



UNIVERSIDAD DE MÁLAGA



INGENIERÍA DEL SOFTWARE

URBANGUARD: UNA PLATAFORMA SOCIAL PARA MEJORAR LA SEGURIDAD URBANA

URBANGUARD: A SOCIAL PLATFORM TO IMPROVE URBAN SAFETY

Realizado por
Nicolás Quintana Marín

Tutorizado por
Gabriel Jesús Luque Polo

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO DE 2022



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DEL SOFTWARE

URBANGUARD: UNA PLATAFORMA SOCIAL PARA MEJORAR LA SEGURIDAD URBANA

URBANGUARD: A SOCIAL PLATFORM TO IMPROVE URBAN SAFETY

Realizado por
Nicolás Quintana Marín

Tutorizado por
Gabriel Jesús Luque Polo

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2022

Fecha defensa: Julio de 2022

Resumen

Este Trabajo Fin de Grado, "UrbanGuard: Una Plataforma Social para Mejorar la Seguridad Urbana", consiste en el desarrollo de una aplicación móvil basada en servicios en la nube que ofrece a la ciudadanía, por el momento, de Málaga, una herramienta social para poder publicar delitos y que el resto de los usuarios puedan consultarlos. Además, ofrece una tanto una serie de estadísticas generales sobre los mismos, como la posibilidad de consultar estadísticas específicas sobre una zona o un tipo de delito concreto. Haciendo uso de las características ofrecidas por los dispositivos móviles actuales, permite el envío de señales de SOS de forma eficiente para avisar al resto de usuario de una situación de peligro y puedan llamar a la policía y ayudarnos. El desarrollo de este TFG supone una primera fase del proyecto, existe una segunda en la que, una vez recogidos los suficientes datos, se integrará un modelo de aprendizaje computacional capaz de recomendar zonas en función del instante temporal y el perfil del usuario. El principal objetivo de esta aplicación es por tanto permitir a los ciudadanos consultar toda la información de los delitos de forma eficiente, al alcance de su móvil, y ayudar a reducir las tasas de criminalidad permitiendo a la ciudadanía tomar decisiones basadas en información real.

Palabras clave: Ciudad Inteligente, Seguridad Urbana, Firebase, React, Ionic

Abstract

This Final Degree Project, "UrbanGuard: A Social Platform to Improve Urban Safety", consists of the development of a mobile application based on cloud services that offers the citizens of Malaga, for the time being, a social tool to publish crimes so that the rest of the users can consult them. In addition, it offers a series of general statistics on crimes, as well as the possibility of consulting specific statistics on an area or a specific type of crime. Using the features offered by current mobile allows SOS signals to be sent efficiently to warn the rest of the users of a dangerous situation so that they can call the police and help us. The development of this FDP is the first phase of the project; there is a second phase in which, once sufficient data has been collected, a computational learning model capable of recommending areas depending on the time and profile of the user will be integrated. Therefore, this application's main objective is to allow citizens to consult all the information on crimes efficiently, at the reach of their mobile phones, and to help reduce crime rates by enabling citizens to make decisions based on accurate information.

Keywords: Smart City, Urban Safety, Firebase, React, Ionic

Índice

1. Introducción	1
1.1 Motivación	1
1.2 Objetivos.....	3
1.3 Estructura de la memoria	4
2. Tecnologías y Herramientas de Desarrollo	7
2.1 Tecnologías del Servidor	7
2.1.1 Firebase	7
2.2 Tecnologías del Cliente	13
2.2.1 Ionic Framework	13
2.2.2 Capacitor	14
2.2.3 React	15
2.3 Herramientas de Desarrollo y Gestión	16
2.3.1 Visual Studio Code	17
2.3.2 Android Studio	17
2.3.3 GitHub.....	18
2.3.4 Diagrams.net.....	18
2.3.5 MagicDraw.....	19
2.3.6 Trello	19
3. Metodología Aplicada	21
4. Captura y Análisis de Requisitos	25
4.1 Requisitos Funcionales.....	25
4.2 Requisitos No Funcionales.....	26
4.3 Casos de Uso	27
4.3.1 Inicio de Sesión	27
4.3.2 Cierre de Sesión	28
4.3.3 Registro en la Plataforma	29
4.3.4 Cambio de Contraseña	30
4.3.5 Restaurar contraseña	31
4.3.6 Actualizar Información Personal.....	32

4.3.7 Borrar Perfil	33
4.3.8 Consulta de Delitos	34
4.3.9 Consulta de Estadísticas Generales	35
4.3.9 Consulta de Estadísticas Específicas	35
4.3.10 Publicación de Delitos.....	36
4.3.11 Consulta de Delitos Publicados.....	37
4.3.12 Eliminación de Delitos	38
4.3.13 Pulsación Botón del Pánico.....	39
5.Diseño y Arquitectura del Sistema	41
5.1 Diagrama de Arquitectura	41
5.2 Diagramas de Clases	42
5.2.1 Tipos de Datos	43
5.2.2 Capa Services	44
5.2.3 Interfaces de Usuario.....	46
5.3 Diagramas de Secuencia.....	47
5.4 Diagrama de Navegación.....	52
6. Desarrollo	55
6.1 Colecciones en Firestore	55
6.1.1 Colección Profiles.....	55
6.1.2 Colección Crimes	56
6.1.3 Colección Zones	57
6.1.4 Colección Stats.....	59
6.1.5 Colección SOS	60
6.1.6 Seguridad de las colecciones	61
6.2 Tipos de Delito y Condiciones de la Víctima	63
6.2.1 Tipos de Delito	63
6.2.2 Condiciones de Interés	66
6.3 Asignación de Zonas	69
6.4 Recomendador de Zonas.....	76
6.4.1 K-Nearest Neighbors.....	76
6.4.2 Desarrollo y Despliegue del Modelo.....	79
6.5 Diseño de la Interfaz de Usuario.....	81
7. Conclusiones y Líneas Futuras	87

Bibliografía	91
Apéndice A: Manual de Usuario	97
Inicio de Sesión	97
Registro	100
Recuperar contraseña	101
Modificar Perfil	102
Cambiar Contraseña	104
Eliminar Perfil	105
Filtrar y Visualizar Delitos	106
Publicar un Delito.....	109
Eliminar Delito	111
Visualizar Estadísticas.....	112
Envío de SOS	114
Apéndice B: Manual de Instalación	117
Instalación en Android.....	117
Instalación en local para desarrollo.....	117
Despliegue de las funciones.....	118

1. Introducción

En esta primera sección se va a realizar una descripción de los principales motivos por los que se ha llevado a cabo el trabajo fin de grado, cuáles son los principales objetivos que persigue y una escueta descripción de cómo se organiza y de que trata cada una de las partes de la memoria.

1.1 Motivación

El mundo lleva años viviendo una época de cambios intensos, la tecnología ha pasado a jugar un papel clave en cualquier sector que se pueda imaginar, llegando al punto de que se habla de estar en una nueva era de la historia, la Era Digital (Llamas, 2021). Esta nueva era está caracterizada por un proceso de transformación digital (Wikipedia, 2022), que consiste, de forma resumida, en aplicar las nuevas tecnologías, como son la Inteligencia Artificial, Big Data y el Internet de las Cosas (IoT) a cualquier aspecto de la sociedad humana.

Este fenómeno ya es visible y está teniendo un impacto en la sociedad, los expertos hablan de la cuarta revolución y del nacimiento de la Industria 4.0 (Alcácer & Cruz-Machado, 2019), en la que la industria tradicional da un paso adelante y alcanza un nivel superior de automatización gracias a la interconexión y digitalización de sus sistemas junto con la aplicación de tecnologías como las antes mencionadas, además de otras como pueden ser el Cloud Computing y la Realidad Aumentada.

Pero la industria no es el único sector que intenta adaptarse a la era digital, las propias ciudades evolucionan convirtiéndose en una Ciudad Digital o Smart City (Iberdrola, 2022). Esta ciudad es resultado de la aplicación de estas nuevas tecnologías a todos o varios de los procesos de una ciudad. Entre los posibles ámbitos de aplicación esta algo tan evidente como la gestión del tráfico, a través de, por ejemplo, semáforos inteligentes, movilidad y transporte

público adaptado, u otros incluso más interesante como la gestión de los recursos energéticos, residuos, espacios de ocio y una multitud más de posibilidades.

Ya hay evidencias de esos avances en ciudades como Málaga (Ayuntamiento de Málaga, 2022), existen aplicaciones que permiten visualizar en tiempo real la situación de los buses y el tiempo estimado de llegada a la parada, otras que informan del punto de recogida de residuos más cercano o la situación de los parkings de la ciudad, pero hay un gran olvidado, la seguridad urbana.

Y no es un tema para olvidar, mientras que la tasa de criminalidad presenta una ligera tendencia a la baja, las agresiones sexuales y violaciones, sobre todo a mujeres, presentan una tendencia alcista más que preocupante (EpData, 2022), intentar reducir este crecimiento ha sido una de las principales motivaciones para la realización de este Trabajo Fin de Grado.

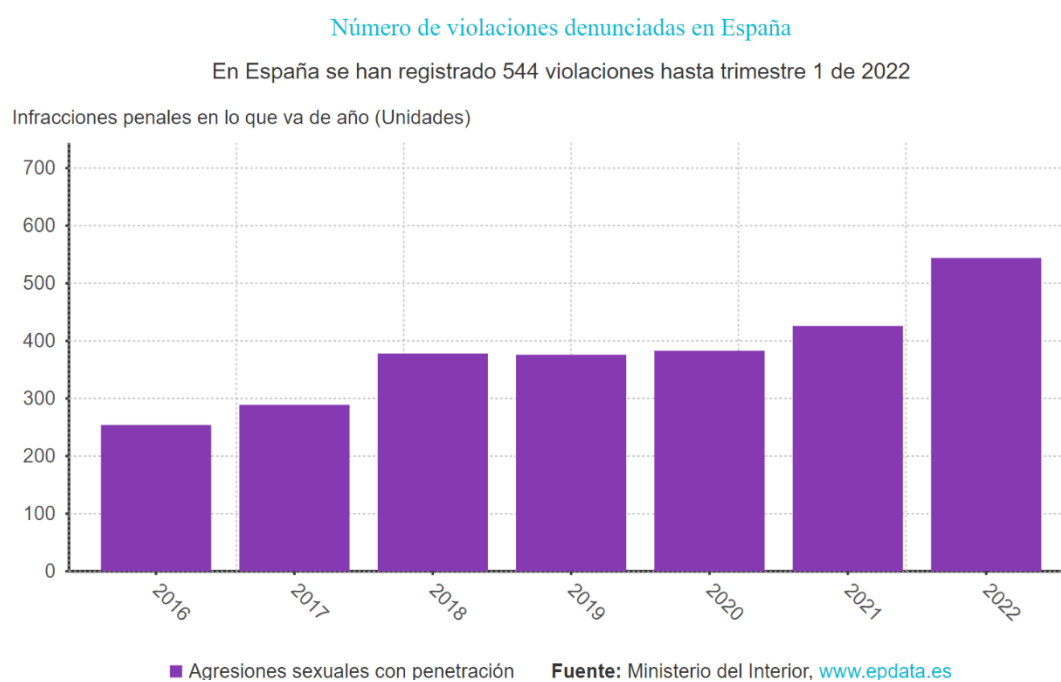


Figura 1.1.1: Número de violaciones denunciadas en España

Hoy en día no existe ningún portal accesible a los ciudadanos que les permita visualizar el estado del crimen en su ciudad, más allá de páginas como EpData (EpData, 2022) que ofrecen estadísticas sobre el número de delitos y su tipo, pero no extraen información de interés como puede ser el momento en el que se produjo, la zona de la ciudad o información de la propia víctima, toda esta información, ofrecida a la ciudadanía e incluso, a las fuerzas del orden,

puede ayudar a reducir la criminalidad a base de permitir a las fuerzas donde y cuando reforzar la vigilancia y, a los ciudadanos, que zonas evitar en cada momento. Además de esto, aprovechar la portabilidad de los smartphones para ofrecer nuevas funcionalidades de interés sin duda es una oportunidad de negocio que ayudaría a mejorar la calidad de una ciudad pionera tecnológicamente como Málaga.

