procede directamente a la actividad principal.

4.4.2. Actividad de Datos de Usuario

La actividad de datos de usuario debe mostrar en su formulario los datos que el usuario haya introducido anteriormente y solo en el caso de que no existan se mostrará en blanco el nombre. En la creación de la actividad se hace una llamada a las preferencias y se inicializan para al archivo de almacenamiento.

Se rellenan los datos y cuando se pulse el botón de guardado creará la estructura de almacenamiento interna o la modificará. En el formulario existen tres campos en los cuales solo uno, el nombre, es relleno por texto. El segundo campo es el teléfono de contacto, el cual es un botón que te redirige a la libreta de contactos del móvil. En la libreta se debe pulsar el contacto que se desee usar y se redirigirá de nuevo a la actividad de datos con el contacto ya elegido. En ese momento se guardará temporalmente el contacto elegido. Por

último, se deberá selección el método de contacto. Para ello, se utiliza un “dropdown” que mostrará en una pequeña lista las opciones que pueden ser seleccionadas.

Al momento de pulsar el botón de guardado se obtendrán los datos que se habían guardado temporalmente y entonces serán almacenados. Se redirigirá entonces a la actividad principal.

4.4.3. Actividad Principal

Esta actividad es un puente entre las demás actividades y por ello posee tres botones. Además de eso, existe un texto de bienvenida que tendrá el nombre del usuario que se añadió al inicio de la aplicación. De la aplicación principal solo se puede salir a través de uno de estos botones o saliendo de la aplicación, el cual si es pulsando el botón “atrás” de Android entonces se deberá pulsar dos veces para evitar que sea por error. En cuanto a los botones, los “listeners” de cada uno de ellos son para redireccionar a las actividades.

4.4.4. Actividad de Historial de Viajes

En esta actividad existe una lista de viajes completados con anterioridad. Está implementado con un “listview” que tomará los datos de una lista de trayectos tomados de la base de datos. Para ello se crea un adaptador a medida que pueda obtener de la lista los datos del trayecto mediante los “getters” y entonces se usan los “setters” de los objetos de la vista para poder insertar los valores del objeto Trayecto. En la creación de la actividad se crea una instancia de la base de datos y se hace una llamada al método que devuelve la lista de trayectos. Esta lista será enviada como parámetro al constructor del adaptador. En la vista usamos un “CardView” que es una estructura para crear interfaces parecidos a cartas de la misma manera que con los botones, puesto que cada objeto de la lista podrá ser pulsado para iniciar un viaje con los datos anteriores llevándonos a la actividad del Viaje. Para ello, al pulsar el viaje se obtendrán los datos del mismo del adaptador de la lista y serán enviados

a través de un “Bundle” de Android como datos extras a la aplicación en los que obtendrá la información e iniciará el contador del viaje.

Tanto el botón de cancelar como al pulsar un viaje anterior cerramos el acceso a la base de datos.

4.4.5. Actividad de Inicio de Viajes

Es una de las actividades más importantes puesto que en esta se hace una llamada “GET” a un recurso de Google API y se obtiene la posición del dispositivo móvil. En la creación de la interfaz usamos un método llamado “getDeviceLocation()” en el cual obtenemos un cliente de localización GPS para obtener la latitud y longitud del teléfono. Como nosotros trabajamos con los nombres de las calles y localizaciones usamos Geocoder, que nos permite obtener dadas unas coordenadas información de la localización. De esta forma, obtenemos el nombre de la calle en la cual iniciamos el viaje y será guardado como origen.

El campo de texto del destino tiene implementado un “TextWatcher” que nos permitirá saber cuándo el campo de texto es modificado y así realizar un cálculo del tiempo entre el destino que se haya escrito y nuestra localización. También se realiza este cálculo cuando se modifique el desplegable de la forma de viaje, siempre y cuando el campo de texto esté relleno.

Para realizar el cálculo del tiempo se usa la API de Google Distance Matrix. Esta API nos permite obtener en formato JSON información entre las dos coordenadas seleccionadas. En el método “getTripTime()” de esta actividad hacemos un request en forma de llamada GET en el cual se obtiene un JSON que trataremos para poder obtener la información que nos interesa. En este archivo podemos encontrar información como la distancia en km. Como solo nos interesa el tiempo, navegaremos por el hasta dar con el atributo en cuestión que será usado para mostrarlo en pantalla y para iniciar el viaje. De esta forma podemos informar al

usuario de cuánto tiempo en minutos aproximadamente tardará su viaje por los distintos métodos.