1.2 Objetivos

Por tanto, el objetivo que persigue este Trabajo Fin de Grado es aprovechar esta oportunidad de negocio desarrollando una aplicación móvil, en la que los usuarios puedan registrarse y crear su perfil, reportar los delitos de los que ha sido víctima, haciendo uso de la ubicación del dispositivo, en los casos en los que sea adecuado. El hecho de que esta información se recoja sin necesidad de denunciar supone un gran avance, ya que, en multitud de ocasiones, la víctima no denuncia por miedo o desconfianza en las fuerzas del orden, gracias a esta aplicación, de una forma anónima y segura pueden reportarlo y que el resto de los ciudadanos sea consciente del mismo.

Además de la publicación de delitos, los usuarios podrían visualizar que delitos han sucedido en un día concreto, con información vinculante como la zona, la hora a la que ha sucedido y el tipo de delito del que se trata, sumando a esto información del perfil de la víctima como la edad, el sexo y condiciones de interés, como por ejemplo, pertenencia a la comunidad LGTBIQ+ o etnia minoritaria, también se podrían ofrecer una serie de estadísticas generales, por cada zona y por cada tipo de delito a todos los usuarios gracias a los datos que los propios usuarios publican.

Sumando a esta idea, se plantea ofrecer un sistema recomendador de zonas haciendo uso de estos datos, el perfil del usuario y técnicas de análisis de datos e Inteligencia Artificial que ofrezca recomendaciones basadas en el usuario y el momento temporal (día de la semana, hora y mes).

Por último, el hecho de que una gran cantidad de usuarios hagan uso de esta aplicación lo convierte en una importante herramienta, es por ello por lo que ofrecer un acceso a un botón

del pánico que notifique al resto de usuarios, con un simple click, de que se está en peligro, dando el nombre y la ubicación del emisor, permitiendo al resto de usuarios acudir en su ayuda y avisar a las autoridades, esta es una funcionalidad realmente interesante que complementa bien a las anteriores mencionadas.

En resumen, la funcionalidad se resume en los siguientes tres puntos:

- **Publicación y visualización de delitos**
- **Visualización de estadísticas y recomendador de zonas**
- **Botón del pánico**

No obstante, las complicaciones para obtener datos sobre delitos y sus víctimas hacen imposible desarrollar un recomendador de zonas funcional que ofrezca resultados de calidad, es por ello por lo que se propone un desarrollo basado en dos fases:

1. **Recogida de datos:** en esta fase se implementaría toda la funcionalidad relacionada a la publicación y visualización de delitos, visualización de estadísticas y botón del pánico, también se desarrollaría el modelo del recomendador y se dejaría listo para ser entrenado con los datos que se vayan obteniendo, sin ofrecerlo como servicio al usuario.
2. **Despliegue del recomendador:** una vez obtenidos los suficientes datos, se podría llevar a cabo un entrenamiento adecuado del recomendador y ofrecerlo al usuario como un servicio.

En este trabajo, por tanto, se llevará a cabo el desarrollo de la primera fase, dejando la segunda para líneas futuras, es por ello por lo que se implementará toda la funcionalidad mencionada en una aplicación móvil con la arquitectura típica cliente-servidor y se definirá el modelo de aprendizaje automático para el proceso de recomendación.

1.3 Estructura de la memoria

Una vez aclarados la motivación y los objetivos, la estructura de la memoria se corresponde con la siguiente:

1. **Introducción:** se plantean la motivación y los objetivos principales del proyecto.

2. **Tecnologías y Herramientas de Desarrollo:** se entra en detalle con las tecnologías seleccionadas para el desarrollo y las herramientas que han permitido la gestión y la implementación.
3. **Metodología Aplicada:** se detalla que modelo de metodología se ha seguido y sus aspectos más relevantes.
4. **Captura y Análisis de Requisitos:** se exponen los resultados obtenidos de la fase de captura de requisitos y su posterior análisis, contiene los propios requisitos y casos de uso.
5. **Diseño y Arquitectura:** se documentan las decisiones de diseño tomadas y el modelo de arquitectura, consiste, principalmente, en una serie de diagramas y su explicación.
6. **Desarrollo:** se abordan aquellas cuestiones del desarrollo de la aplicación que han sido más relevantes.
7. **Conclusiones y Líneas Futuras:** se comentan los pensamientos finales del desarrollo del proyecto y los posibles caminos a tomar.

En el resto del documento se expone la bibliografía consultada, así como una serie de apéndices con el manual de usuario y el de instalación.

2. Tecnologías y Herramientas de Desarrollo

Esta sección tiene el objetivo de documentar todas las tecnologías y herramientas utilizadas a la hora de llevar a cabo el ciclo de desarrollo del proyecto software, además de justificar su elección.

2.1 Tecnologías del Servidor

En primer lugar, se van a exponer las tecnologías escogidas en el lado del servidor, donde se realizará la persistencia de los datos, la autenticación, y ciertos cálculos sobre los datos.

2.1.1 Firebase



Figura 2.1.1: Logo de Firebase

Para el desarrollo del backend se ha decidido elegir Firebase (Google Developers, 2022), esta es una plataforma orientada a facilitar el desarrollo de aplicaciones webs, móviles y juegos que está respaldada por la comunidad de Google que, integrada con los servicios de computación en la nube de Google, Google Cloud Platform (Google, 2022), ofrece una serie de servicios a

los desarrolladores para llevar a cabo la gestión del backend de forma eficiente, automatizando el escalado y permitiendo generar como resultado aplicaciones de alta calidad, al estar amparado por los servicios de Google.

Se ha escogido esta plataforma por las evidentes ventajas que ofrece a la hora de desarrollar, sobre todo, una aplicación móvil, ya que los servicios que ofrece y la infraestructura que aporta es la perfecta para el desarrollo de una aplicación móvil de estas características, además la existencia de una gran cantidad de documentación para el desarrollo en Internet y de permitir la sincronización entre proyectos de distinto tipo, permitiendo que, si en un futuro, la aplicación se expande y se ofrece, por ejemplo, una aplicación de escritorio o incluso un portal web, el desarrollo de estos no suponga realizar grandes cambios en el backend.

Firebase ofrece multitud de servicios, de los cuales se han utilizado para el desarrollo los siguientes.

2.1.1.1 Firebase Authentication

Entre los servicios más interesantes que ofrece Firebase esta Authentication (Google Developers, 2022), ofrece, además de todos los servicios de backend necesarios para la autenticación necesarios para la gestión de los usuarios, una serie de SDKs adaptados a cada plataforma que permiten acceder a estos servicios de una forma intuitiva y fácil de aprender.

Además de la clásica autenticación mediante usuario y contraseña, Firebase nos brinda la posibilidad de ofrecer al usuario múltiples formas de inicio de sesión, como pueden ser a través de número de teléfono o entidades federadas, como la propia Google, además de otras muchas, para las que ofrece una biblioteca de UI para agilizar y asegurar el proceso.

Al usar estándares como OAuth 2.0 y OpenID, nos ofrece también la posibilidad de integrarlo con un backend personalizado, además de ofrecer una buena capa de seguridad que nos ofrecen los propios estándares y las bibliotecas de Google.

En este caso, se ha utilizado Authentication para la manejar la autenticación y autorización de usuarios, de forma que sea eficiente, segura y escalable a otros modos de autenticación en el futuro.

2.1.1.2 Cloud Firestore

Otro de los servicios más interesantes de Firebase es Firestore (Google Developers, 2022), es una base de datos NoSQL que ofrece flexibilidad y escalabilidad. Para la gestión de estos datos, Firebase ofrece la posibilidad de realizar consultas únicas o generar objetos de escucha en tiempo real, con una optimización excelente, además de ofrecer un soporte sin conexión.

En cuanto al modelo de datos que utiliza, está orientado a documentos, que a su vez se almacenan en colecciones. Estos documentos están compuestos de pares clave-valor, otras colecciones y objetos anidados. Firestore está optimizado para manejar una gran cantidad de documentos pequeños, por lo que es importante orientar el modelo de datos a este estilo.

Dentro de los pares clave valor, admite los siguientes tipos de valores:

- Array
- Boolean
- Bytes
- Datetime
- Float
- Position (no recomendado)
- Integer
- Map
- Null
- Referencia
- String

Sobre este modelo de datos, Firestore ofrece un sistema de consultas muy expresivo, bastante similar al clásico SQL, que permite recuperar un único documento o una serie de ellos que cumplan con una serie de filtros, fácilmente encadenables y permitiendo elegir el criterio de orden. Lo realmente interesante es que permite la indexación de búsquedas, por tanto, la

eficiencia de estas se vuelve proporcional al conjunto resultado del filtrado, no al conjunto total de los datos.

Firestore también tiene una alta compatibilidad con el resto de los servicios de Firebase, uno de los más destacados es la facilidad que ofrece para incluir una capa de seguridad, en concreto, autenticación y autorización a través del servicio Authentication. De esta forma es posible, a través de definir una serie de reglas de seguridad en formato JSON, de una forma simple y realmente expresiva, definir qué condiciones son necesarias para acceder a ciertas colecciones y documentos, incluso restringir el acceso a ciertos campos del documento. Esta metodología se llevó a cabo para accesos de cliente, en el caso del servidor se hace uso de la propia Administración de Identidades y Accesos (IAM).

La otra integración más interesante tiene lugar con Cloud Functions, servicio que se explicará más adelante. De forma resumida, este servicio permite establecer triggers que se ejecutan con la creación, actualización y eliminación de documentos, lo que permite generar mucha funcionalidad interesante.

Pero no todo son ventajas, Firestore presenta una serie de limitaciones, entre las que han afectado, ya sea en mayor o menor medida a este proyecto, se encuentran:

- Las cláusulas de filtrado "in", "not-in" y "array-contains-in" admite hasta 10 valores.
- Solo es posible encadenar una cláusula "in", "not-in" y "array-contains-in" por consulta, sin poder combinarlos entre ellos.
- Para evitar problemas de latencia, se recomienda limitar la escritura en un mismo documento a 1 por segundo.

Estas limitaciones no han supuesto un grave problema a la hora de llevar a cabo el desarrollo, pero han limitado de forma menor, algunas de las funcionalidades planteadas.

En el caso de Firestore, se ha utilizado como base de datos principal de la aplicación, en ella se almacenará toda la información relevante a perfiles de usuario, crímenes, estadísticas, etc.

2.1.1.3 Cloud Storage

Cloud Storage (Google Developers, 2022) es el servicio de almacenamiento de objetos en la nube de Firebase que ofrece una alta potencia, escalabilidad y rentabilidad. Permite el almacenamiento de cualquier contenido que el usuario pueda generar, como, por ejemplo, fotos, videos y audio.

Este almacenamiento se realiza en un bucket de Google Cloud Storage y ofrece, de la misma forma que Firestore, un sistema similar para definir reglas de seguridad y limitar el acceso a los objetos.

También tiene soporte para conexiones débiles, de forma que, si la transferencia se interrumpe, el cliente puede retomar la transferencia por el momento en el que se quedó. Cabe destacar que también ofrece procesamiento avanzado como filtrado de imágenes y transcodificación de video, en el caso de que sea necesario.

Cloud Storage será el sistema de almacenamiento donde se reúnen todas las imágenes de perfil de los usuarios.

2.1.1.4 Cloud Functions

Cloud Functions (Google Developers, 2022) es un framework, ofrecido por Firebase, que permite el desarrollo y despliegue de funciones en el backend en respuesta a eventos que pueden ser, principalmente, triggers definidos por el desarrollador o llamadas HTTPS.

La implementación de estas funciones se ejecutan en NodeJS con un modelo serverless (Amazon Web Services, 2022), es decir, Firebase asigna los servidores y recursos necesarios para la ejecución en el momento que lo necesita, aplicando escalabilidad y adaptando la facturación a esta ejecución. Al ejecutarse en NodeJS, admite la implementación en JavaScript y TypeScript y son fácilmente desplegables mediante un simple comando de consola.

Para facilitar el desarrollo y las pruebas, Firebase ofrece un emulador local para probar el funcionamiento de las funciones y evitar perder tiempo realizando el despliegue con cada vez

que se desee probar una función. Las funciones se despliegan de forma independiente entre sí y con su propio entorno y configuración.

En el entorno de Cloud Functions será donde se actualicen las estadísticas al publicar o eliminar un delito y donde se realice el envío de señales de SOS.

2.1.1.5 Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) (Google Developers, 2022) es una solución ofrecida por Firebase para ofrecer un servicio de mensajería multiplataforma orientada a notificaciones, de una forma segura.

Se compone de dos actores principales, un entorno seguro, como Cloud Functions, para la generación y envío de mensajes y un conjunto de aplicaciones listas para recibirlos.

El protocolo de comunicación se basa los dos actores mencionados, entrando más en detalle, el cliente se registra para poder recibir mensajes y recibe un token que le identifica inequívocamente, para que el servidor pueda enviarle mensajes. Para facilitar el envío a múltiples dispositivos, un cliente puede suscribirse a un tema, de forma que el servidor puede enviar mensajes que reciban todos los usuarios suscritos a un tema. Además de esto, también es participe la capa de transporte de la plataforma en cuestión y el propio backend de FCM, la Figura 2.1.2 sirve como referencia del proceso.

FCM se ha utilizado para gestionar el envío de notificaciones de forma eficiente y segura.

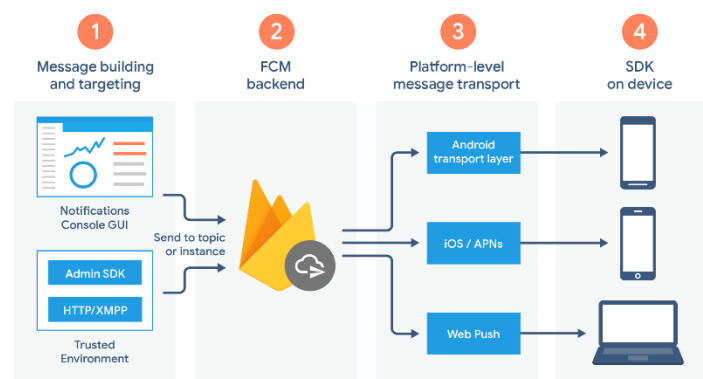


Figura 2.1.2: Modelo de Arquitectura de FCM

2.2 Tecnologías del Cliente

Una vez vistas las tecnologías del lado del servidor, vamos a explorar que tecnologías se han elegido para desarrollar el cliente.

2.2.1 Ionic Framework



Figura 2.2.1: Logo de Ionic Framework

Ionic Framework (Ionic, 2022) es un conjunto de herramientas open-source que permiten el desarrollo de interfaces de usuario de calidad para aplicaciones de escritorio y móviles haciendo uso de las tecnologías de desarrollo web, combinando por tanto las posibilidades de diseño y la experiencia de usuario de una aplicación web con el rendimiento de una aplicación nativa.

Como se ha mencionado, Ionic hace uso de tecnologías web (HTML, CSS, JavaScript) y permite el desarrollo mediante únicamente JavaScript o mediante integraciones con los frameworks de desarrollo web más populares, pudiendo elegir a preferencia del desarrollador entre Angular, React y Vue.

Al desarrollarse mediante tecnologías web, permite el despliegue de las aplicaciones de forma nativa, haciendo uso de herramientas como Capacitor o Cordova o en forma de aplicación web tradicional ofreciéndose en forma de Progressive Web App (MDN, 2020), que, de forma resumida, consiste en ofrecer la funcionalidad de una aplicación nativa, pero accediendo desde el navegador, sin necesidad de descargarla y haciendo uso de APIs y funciones emergentes del propio navegador.

La primera impresión que puede dar este conjunto de herramientas es que, al tratarse de interfaces web, pueden ser poco eficientes y resultar un problema para el desarrollo móvil, pero esto no es un problema, a pesar de no poder ofrecer toda la potencia y el rendimiento

de un desarrollo completamente nativo, Ionic está completamente optimizado para ofrecer un increíble rendimiento en todos los dispositivos móviles, haciendo uso de técnicas como transiciones aceleradas por hardware y esta optimizado para la gestión de la pantalla táctil.

Además de toda esta funcionalidad, ofrece un increíble conjunto de componentes de interfaz diseñadas para ofrecer una buena experiencia de usuario y permitir realizar un diseño responsive, desde botones, listas y campos de texto, hasta mallas para el diseño de la pantalla, menús y herramientas para refrescar contenido.

Por último, Ionic ofrece una Command Line Interface (CLI) necesaria para la creación de la plantilla del proyecto, generación del proyecto en plataformas nativas y desplegar en local un servidor de desarrollo, además de un conjunto muy amplio de funcionalidades interesantes para el desarrollo.

Ha sido la tecnología base del frontend, a partir de combinaciones con otros frameworks se ha desarrollado la interfaz de la aplicación.

2.2.2 Capacitor



Figura 2.2.2: Logo de Capacitor

Capacitor (Capacitor, 2022) es una herramienta que permite desarrollar páginas web que se ejecutan de forma nativa en iOS, Android y Web, ofrece una serie de APIs que permiten a la aplicación seguir ofreciendo la experiencia web, sin perder de vista las ventajas del desarrollo en nativo.

Capacitor esta desarrollado por la misma comunidad que Ionic, por lo que la compatibilidad entre ambos es más que evidente y es la evolución de Apache Cordova, de forma que la gran mayoría de plugins desarrollados para Cordova son compatibles también con Capacitor.

Y ese es el verdadero potencial de Capacitor, la propia herramienta ofrece una serie de plugins que hacen uso de SDKs nativos para hacer uso de funcionalidades como la cámara, el almacenamiento interno o el sistema de notificaciones, pero no solo eso, tiene una inmensa comunidad detrás que está constantemente desarrollando nuevos plugins de forma gratuita.

Para el desarrollo de este proyecto, se han usado los siguientes plugins:

- **Camera** (Capacitor, 2022): permite el acceso a la cámara del dispositivo y realizar fotos con la misma.
- **Cordova/Image-Picker** (Ionic, 2022): permite seleccionar una imagen existente de la galería del usuario.
- **Geolocation** (Capacitor, 2022): permite acceder a la ubicación del dispositivo haciendo uso del GPS.
- **Push Notifications** (Capacitor, 2022): ofrece una API para facilitar la integración con Firebase Cloud Messaging y recibir notificaciones.
- **Comunidad/FCM** (Silva & Pereira, 2021): amplía la funcionalidad de Push Notifications, permitiendo suscribir o anular la suscripción de un usuario a un tema concreto.

Ha sido la principal fuente de funcionalidad a la hora de obtener servicios nativos del dispositivo móvil.

2.2.3 React



Figura 2.2.3: Logo de React

React (Meta Platforms, 2022) es una biblioteca de JavaScript de código abierto desarrollada por una gran comunidad de usuarios e impulsada por Meta, anteriormente Facebook, ideal para el desarrollo de interfaces de usuario web de una sola página, sigue el patrón de arquitectura modelo-vista-controlador, desempeñando el rol de vista.

Sigue un modelo declarativo, lo que ofrece el código sea expresivo y más predecible a la hora de realizar pruebas, además de que React se encarga del renderizado y la actualización de los componentes en el momento adecuado y de forma eficiente.

Esta principalmente basado en componentes, de forma es posible definir componentes, que manejan su propio estado y comportamiento, ofreciendo encapsulamiento. El desarrollo de estos componentes se realiza en JavaScript (o TypeScript, que es el caso de este proyecto), de forma que la gestión de los datos de la aplicación resulta más sencilla.

Al tener una inmensa comunidad detrás, y el respaldo de una importante empresa como Meta, lo hace una interesante opción, ya que ofrece, además de una curva de aprendizaje bastante asequible, una gran cantidad de documentación y de bibliotecas que amplían la funcionalidad, de los que se han utilizado los siguientes:

- **Firebase** (Google Developers, 2022): SDK oficial de Firebase, ofrece toda la funcionalidad necesaria para hacer uso de los servicios.
- **React-Router** (Remix, 2022): biblioteca para gestionar la navegación en React a través de distintas URL, manteniendo el modelo de una sola página.
- **React-Google-Maps/Api** (Lyakhov & Klar, 2022): integra los servicios de la API de Google Maps en componentes de React, de estos servicios se han usado el propio sistema de Mapas y Distance Matrix, que permite calcular la distancia y el tiempo de llegada entre varios puntos de origen y destino, según el medio de transporte.
- **Recharts** (Recharts Group, 2021): ofrece componentes de React para la visualización de datos en forma de distintas gráficas.

2.3 Herramientas de Desarrollo y Gestión

En último lugar, se van a documentar las herramientas utilizadas para realizar el desarrollo del proyecto y su gestión.

2.3.1 Visual Studio Code

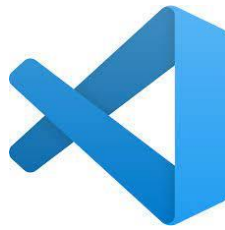


Figura 2.3.1: Logo de Visual Studio Code

Como entorno de desarrollo se ha elegido Visual Studio Code (Microsoft, 2022), un editor de código fuente gratuito desarrollado por Microsoft. Contiene toda la funcionalidad necesaria para el desarrollo de proyectos a gran escala para casi cualquier lenguaje, entre la que destaca la depuración, resaltado de sintaxis, recomendaciones mediante IA y refactorización.

Tiene una increíble posibilidad de personalización gracias a su gran número de extensiones disponibles que ofrecen desde soporte a Git mediante una GUI, hasta herramientas para la edición y depuración de código en tiempo real con más de una persona, mediante trabajo colaborativo.

2.3.2 Android Studio



Figura 2.3.2: Logo de Android Studio

Android Studio (Google Developers, 2022) es un entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Además de todas las funcionalidades y ventajas propias de IntelliJ IDEA (JetBrains, 2022) ofrece una amplia gama de mejoras como un emulador integrado, compatibilidad con Google Cloud y la posibilidad de aplicar cambios en la aplicación sin tener que reiniciarla.

Para este desarrollo, se ha usa Android Studio para, una vez generado el proyecto nativo Android desde Ionic Capacitor, poder generar un apk y realizar pruebas sobre un dispositivo físico.

2.3.3 GitHub



Figura 2.3.3: Logo de GitHub

A fin de poder ejecutar una buena gestión de la configuración sobre el proyecto, además de poder alojar el código fuente en un entorno seguro para evitar problemas de pérdida, se ha utilizada GitHub (GitHub, 2022), un gestor de repositorios en la nube que permite la gestión de código mediante el sistema Git (Software Freedom Conservancy, 2022), el sistema de control de versiones más extendido en la actualidad.

Obviando la posibilidad de crear versiones y publicarlas, permite el desarrollo en paralelo, mediante ramas, ideal para equipos de desarrollo en los que se asignan funcionalidades a distintos miembros, también ofrece la posibilidad de unir ramas, informando de las situaciones de conflicto para que el usuario pueda tomar una decisión.

2.3.4 Diagrams.net



Figura 2.3.4: Logo de Diagrams.net

Diagrams.net (JGraph Ltd, 2021) es un stack de tecnologías open-source para el desarrollo de diagramas de aplicaciones. Ofrece, a través de una interfaz web una increíble cantidad de herramientas para el diseño, tanto de diagramas como pueden ser de clases, de navegación, de flujo, hasta pequeños mockups y prototipos de la interfaz de usuario.

2.3.5 MagicDraw



Figura 2.3.5: Logo de MagicDraw

MagicDraw (Dassault Systèmes, 2022) es una herramienta de modelado software profesional, además de ofrecer soporte al metamodelo de UML 2, plantea una amplia gama de posibilidades, como generación de patrones de arquitectura y diseño, generación del esqueleto del código a raíz del modelo y herramientas para la simulación y validación de modelos en tiempo real, es por ello por lo que se trata de una de las herramientas de modelado más populares de la actualidad.

2.3.6 Trello



Figura 2.3.6: Logo de Trello

Trello (Atlassian, 2021) es una plataforma web para la administración y gestión de proyectos. Permite la creación de tableros en los que se pueden definir listas que, a su vez pueden contener tarjetas, que pueden pertenecer a distintos grupos.

De esta forma, es fácil la creación de tableros tipo Scrum Board o Kanban en los que se definen las distintas tareas, el tipo de tarea del que se trata y el estado de estas, algo indispensable para la ejecución de metodologías ágiles como las mencionadas.

3. Metodología

Aplicada

En el campo del desarrollo de software y en la gestión de proyectos, resulta esencial y, en una gran cantidad de casos, decisivo aplicar una metodología adecuada a las cualidades propias del software. Es bien sabido por los expertos, que la aplicación de metodologías ágiles, caracterizadas por su adaptación al cambio, involucración del cliente y constante retroalimentación (Cadavid Navarro, Fernández Martínez, & Morales Vélez, 2013), es un factor, cada vez más decisivo a la hora de llevar a cabo un proyecto software, independientemente de su complejidad (Díaz Piraquive, Gonzalez Crespo, Pérez Castillo, & Medina García, 2015).