Una vez iniciado el viaje se pasarán como datos extras por un “Bundle” de Android al “Intent” realizado para el inicio de la siguiente actividad. Los datos extras son el tiempo, el origen y el destino que serán usados para contabilizar el tiempo restante y en el método de contacto escrito.

4.4.6. Actividad de Viaje

Esta última actividad tiene como misión el contar el tiempo que llevamos de viaje, mostrar cuanto le queda al usuario y realizar el contacto de emergencia cuando los tiempos de espera hayan sido agotados.

En la creación de la actividad se inicializa la base de datos para poder insertar el viaje en caso de ser completado, se obtienen las preferencias para acceder a los datos de usuario se crean los canales de notificación y se obtienen los datos extras obtenidos de las actividades anteriores. Posteriormente se dan los valores de máximo y mínimo de la rueda de números y se inicializa el contador.

Existe la posibilidad de añadir minutos al contador usando el botón de añadir minutos. Este botón obtendrá el número que está actualmente marcado en la rueda de números superior y añadirá dicho tiempo al contador. Dado que para el contador se usan milisegundos, se usa el objeto TimeUnit de JAVA que permite una conversión a distintas unidades de tiempo y con el cual usamos un formateo en el String de la interfaz del contador para mostrar el tiempo en minutos y segundos. Estas acciones de conversión se realizan en el método onTick() del contador que llevará la cuenta del tiempo que ha pasado.

Cuando el contador se acaba se iniciará el método onFinish(), se mostrará un “Toast” que es un mensaje que aparece en la pantalla durante unos segundos y vibrará el móvil para notificar al usuario que se acabó el tiempo, se mostrará una notificación en Android e

inicializará el contador de espera para el contacto de emergencia. Este contador tiene como tiempo límite 5 minutos y podrá ser detenido al añadir minutos, cancelar el viaje o indicar que se llegó al destino.

Cuando se indica la llegada al destino se hace una llamada a la base de datos para que inserte un nuevo trayecto y volvemos a la actividad principal a la espera.

4.5. Recursos

Los recursos de Android son archivos con valores de información que son usados en la aplicación móvil. Entre estos archivos también están los “layouts” para las interfaces. Se explicarán los más importantes.

4.5.1. Idiomas

Para los idiomas se usa el recurso de Android “strings.xml”. En este recurso se pueden asociar un identificador con un texto de forma que cuando existen varios archivos con distinta localidad asociada se usa uno u otro teniendo en cuenta el idioma sobre el que esté configurado el móvil. De esta forma usamos el identificador en los recursos de texto y se realiza una traducción.

4.5.1.Array

Para la lista de métodos de contacto de emergencia se usa un recurso llamada “array.xml”. Se pueden almacenar listas de “Strings” con un identificador. Se usa de la misma forma que “strings.xml”.

4.5.1.Layout

Los Layouts son recursos visuales que muestran las interfaces. Además de las interfaces principales también tiene la interfaz que se usa para rellenar la lista de viajes anteriores.

4.5.1.Drawable

Estos archivos xml son recursos visuales que van desde iconos hasta estilos para botones o backgrounds. En estos archivos se encuentran el icono que aparece en la notificación de emergencia, los estilos para los botones y los campos de texto editables y el background para los distintos layouts.

5. Pruebas

Para el desarrollo de las pruebas se usa la librería de Android UiAutomator y Espresso.

UiAutomator nos permite obtener el dispositivo simulado sobre el que realizaremos las distintas operaciones. Cuando ejecutemos el test, esta librería nos abrirá la aplicación para que podamos acceder a los elementos de la interfaz. Para ello, lo primero de todo será realizar una espera para que la interfaz de inicio cargue la actividad principal.

Una vez hemos llegado a la actividad principal, Espresso se ocupará de obtener de la vista los distintos objetos y realizar acciones sobre ellos. La obtención de estos elementos se realizará con los ID de los objetos que aparecen en la vista. Para aquellos elementos sin ID usaremos el parámetro del texto si lo poseen.

Las primeras pruebas están centradas en comprobar la navegabilidad de la aplicación. En ellas simplemente pulsamos los botones y comprobamos que nos encontramos en la nueva actividad.

A continuación, realizamos pruebas de la funcionalidad de la aplicación. Para ello haciendo uso de las mismas librerías navegamos hasta la interfaz que deseamos probar. Una vez nos encontremos en dicha actividad realizamos las pruebas necesarias y comprobamos que se ejecuta sin ningún error.

Un ejemplo de una prueba es la actualización de datos de usuario. Como precondition de la prueba tendremos la interfaz principal y el cambio de nombre de usuario como postcondición. Al ejecutar la prueba se obtiene que el resultado es satisfactorio conteniendo el nombre introducido en el texto de bienvenida de la aplicación principal.