Es por ello, por lo que se ha aplicado una metodología basada en Scrum (Rubin, 2012), adaptada, obviamente, a un desarrollo en solitario. Para lograr este objetivo, se han definido las siguientes fases de trabajo, compuestas por sprints de 2 semanas de duración:

- **Formación en las tecnologías (1/2 sprint):** en esta primera fase de introducción al proyecto se ha llevado a cabo un estudio de la documentación de las tecnologías seleccionadas, combinado con el visionado de videos explicativos de las mismas y algunos repositorios de ejemplos.
- **Captura y análisis de requisitos (1/2 sprint):** en esta fase se ha llevado a cabo el proceso de extracción de requisitos, se ha detallado cual será la funcionalidad que implementar y se ha documentado mediante el listado de requisitos y los casos de uso.
- **Diseño (1/2 sprint):** en esta fase se han tomado todas las decisiones referentes al diseño de la aplicación, tanto a nivel de arquitectura, como a nivel de modelado e

incluso interfaz de usuario, se han definido por tanto los modelos y diagramas necesarios.

- **Desarrollo y pruebas de la autenticación y gestión de usuarios (1/2 sprint):** esta fase ha consistido en la implementación de toda la funcionalidad asociada al proceso de autenticación y gestión de la información de los usuarios de la aplicación.
- **Desarrollo y pruebas de las funcionalidades referentes a la publicación/consulta de delitos (3 sprints):** esta ha sido la fase más compleja de la aplicación, ha consistido en la implementación de toda la funcionalidad para la publicación de delitos, su posterior visualización y la generación de las estadísticas.
- **Desarrollo y pruebas de la funcionalidad “Botón del pánico” (2 sprints):** esta ha sido la última fase del desarrollo, se ha implementado toda la funcionalidad para permitir al usuario enviar mensajes de socorro y, al usuario receptor, ofrecer una buena visualización del mensaje, además de obviamente, recibirlo.
- **Elaboración de la Memoria del TFG y documentación relacionada (1 sprint):** por último, se ha procedido con la redacción de esta memoria y la documentación relacionada.

Se plantean por tanto un total de 8 sprints, en los que, al tratarse de un desarrollo ágil, realmente algunas fases como las de captura de requisitos y diseño se retoman al encontrarse cara a cara con los problemas del desarrollo, pero, a grandes rasgos, resulta en la planificación planteada en un diagrama de Gantt de la Figura 3.1.

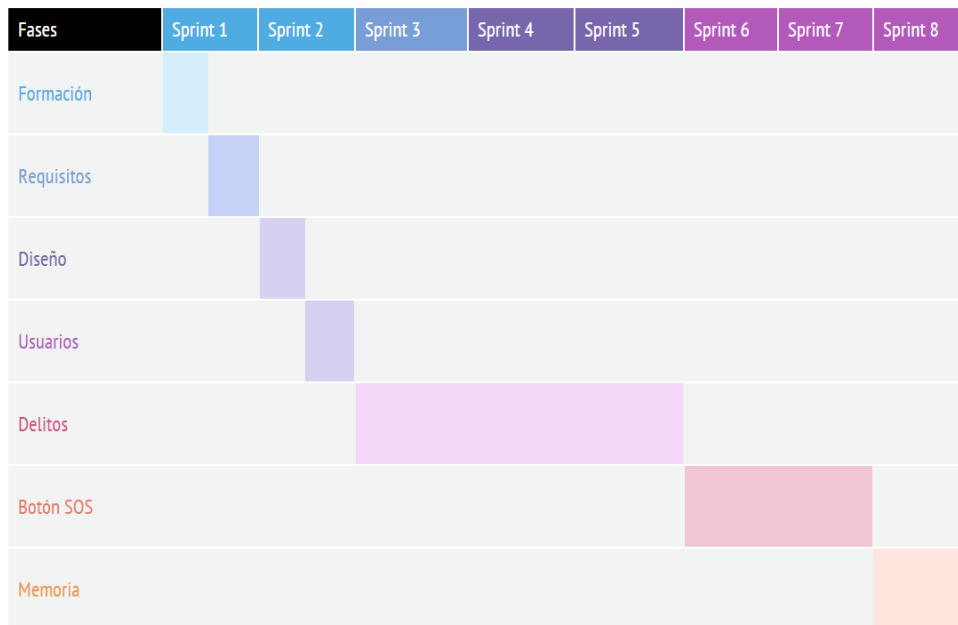


Figura 3.1: Diagrama de Gantt del proyecto

En cuanto a la gestión de las tareas, como ya se ha mencionado, se ha usado Trello con la distribución de tareas propia de Scrum **To Do - In Progress - Done**, donde To Do son las tareas pendientes del sprint, In Progress, aquellas que ya están comenzadas, y Done, las que ya están cerradas.

Además, se han añadido las listas Tasks, que corresponde con el Product Backlog que define Scrum, es decir, un listado con todas las tareas identificadas hasta el momento y que deben ser acabadas para que el proyecto se dé por concluido. También se han añadido listas para gestionar los problemas encontrados durante el desarrollo. En la Figura 3.2 se puede observar una captura de pantalla del tablero de Trello durante el desarrollo del proyecto.

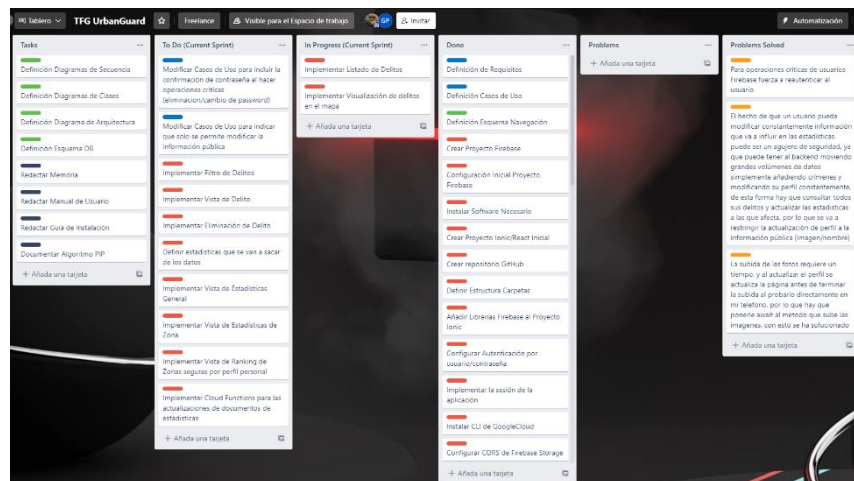


Figura 3.2: Scrum Board del proyecto

A un nivel más personal, la gestión diaria de las tareas se ha llevado a cabo seleccionando diariamente un subconjunto de tareas, lo más relacionadas posible, que se estimen en una duración total de entre 3 y 4 horas, aplicando esta norma, se ha logrado generar una rutina de trabajo con éxito, de forma que se realizaban pequeños avances día a día, sin excepción, pero, a su vez, sin provocar una sobrecarga de trabajo que conlleven a una desmotivación a la hora de continuar con el proyecto.

4. Captura y Análisis de Requisitos

En esta sección, se muestran los resultados obtenidos de la fase inicial de captura y análisis de requisitos.

4.1 Requisitos Funcionales

Se exponen a continuación el listado de requisitos a implementar, agrupados por el subgrupo al que pertenecen.

ID	Descripción
Autenticación y Usuarios	
RF-01	Un usuario debe poder iniciar sesión con su correo electrónico y contraseña
RF-02	Un usuario debe poder cerrar sesión
RF-03	Un usuario debe poder registrarse en la aplicación con su correo electrónico, contraseña e información personal
RF-04	Un usuario debe poder cambiar su contraseña
RF-05	Un usuario debe poder restaurar su contraseña
RF-06	Un usuario debe poder modificar su información personal
RF-07	Un usuario debe poder eliminar su perfil de la plataforma
Consulta/Publicación de Delitos	
RF-08	Un usuario debe poder visualizar en un listado los delitos registrados en la plataforma
RF-09	Un usuario debe poder visualizar en un mapa los delitos registrados en la plataforma
RF-10	Un usuario debe poder filtrar los delitos registrados en la plataforma

RF-11	Un usuario debe poder consultar las estadísticas generales
RF-12	Un usuario debe poder consultar las estadísticas de una zona concreta
RF-13	Un usuario debe poder consultar las estadísticas de un tipo de delito concreto
RF-14	Un usuario debe poder publicar un delito en la plataforma
RF-15	Un usuario debe poder consultar sus delitos publicados
RF-16	Un usuario debe poder eliminar un delito que haya publicado en la plataforma
Botón del Pánico	
RF-17	Un usuario debe poder enviar una señal de socorro de forma rápida y eficaz
RF-18	Un usuario debe recibir una notificación cuando otro envíe una señal de socorro
RF-19	Un usuario debe poder ver la información de quien envió la señal de socorro que recibió y la ubicación desde la que se lanzó

Tabla 4.1.1: Listado de requisitos funcionales

4.2 Requisitos No Funcionales

En este caso, se muestra el conjunto de requisitos no funcionales, esta vez, agrupados por categoría.

ID	Descripción
Seguridad/Privacidad	
RNF-01	Solo podrá existir una cuenta por correo electrónico
RNF-02	El correo electrónico debe ser válido
RNF-03	Ningún usuario podrá acceder a la información personal sensible de otro
RNF-04	El propio usuario será el único capaz de modificar su información personal
RNF-05	Será necesario estar autenticado para publicar un delito
RNF-06	Será necesario estar autenticado para enviar un SOS
Eficiencia	
RNF-07	La aplicación debe ejecutarse, como mínimo, en el sistema operativo Android 8 (API 26)
RNF-08	Las notificaciones de SOS deben recibirse, a lo sumo, 5 segundos tras su envío

Usabilidad	
RNF-09	La interfaz debe ser “responsive”, es decir, debe adaptarse al dispositivo en el que se ejecute
RNF-10	La interfaz ofrecida debe ser fácil de usar para los usuarios y no debe suponer una dificultad para el mismo
RNF-11	La interfaz debe tener una paleta de colores adecuada y evitar problemas de confusión debido al color

Tabla 4.2.1: Listado de requisitos no funcionales

4.3 Casos de Uso

A continuación, se detalla un listado con cada caso de uso identificado, explorando sus escenarios y características de interés.

4.3.1 Inicio de Sesión

Título	Inicio de sesión
Descripción	El usuario introduce su información de autenticación y el sistema comprueba si es correcta, en caso de serlo se produce la autenticación, en otro caso se informa del error
Pre-condición	El usuario ha abierto la aplicación y no ha iniciado sesión aún
Post-condición	El usuario accede a la aplicación de forma autenticada
Prioridad	Alta
Requisitos	RF-01 RNF-01 RNF-02
Escenario principal	

1. El usuario pulsa en “Iniciar sesión” 2. El sistema muestra la pantalla de inicio de sesión 3. El usuario introduce su correo electrónico 4. El usuario introduce su contraseña 5. El usuario pulsa en “Iniciar sesión” 6. El sistema comprueba que existe una cuenta asociada al correo electrónico 7. El sistema comprueba que la contraseña es correcta 8. El sistema comprueba que el correo electrónico está verificado 9. El sistema autentica al usuario y muestra la pantalla principal de la aplicación, de forma autenticada
Escenarios alternativos
6.b0 El sistema comprueba que no existe una cuenta asociada al correo electrónico 6.b1 El sistema muestra un mensaje de error indicando que alguno de los campos es incorrecto
7.b0 El sistema comprueba que la contraseña es incorrecta 7.b1 El sistema muestra un mensaje de error indicando que alguno de los campos es incorrecto
8.b0 El sistema comprueba que el correo electrónico no está verificado 8.b1 El sistema muestra la pantalla principal de la aplicación, sin autenticar al usuario 8.b2 El sistema muestra un mensaje al usuario informando de que debe verificar el correo electrónico para acceder a la misma 8.b3 El sistema envía un email a la cuenta de correo electrónico con el enlace necesario para verificarlo

Tabla 4.3.1: Caso de Uso "Inicio de Sesión"

4.3.2 Cierre de Sesión

Título	Cierre de sesión
Descripción	El usuario cierra sesión y accede a la aplicación de forma anónima
Pre-condición	El usuario ha abierto la aplicación y ha iniciado sesión
Post-condición	El usuario accede a la aplicación de forma anónima
Prioridad	Alta
Requisitos	RF-02
Escenario principal	

1. El usuario pulsa en “Cerrar sesión” 2. El sistema muestra un dialogo preguntando al usuario si realmente quiere cerrar la sesión 3. El usuario pulsa en “Si” 4. El sistema cierra la sesión y muestra la pantalla principal de la aplicación, de forma anónima
Escenarios alternativos
3.b0 El usuario hace pulsa en “No” 3.b1 El sistema cierra el dialogo y vuelve a mostrar la pantalla en la que se encontraba

Tabla 4.3.2: Caso de Uso "Cierre de Sesión"

4.3.3 Registro en la Plataforma

Título	Registro en la plataforma
Descripción	El usuario introduce su información de registro, el sistema comprueba que la información sea válida, lo almacena en la base de datos y envía un email para poder verificar el correo electrónico
Pre-condición	El usuario ha abierto la aplicación y no esta registrado en el sistema
Post-condición	El usuario ha creado una cuenta en el sistema
Prioridad	Alta
Requisitos	RF-03 RNF-01 RNF-02
Escenario principal	

<ol style="list-style-type: none"> 1. El usuario pulsa en “Registrarse” 2. El sistema muestra la pantalla de registro 3. El usuario introduce su correo electrónico 4. El usuario introduce su nombre y apellidos 5. El usuario selecciona una foto de su galería 6. El usuario introduce su contraseña 7. El usuario vuelve a introducir su contraseña en el campo de confirmación de contraseña 8. El usuario introduce su fecha de nacimiento 9. El usuario elige su sexo 10. El usuario pulsa según su situación personal en las distintas opciones ofrecidas en la sección Condiciones de Interés 11. El usuario pulsa en “Crear usuario” 12. El sistema comprueba que el formulario tiene los campos necesarios 13. El sistema comprueba que las contraseñas introducidas coinciden y cumplen la longitud mínima de 8 caracteres 14. El sistema comprueba que no existe una cuenta asociada al correo electrónico introducido 15. El sistema almacena la información en la base de datos 16. El sistema informa de que la creación se ha completado con éxito 17. El sistema muestra la pantalla inicial de la aplicación
Escenarios alternativos
5.b0 El usuario saca una foto desde la cámara
5.c0 El usuario no elige una foto 5.c1 El sistema asigna al usuario la imagen por defecto
12.b0 El sistema comprueba que quedan campos obligatorios sin completar 12.b1 El sistema muestra un mensaje informando de los campos que deben ser completados
13.b0 El sistema comprueba que las contraseñas no coinciden 13.b1 El sistema muestra un mensaje informando de que las contraseñas no coinciden
13.c0 El sistema comprueba que las contraseñas no cumplen la longitud mínima 13.c1 El sistema muestra un mensaje informando de que las contraseñas no cumplen con la longitud mínima de 8 caracteres
14.b0 El sistema comprueba que existe una cuenta asociada al correo electrónico introduce 14.b1 El sistema muestra un mensaje informando de que el correo electrónico está en uso

Tabla 4.3.3: Caso de Uso "Registro en la Plataforma"

4.3.4 Cambio de Contraseña

Título	Cambio de contraseña
Descripción	El usuario accede a la información de su perfil, introduce su contraseña y el sistema lo actualiza en la base de datos

Pre-condición	El usuario ha abierto la aplicación y ha iniciado sesión
Post-condición	El usuario ha cambiado su contraseña
Prioridad	Media
Requisitos	RF-04 RNF-04
Escenario principal	
1. El usuario pulsa "Mi Perfil" 2. El sistema muestra la pantalla de perfil con la información del usuario 3. El usuario pulsa "Cambiar contraseña" 4. El sistema muestra la pantalla de cambio de contraseña 5. El usuario introduce su contraseña actual 6. El usuario introduce su nueva contraseña 7. El usuario vuelve a introducir su contraseña en el campo de confirmación de contraseña 8. El usuario pulsa "Cambiar contraseña" 9. El sistema comprueba que la contraseña actual es correcta 10. El sistema comprueba que las contraseñas coinciden 11. El sistema comprueba que la contraseña tiene la longitud mínima de 8 caracteres 12. El sistema actualiza la contraseña del usuario en la base de datos 13. El sistema muestra la pantalla de perfil	
Escenarios alternativos	
9.b0 El sistema comprueba que la contraseña actual no es correcta 9.b1 El sistema muestra un mensaje informando de que la contraseña introducida no es correcta	
10.b0 El sistema comprueba que las contraseñas no coinciden 10.b1 El sistema muestra un mensaje informando de que las contraseñas no coinciden	
11.b0 El sistema comprueba que la contraseña no tiene la longitud mínima 11.b1 El sistema muestra un mensaje informando de que las contraseñas deben tener una longitud mínima de 8 caracteres	

Tabla 4.3.4: Caso de Uso "Cambio de Contraseña"

4.3.5 Restaurar contraseña

Título	Restaurar contraseña
Descripción	El usuario ha olvidado su contraseña y solicita al sistema que le envíe un correo para restaurarlo
Pre-condición	El usuario ha abierto la aplicación, tiene cuenta registrada pero no ha iniciado sesión aún

Post-condición	El usuario restaura su contraseña
Prioridad	Alta
Requisitos	RF-05 RNF-04
Escenario principal	
1. El usuario pulsa en “Iniciar sesión” 2. El sistema muestra la pantalla de inicio de sesión 3. El usuario pulsa en “He olvidado mi contraseña” 4. El sistema muestra la pantalla de restablecimiento de contraseña 5. El usuario introduce su correo electrónico 6. El usuario pulsa en recuperar contraseña 7. El sistema muestra un mensaje informando de que si existe una cuenta asociada a ese correo recibirá un email para restaurar la contraseña 8. El sistema comprueba que existe una cuenta asociada a ese correo electrónico y envía un email a dicha dirección 9. El usuario accede a su cliente de correo y pulsa el enlace recibido 10. El sistema muestra la pantalla de restablecimiento de contraseña 11. El usuario introduce su nueva contraseña 12. El usuario pulsa en “Restablecer contraseña” 13. El sistema actualiza la contraseña en la base de datos	
Escenarios alternativos	
8.b0 El sistema comprueba que no existe una cuenta asociada al correo electrónico 8.b1 El sistema no hace nada	

Tabla 4.3.5: Caso de Uso "Restaurar Contraseña"

4.3.6 Actualizar Información Personal

Título	Actualizar información personal
Descripción	El usuario actualiza su información personal y el sistema actualiza dicha información en la base de datos
Pre-condición	El usuario ha abierto la aplicación y ha iniciado sesión
Post-condición	El usuario actualiza su información personal
Prioridad	Media
Requisitos	RF-06 RNF-04
Escenario principal	

<ol style="list-style-type: none"> 1. El usuario pulsa en “Mi Perfil” 2. El sistema muestra la pantalla de perfil con la información del usuario 3. El usuario pulsa en “Modificar Perfil” 4. El sistema muestra la pantalla de edición de perfil con la información del usuario 5. El usuario modifica los campos que desee cambiar 6. El usuario pulsa en “Actualizar Perfil” 7. El sistema comprueba que los cambios son correctos 8. El sistema actualiza la información en la base de datos 9. El sistema muestra la pantalla de perfil con la información del usuario actualizada
Escenarios alternativos
<p>7.b0 El sistema comprueba que algún campo no está correcto</p> <p>7.b1 El sistema muestra un mensaje informando del error y el campo a corregir</p>

Tabla 4.3.6: Caso de Uso "Actualizar Información Personal"

4.3.7 Borrar Perfil

Título	Borrar perfil
Descripción	El usuario accede a su perfil y lo elimina del sistema
Pre-condición	El usuario ha abierto la aplicación y ha iniciado sesión
Post-condición	El usuario elimina su perfil
Prioridad	Media
Requisitos	RF-07
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa en “Mi Perfil” 2. El sistema muestra la pantalla de perfil con la información del usuario 3. El usuario pulsa en “Eliminar Perfil” 4. El sistema muestra un dialogo solicitando al usuario que introduzca la contraseña para confirmar la eliminación 5. El usuario introduce la contraseña 6. El usuario pulsa en “OK” 7. El sistema comprueba que la contraseña es correcta 8. El sistema elimina la información del usuario, relativa a su perfil, de la base de datos 9. El sistema muestra la pantalla de inicio de la aplicación 	
Escenarios alternativos	
<p>6.b0 El usuario pulsa “Cancelar”</p> <p>6.b1 El sistema cierra el dialogo</p>	

7.b0 El sistema comprueba que la contraseña es incorrecta
 7.b1 El sistema informa de que la contraseña introducida es incorrecta

Tabla 4.3.7: Caso de Uso "Borrar Perfil"

4.3.8 Consulta de Delitos

Título	Consulta de delitos
Descripción	El usuario accede a la aplicación y puede consultar los delitos, ya sea en un listado o visualizado en un mapa, además de filtrar según sus intereses
Pre-condición	El usuario ha descargado la aplicación
Post-condición	El usuario consulta los delitos registrados
Prioridad	Alta
Requisitos	RF-08 RF-09 RF-10
Escenario principal	
1. El usuario accede a la aplicación 2. El sistema consulta los delitos que han sucedido ese mismo día 3. El sistema muestra un listado con los delitos obtenidos e información relevante al mismo 4. El usuario pulsa "Mapa" 5. El sistema muestra un mapa localizando los delitos 6. El usuario pulsa en un delito 7. El sistema muestra información sobre el delito 8. El usuario pulsa "Filtrar" 9. El sistema muestra la pantalla de filtros 10. El usuario modifica los campos de los filtros 11. El usuario pulsa "Filtrar" 12. El sistema consulta los delitos que se corresponden con los filtros introducidos 13. El sistema actualiza el listado/mapa con los nuevos delitos 14. El usuario hace pulsa en un delito 15. El sistema muestra una pantalla con toda la información accesible del delito, además de un mapa con su localización	
Escenarios alternativos	
2.b0 El sistema comprueba que no existen delitos ese día 2.b1 El sistema informa de que no existen delitos	
12.b0 El sistema comprueba que ningún delito se adapta a los filtros introducidos 12.b1 El sistema informa de que no existen delitos relativos a los filtros	

Tabla 4.3.8: Caso de Uso "Consulta de Delitos"

4.3.9 Consulta de Estadísticas Generales

Título	Consulta de estadísticas generales
Descripción	El usuario accede a la aplicación y puede consultar las estadísticas generales
Pre-condición	El usuario ha abierto la aplicación
Post-condición	El usuario consulta las estadísticas generales
Prioridad	Alta
Requisitos	RF-11
Escenario principal	
1. El usuario pulsa "Estadísticas" 2. El sistema muestra una pantalla con los tipos de estadísticas disponibles 3. El usuario pulsa en "Generales" 4. El sistema consulta las estadísticas generales almacenadas en la base de datos 5. El sistema muestra las estadísticas generales registradas en la aplicación	
Escenarios alternativos	
4.b0 El sistema comprueba que no hay estadísticas aún 4.b1 El sistema muestra un mensaje informando de que no hay estadísticas aún	

Tabla 4.3.8: Caso de Uso "Consulta de Estadísticas Generales"

4.3.9 Consulta de Estadísticas Específicas

Título	Consulta de estadísticas específicas
Descripción	El usuario accede a la aplicación y puede consultar las estadísticas específicas
Pre-condición	El usuario ha abierto la aplicación
Post-condición	El usuario consulta las estadísticas específicas
Prioridad	Media
Requisitos	RF-12 RF-13
Escenario principal	

1. El usuario pulsa "Estadísticas" 2. El sistema muestra una pantalla con los tipos de estadísticas disponibles 3. El usuario pulsa "Por Zona" 4. El sistema comprueba las estadísticas de una zona por defecto 5. El sistema muestra las estadísticas de la zona 6. El usuario pulsa "Elegir zona" 7. El usuario elige otra zona 8. El usuario pulsa "OK" 9. El sistema comprueba las estadísticas de la zona elegida 10. El sistema muestra las estadísticas
Escenarios alternativos
3.b El usuario pulsa "Por Tipo de Delito" 3.b1 El sistema comprueba las estadísticas de un tipo de delito por defecto 3.b2 El sistema muestra las estadísticas del tipo de delito 3.b3 El usuario pulsa "Elegir Tipo" 3.b4 El usuario elige otro tipo 3.b5 El usuario pulsa "OK" 3.b6 El sistema comprueba las estadísticas del tipo elegido 3.b7 El sistema muestra las estadísticas
4.b0 El sistema comprueba que no hay estadísticas aún 4.b1 El sistema muestra un mensaje informando de que no hay estadísticas aún
9.b0 El sistema comprueba que no hay estadísticas aún 9.b1 El sistema muestra un mensaje informando de que no hay estadísticas aún