Para probar la base de datos y el correcto funcionamiento de la lista de viajes anteriores realizamos una prueba en la cual insertaremos el valor de destino en la ventana de inicio del viaje. Una vez iniciado se procederá a confirmar directamente la llegada del viaje. Por último, comprobaremos si la base de datos tiene un elemento en su lista con un assert.

Por último, se realizó una fase de testeo con personas que probaron la aplicación y dieron feedback sobre los distintos errores posibles encontrados en la aplicación.

Uno de los errores encontrados consistía en la obtención del JSON de la API de Google Maps que daba como resultado el cierre de la aplicación.

6. Conclusiones y trabajos futuros

El proyecto se define por el diseño minimalista que evita el exceso de información en las interfaces y de dos funcionalidades claras.

El diseño minimalista está pensando para que personas mayores no tengan mayores dificultades a la hora de entender la aplicación. Esto a su vez exige que las interfaces tengan tamaños adecuados y que la información que deseen transmitir sea clara y concisa.

La primera funcionalidad principal de la aplicación es como método de seguridad para las personas mayores. Con la aplicación en uso los familiares o cuidadores de la persona que la usa serán informados en el momento que el usuario no llegue a algún destino y podrán tomar las medidas necesarias para garantizar su seguridad.

La segunda funcionalidad reside en obtener el tiempo aproximado que tomará realizar un viaje por los distintos métodos. De esta forma la persona que usa la aplicación escogerá la forma de viaje que le resulte más conveniente.

El principal problema reside en los métodos de contacto. Existen tres métodos de contacto los cuales son la llamada telefónica, el SMS y el WhatsApp. Sin embargo, la API de WhatsApp no permite el envío de mensaje sin una acción del usuario previa. Esto genera un problema importante, puesto que si el usuario no puede realizar dicha acción el contacto de emergencia no se realizará.

Como trabajo futuro se podría realizar un autocompletado del texto de destino de la aplicación de forma que el usuario pueda seleccionar exactamente el destino que desea de entre las opciones que se muestren. Se podría realizar la adición de un mapa junto con el formulario para poder definir exactamente la posición que el usuario desea como destino.

Otro trabajo futuro consistiría en realizar una obtención GPS de la posición del usuario en el momento del contacto de emergencia, revelando la posición más aproximada del usuario, de manera que el contacto de emergencia tenga información suficiente para poder actuar acorde.

Bibliografía

- [1] **Android** - https://www.android.com/intl/es_es/what-is-android/
- [2] **SQLite** <https://www.sqlite.org/index.html>
- [3] **JAVA** https://www.java.com/es/about/whatis_java.jsp?bucket_value=desktop-firefox68-osx1014&in_query=no
- [4] **Gradle Inc, 2020**, <https://gradle.org/>
- [5] **Android Studio** <https://developer.android.com/studio>
- [6] **XML** <https://www.w3.org/XML/>
- [7] **Github** <https://github.com/>
- [8] **GitKraken: © 2020 Axosoft, LLC** <https://www.gitkraken.com/>
- [9] **Emulator** <https://developer.android.com/studio/run/emulator>
- [10] **Postman** <https://learning.postman.com/docs/getting-started/introduction/>
- [11] **Documentación de Android** <https://developer.android.com/docs>
- [12] **Documentación de Google** <https://developers.google.com/>
- [13] **Documentación de Google Maps** <https://cloud.google.com/maps-platform/>
- [14] **Documentación de SQLite** <https://www.sqlite.org/docs.html>
- [15] **Stackoverflow** <https://stackoverflow.com/>
- [16] **Mikana** <https://github.com/danllopis/mikana>

Apéndice

Manual de uso

Al iniciar la aplicación principal se abrirá la pantalla de carga.



Ilustración 11: Inicio de la aplicación

Al ser la primera vez que se accede a la aplicación se pedirán permisos al usuario para poder ejecutar correctamente la aplicación.

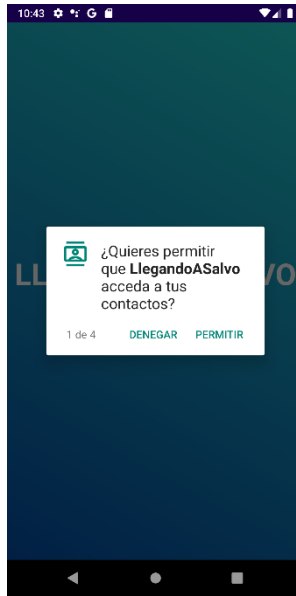


Ilustración 12: Inicio de la aplicación

Una vez hayan sido aceptados se accederá a la pantalla Datos de usuario.