Tabla 4.3.9: Caso de Uso "Consulta de Estadísticas Específicas"

4.3.10 Publicación de Delitos

Título	Publicación de delitos
Descripción	El usuario accede a la aplicación y publica un delito, aportando el tipo de delito, el momento en el que tuvo lugar y la ubicación, el sistema lo almacena y actualiza las estadísticas almacenadas
Pre-condición	El usuario ha abierto la aplicación y ha iniciado sesión
Post-condición	El usuario ha publicado un delito
Prioridad	Alta
Requisitos	RF-14 RNF-05
Escenario principal	

<ol style="list-style-type: none"> 1. El usuario pulsa "Publicar delito" 2. El sistema muestra la pantalla de publicación de delito 3. El usuario elige el tipo de delito del que se trata 4. El usuario indica que el delito ha tenido lugar ahora mismo 5. El usuario pulsa "Publicar delito" 6. El sistema comprueba que se ha indicado el tipo de delito, la hora y el lugar (si se ha marcado que ha tenido lugar ahora mismo el sistema comprueba la fecha, hora y ubicación) 7. El sistema comprueba la información sensible del usuario y los asocia al delito 8. El sistema comprueba la ubicación del delito y le asigna una zona 9. El sistema añade el delito en la base de datos 10. El sistema actualiza los ficheros de estadísticas con la información del nuevo delito 11. El sistema muestra la pantalla inicial de la aplicación
Escenarios alternativos
<ol style="list-style-type: none"> 4.b0 El usuario indica que no ha tenido lugar ahora mismo 4.b1 El usuario pulsa "Seleccionar fecha y hora" 4.b2 El sistema muestra un dialogo para seleccionar la fecha y hora 4.b3 El usuario selecciona la fecha y hora y pulsa "Aceptar" 4.b4 El sistema oculta el dialogo y almacena la fecha y la hora elegida, además de mostrarla 4.b5 El usuario pulsa una zona del mapa 4.b6 El sistema almacena la posición elegida 4.b7 Avanza al punto 5 del escenario principal
<ol style="list-style-type: none"> 6.b0 El sistema comprueba que falta información por indicar 6.b1 El sistema muestra un mensaje informando de la información que falta por indicar

Tabla 4.3.10: Caso de Uso "Publicación de Delitos"

4.3.11 Consulta de Delitos Publicados

Título	Consulta de delitos publicados
Descripción	El usuario accede a la aplicación y puede consultar los delitos que ha publicado, puede aplicar los mismos filtros disponibles para una consulta de delitos general
Pre-condición	El usuario ha abierto la aplicación y ha iniciado sesión
Post-condición	El usuario consulta sus delitos publicados
Prioridad	Media
Requisitos	RF-15 RNF-03
Escenario principal	

1. El usuario pulsa “Mis delitos publicados” 2. El sistema consulta los delitos publicados por el usuario 3. El sistema muestra una pantalla similar al caso “Consulta de Delitos”, con la misma funcionalidad
Escenarios alternativos
2.b0 El sistema comprueba que no existen delitos publicados por ese usuario 2.b1 El sistema muestra un mensaje informando de que aún no ha publicado ningún delito

Tabla 4.3.11: Caso de Uso "Consulta de Delitos Publicados"

4.3.12 Eliminación de Delitos

Título	Eliminación de delitos
Descripción	El usuario consulta sus delitos, se da cuenta de que ha cometido un error y decide eliminarlo, el sistema lo elimina y actualiza los ficheros de estadísticas
Pre-condición	El usuario ha abierto la aplicación, ha iniciado sesión y está consultando un delito que ha publicado
Post-condición	El usuario elimina el delito
Prioridad	Media
Requisitos	RF-16 RNF-05
Escenario principal	
1. El sistema muestra la pantalla de información del delito 2. El usuario pulsa “Eliminar delito” 3. El sistema muestra un dialogo informando de que está a punto de eliminar el delito 4. El usuario pulsa “Si” 5. El sistema elimina el delito de la base de datos 6. El sistema actualiza los ficheros de estadísticas teniendo en cuenta la eliminación del delito 7. El sistema muestra el listado de delitos publicados por el usuario	
Escenarios alternativos	
4.b0 El usuario pulsa “No” 4.b1 El sistema cierra el dialogo y vuelve a mostrar la pantalla de información del delito	

Tabla 4.3.12: Caso de Uso "Eliminación de Delitos"

4.3.13 Pulsación Botón del Pánico

Título	Pulsación botón del pánico
Descripción	Un usuario envía una señal de emergencia, el sistema la recibe, la comprueba y notifica al resto de usuarios con la información de esta
Pre-condición	Los usuarios han iniciado sesión en la aplicación
Post-condición	El usuario emisor consigue notificar de que está en peligro y el resto de los usuarios consigue ser notificado
Prioridad	Alta
Requisitos	RF-17 RF-18 RF-19 RNF-08
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario emisor pulsa el “Botón del Pánico” 2. El sistema consulta la ubicación del usuario emisor 3. El sistema consulta la información pública del usuario emisor 4. El sistema recibe la alerta 5. El sistema reenvía la alerta en forma de notificación push al resto de los usuarios 7. El sistema detecta que la aplicación está cerrada, lanza una notificación push informando al usuario receptor de que ha recibido una señal de emergencia 8. El usuario receptor pulsa la notificación push 9. El sistema abre la aplicación 10. El sistema muestra la pantalla de información de la alerta, donde puede ver la información pública de la víctima, el momento y la ubicación del lanzamiento de la alerta y el tiempo de llegada a pie y en vehículo a la ubicación de la víctima 	
Escenarios alternativos	
<p>7.b0 La aplicación está abierta</p> <p>7.b1 Avanza directamente al punto 10 del camino principal</p>	

Tabla 4.3.13: Caso de Uso "Pulsación Botón del Pánico"

5. Diseño y Arquitectura del Sistema

Una vez definidos los requisitos y el alcance del proyecto, en esta sección se va a presentar los resultados obtenidos de la fase de diseño que consiste, básicamente, los diagramas que representan el sistema y qué decisiones han sido necesario tomar.

5.1 Diagrama de Arquitectura

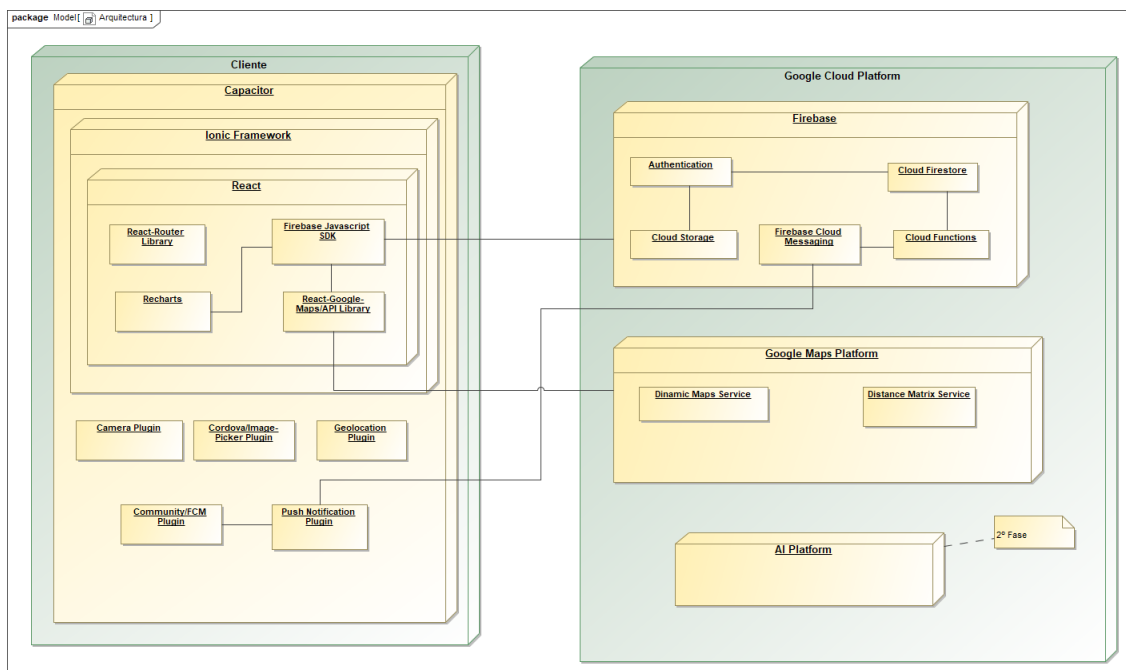


Figura 5.1.1: Diagrama de Arquitectura

En el diagrama de arquitectura presentado en la Figura 5.1.1 se exponen todos los componentes, ya sean servicios o bibliotecas, antes mencionados, además de las interconexiones entre sí, a muy alto nivel, ayudando así a entender cómo se compone tanto el cliente como el servidor, además de documentar de forma clara como se comunican entre sí.

Se puede ver que las principales conexiones son Firebase <-> Firebase SDK, Google Maps Platform <-> React Google Maps API Library y Push Notifications Plugin <-> Firebase Cloud Messaging, de esta forma se identifican inequívocamente estas conexiones. Además, podemos ver la composición de componentes dentro del propio sistema. también se indica que para desplegar el modelo recomendador de zonas, en la segunda fase, se hará uso de AI Platform (Google Developers, 2022), el servicio que ofrece Google para el entrenamiento y despliegue de modelos de aprendizaje automático. En un primer momento se planteó realizar el despliegue en AWS, pero se decidió mantenerse en el mismo ecosistema para evitar problemas de compatibilidad.

5.2 Diagramas de Clases

Una vez vista la arquitectura, se va a definir todas las clases y tipos de datos necesarias para el desarrollo del proyecto, esto permite visualizar de forma eficaz donde se encuentra la funcionalidad y la distribución de esta, además que permite entender mejor el sistema.

Para realizar un desarrollo de calidad, se ha llevado a cabo una división de funcionalidades, de forma que existe una capa, *service*, que ofrece toda la funcionalidad necesaria para la comunicación con el backend. Por otro lado, están los componentes, que son las distintas vistas de la interfaz que, como se ha mencionado anteriormente, se definen en TypeScript, de esta forma ofrecemos modularidad.

Además de estos componentes, se han definido una serie de interfaces necesarias para el tratamiento de los datos dentro de la aplicación, en primer lugar, se van a mostrar estos, después su relación con los *services* y, por último, con los componentes.

5.2.1 Tipos de Datos

Como se ha mencionado, primero se van a mostrar los tipos de datos que ha sido necesario generar para poder hacer un uso adecuado de la información en las distintas capas.

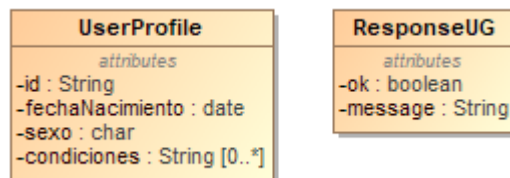


Figura 5.2.1: Tipos de Datos para la información del usuario y respuestas

Estas dos primeras clases, representadas en el Figura 5.2.1, se han generado para manejar la información del perfil del usuario, en el caso de **UserProfile** y la otra, **ResponseUG**, para los casos en los que la respuesta no contenía datos, simplemente hay que comprobar que algo ha salido bien, para ello se devuelve el valor lógico ok y un mensaje de error, en el caso de que sea necesario.

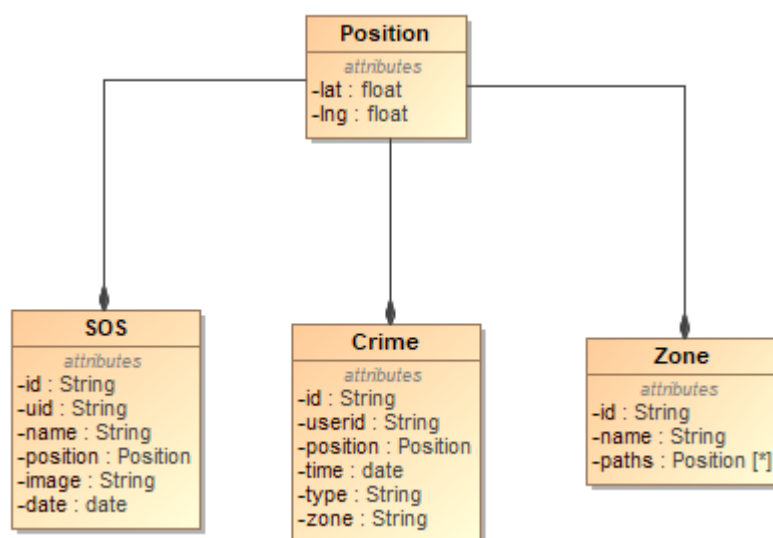


Figura 5.2.2: Tipos de Datos para Posición, Crímenes, Zonas y mensaje de SOS

Los 4 tipos de datos mostrados en la Figura 5.2.2 están relacionados entre sí ya que todos usan, al menos, una posición geográfica.

SOS es la información recogida sobre, valga la redundancia, un mensaje de SOS enviado por otro usuario, en el caso de **Crime** es igual y **Zone** se utiliza, para poder obtener las zonas

registradas en el sistema y, por ejemplo, ofrecerlas en el filtrado, incluso mostrarlas en un mapa.

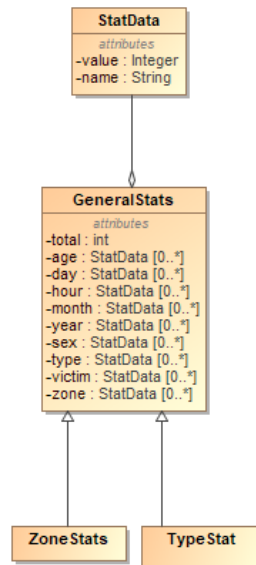


Figura 5.2.3: Tipos de Datos para Estadísticas

Por último, en la Figura 5.2.3 se muestran las clases necesarias para la gestión de los datos de las estadísticas, StatData es el tipo básico que corresponde con un dato de la estadística, por ejemplo, puede ser {"year": 2022, "value": 2}, indicando el nombre del dato y su valor.

Con la composición de este tipo básico de dato, se construye una estadística general que contiene información sobre el total de delitos registrados, la edad, el día de la semana, la franja horaria, el mes, el año, el sexo de la víctima, las condiciones de interés, el tipo de delito y la zona.

También existen estadísticas por zona y por tipo, estas consisten en la misma información que la general, pero contienen información sobre la zona y el tipo de delito respectivamente.

5.2.2 Capa Services

De la misma forma, se va a mostrar la funcionalidad ofrecida en la capa *services*.

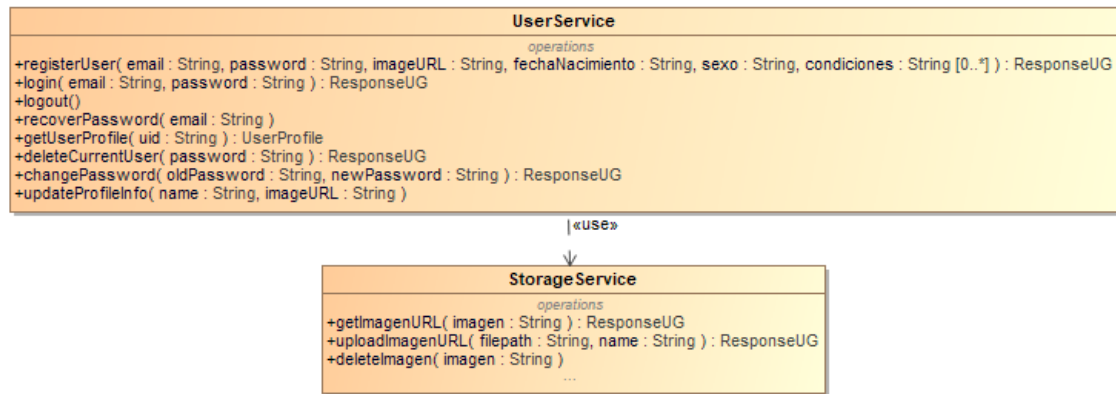


Figura 5.2.4: Diagrama de Clases de la capa service para la funcionalidad de gestión de usuarios

Como se observa en la Figura 5.2.4 la funcionalidad referente a la autenticación de usuarios hace uso de 2 clases, UserService, que ofrece funcionalidad para registrar usuarios, hacer operaciones de inicio de sesión y cierre, recuperación de contraseña, eliminación y actualización de perfil y cambio de contraseña.

Para acompañar a todo esto, se ofrece StorageService, que permite la subida, eliminación y obtención de objetos de Cloud Storage, en concreto, imágenes, lo que permite gestionar las imágenes de perfil.

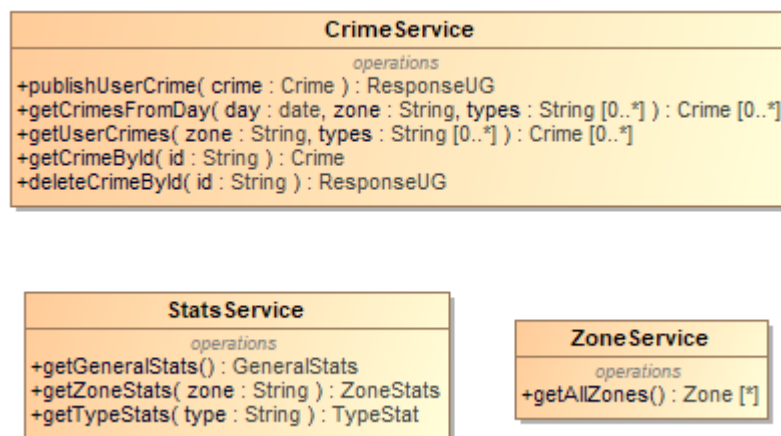


Figura 5.2.5: Diagrama de Clases de la capa service para la funcionalidad de publicación/gestión de delitos

Para el caso de la funcionalidad, tanto de publicar y consultar delitos, como para visualizar estadísticas, se han desarrollado estas clases (Figura 5.2.5), CrimeService ofrece toda la parte de publicar delitos, obtenerlos, filtrarlos y eliminarlos, mientras que StatsService y ZoneService permite obtener las distintas estadísticas y zonas respectivamente.

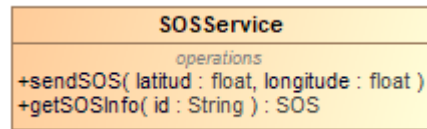


Figura 5.2.6: Clase de la capa services para ofrecer la funcionalidad de SOS

Por último, esta clase (Figura 5.2.6) permite generar una señal de SOS y, recuperar su información asociada.

5.2.3 Interfaces de Usuario

Las clases necesarias para generar las distintas vistas de la aplicación son las siguientes.

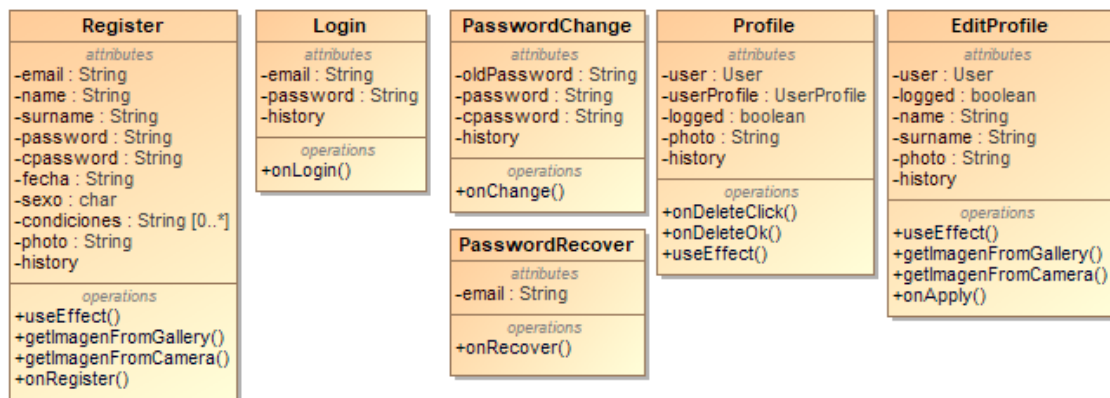


Figura 5.2.7: Diagrama de Clases para la interfaz de la funcionalidad de gestión de usuarios

En cuanto a la funcionalidad referente a la gestión de usuarios, como se puede observar en la Figura 5.2.7, se han desarrollado interfaces para el registro, inicio de sesión, cambio y recuperación de contraseña, visualización y edición de perfil.

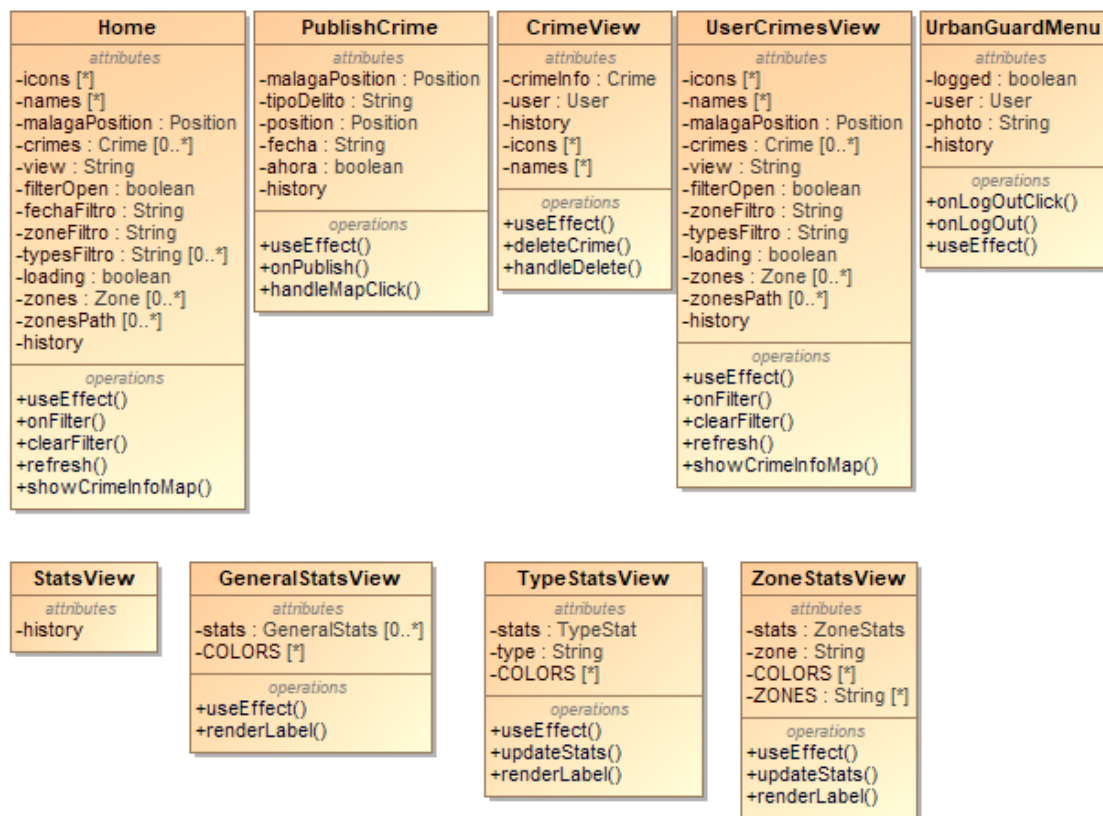


Figura 5.2.8: Diagrama de Clases para la interfaz de la funcionalidad de publicación/consulta de delitos

Para la publicación y consulta de delitos y visualización estadísticas, estas han sido las interfaces desarrolladas (Figura 5.2.8), por aclarar, StatsView es una pantalla de selección de tipo de estadísticas.

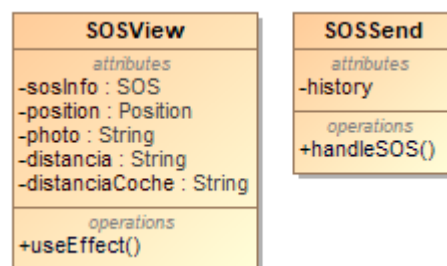


Figura 5.2.9: Diagrama de Clases para la interfaz de la funcionalidad de SOS

Por último, las interfaces para el envío y recepción de mensajes de emergencia se detallan en la Figura 5.2.9.

5.3 Diagramas de Secuencia

Para documentar el funcionamiento de los casos de uso más complejos, se han desarrollado una serie de diagramas de secuencia.