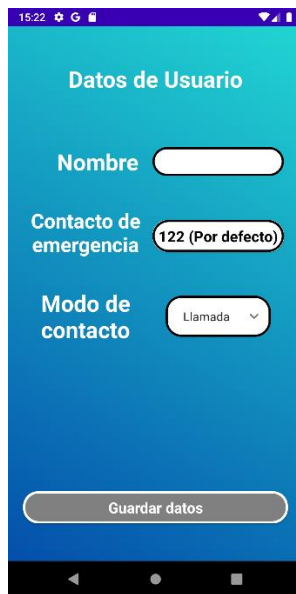


Ilustración 13: Datos de Usuario

En esta interfaz se debe introducir el nombre en el cuadro de texto blanco que aparece al principio en la ilustración. Si se pulsa el botón de contacto de emergencia aparecerá la lista de contactos del teléfono móvil y habrá que pulsar uno para seleccionar el teléfono que se

deseo. Una vez seleccionado, la aplicación volverá a la pantalla de datos. Por último, en el desplegable se puede seleccionar el método de contacto. Después de rellenar todos los campos del formulario se debe pulsar el botón de guardado.

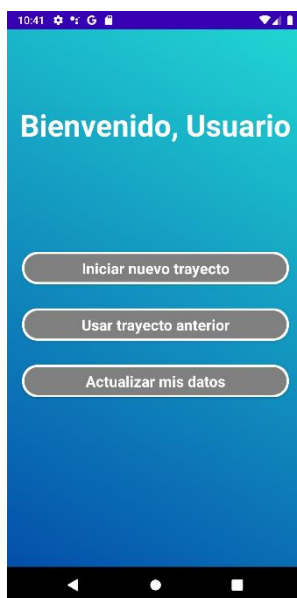


Ilustración 14: Interfaz Principal

Esta interfaz mostrada es la pantalla principal de la aplicación. Es la interfaz a la que se redirigirán todas las demás interfaces desde este momento. En ella tenemos tres botones que nos permiten realizar distintas acciones. Si pulsamos el primer botón, se nos redirige a la ventana de inicio de viaje.



Ilustración 15: Iniciar nuevo trayecto

En esta interfaz deberemos escribir el destino que deseemos y cambiar el método de viaje por el que queramos. Las opciones para viajar son andando, en coche y en transporte público. Una vez hayamos escrito el destino aparecerá el tiempo estimado hasta llegar a él como se ve en la siguiente ilustración.



Ilustración 16: Iniciar nuevo trayecto con tiempo

Si pulsamos atrás volveremos a la interfaz principal y si pulsamos iniciar viaje iniciaremos el viaje como se muestra en la siguiente imagen y se mostrará un texto que nos indica que se inició el viaje.



Ilustración 17: Viaje iniciado

En esta interfaz tenemos una rueda de números y tres botones, además del contador de tiempo. Si pulsamos el primer botón se añadirán tantos minutos como el número que aparezca en el centro de la rueda. Con el segundo botón, acabaremos el viaje y se guardará en la lista de viajes anteriores y volveremos a la interfaz principal. Con el último botón solo volveremos a la interfaz principal y se acabará el viaje, aunque no será guardado.

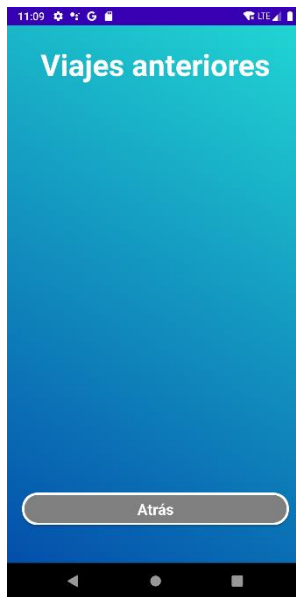


Ilustración 18: Viajes anteriores vacío

En esta interfaz se mostrarán los viajes anteriores que se completasen, es decir, habiendo pulsado el segundo botón de la interfaz del viaje. Cuando se inicia por primera vez la aplicación no se muestran puesto que no hay ninguna. En caso de existir se mostrarían como en la ilustración a continuación.



Ilustración 19: Viajes anteriores

Una vez tenemos trayectos anteriores se mostrará la información del viaje, así como el tiempo en minutos que hemos tardado en realizarlo y podemos pulsarlos para volver inmediatamente a la interfaz del viaje con el mismo ya iniciado.



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S. DE INGENIERÍA

E.T.S de Ingeniería Informática

Bulevar Louis Pasteur, 35

Campus de Teatinos

29071 Málaga