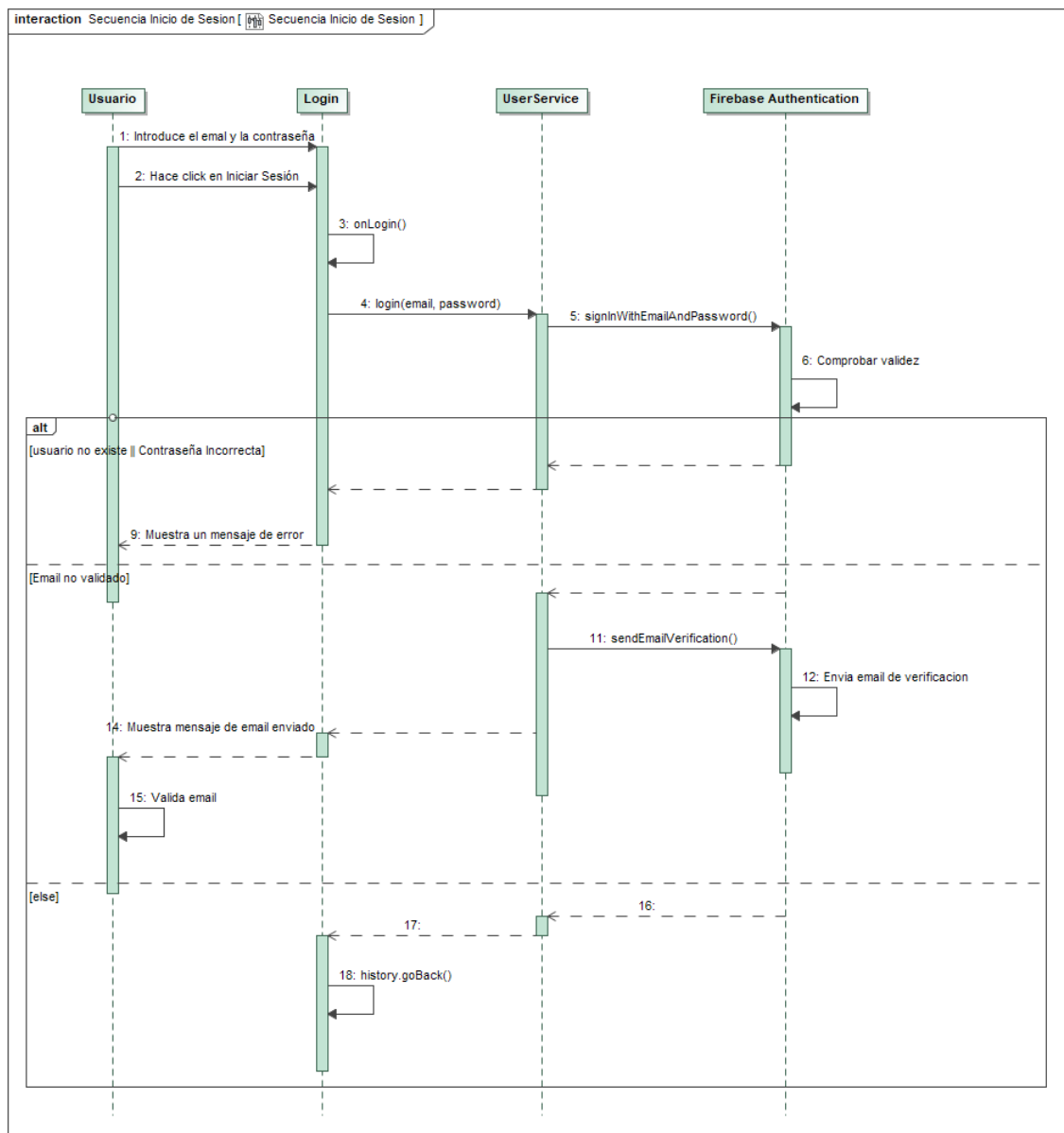


Figura 5.3.1: Diagrama de secuencia del Inicio de Sesión

Para el caso de inicio de sesión (Figura 5.3.1), se puede comprobar que se contemplan los casos de error y, en el caso de que el email del usuario no este verificado, se envía el correo electrónico de confirmación, cuando el usuario acceda a ese correo y pulse en el enlace, podrá iniciar sesión en la plataforma.

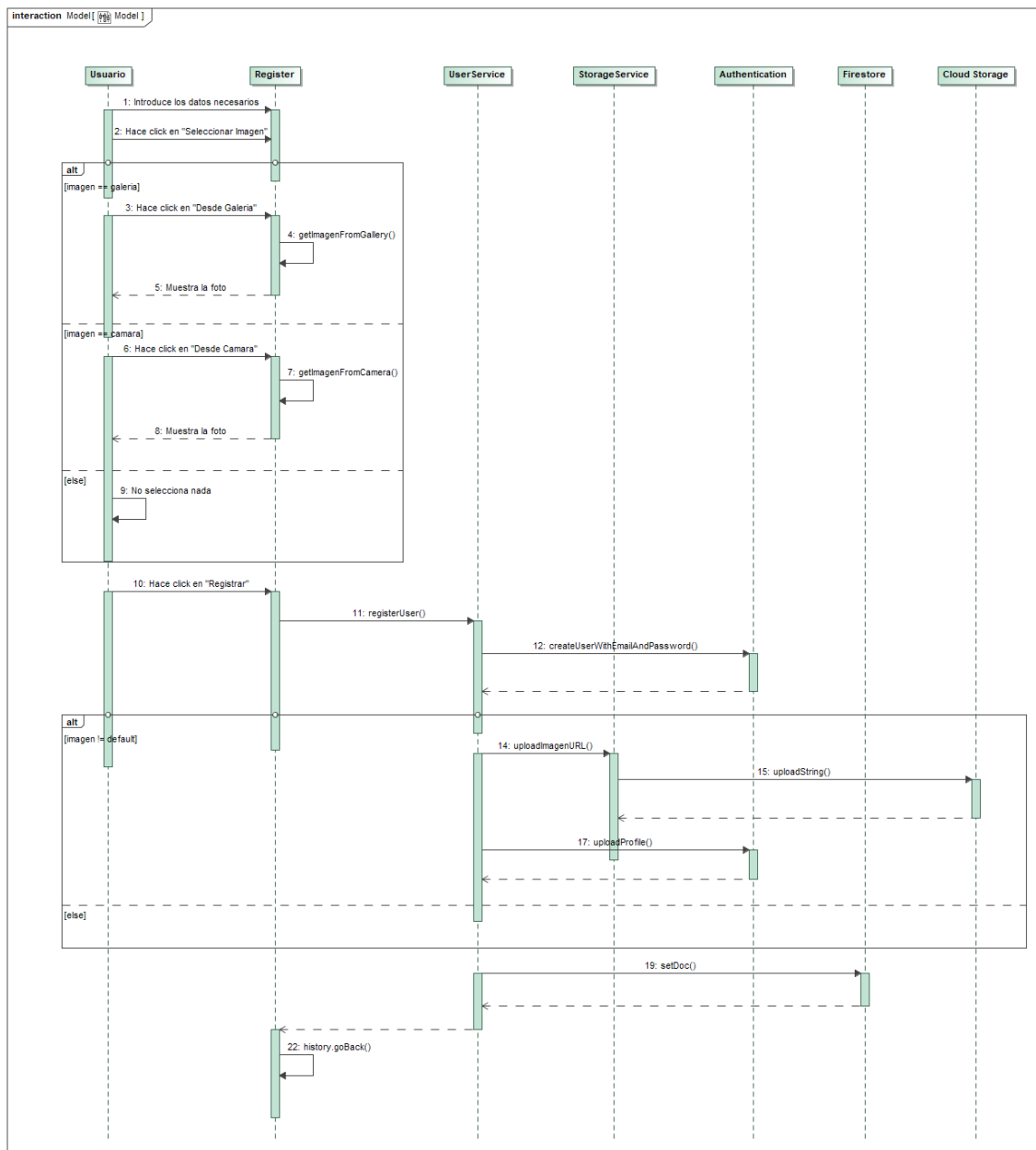


Figura 5.3.2: Diagrama de secuencia del registro de usuario

En el caso del registro, visible en la Figura 5.3.2, además de añadir información en forma de formulario, se da la opción de subir una imagen de perfil, ya que puede ser de utilidad, sobre todo, para ser mostrada en los mensajes de SOS, por ello, se permite la carga desde cámara o galería, pero, aunque no es recomendado, no es necesario seleccionar una imagen y dejar una por defecto. Este diagrama resulta similar al de actualización de perfil.

Esta imagen se almacena en Cloud Storage de forma que, si es necesario, pueda ser recuperada para mostrarse tanto a él, como a otro usuario.

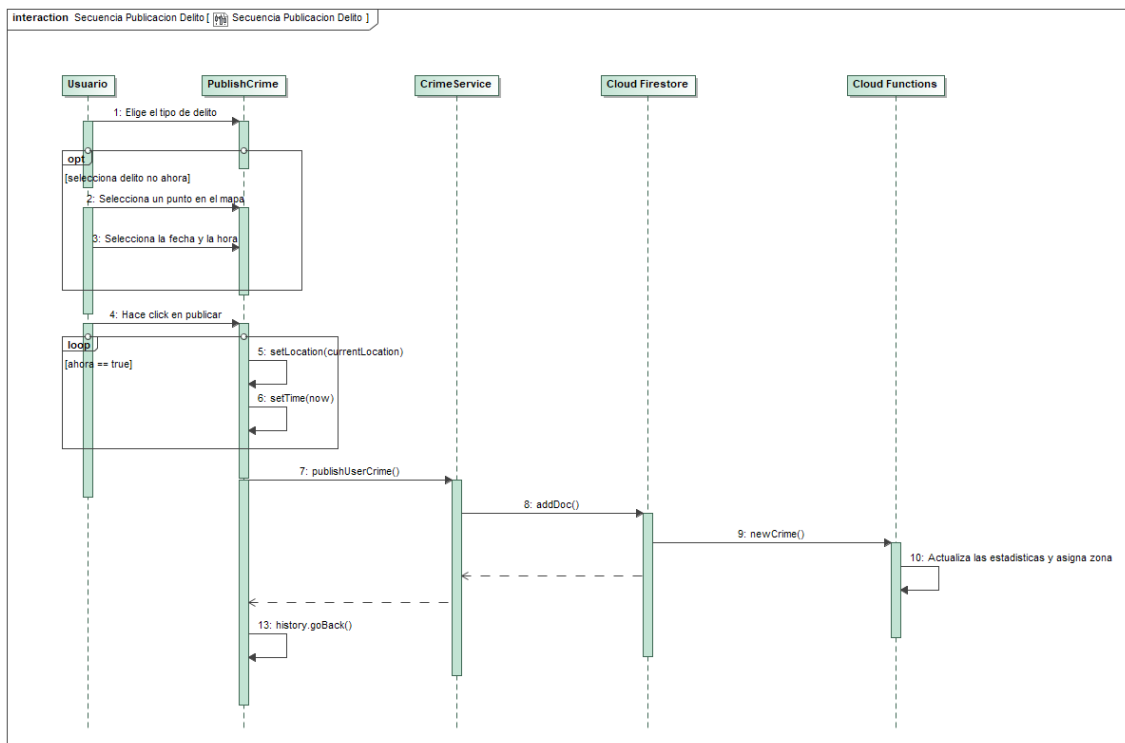


Figura 5.3.3: Diagrama de secuencia de publicación de delito

A la hora de publicar un delito, documentado en la Figura 5.3.3, es interesante ofrecer al usuario la posibilidad de reportar un delito que acabe de sufrir o, por evidentes motivos, no puede en ese momento, ofrecerle la posibilidad de hacerlo en otro momento.

Para lograrlo, se ofrecen dos modos de publicación, en el primero se recoge la localización, pidiendo permiso al usuario, y la hora del dispositivo, en el caso contrario, se ofrece un mapa para que el usuario pueda marcar el punto exacto y el día y la hora.

Para ofrecer las estadísticas, como se verá más adelante, es inviable calcular las estadísticas cada vez que un usuario las solicite, por lo que se generará un documento que, conforme se generen nuevos delitos, sea actualizado, por ello mismo, la zona debe ser asignada y los ficheros de estadísticas actualizados cada vez que un nuevo delito se publique, de la misma forma, deben ser actualizados cada vez que se elimine.

Esto se ha implementado mediante un trigger que lanza una ejecución de Cloud Functions, y la asignación de la zona, para evitar errores provocados por el usuario, se ha automatizado mediante un algoritmo, se darán más detalles de este más adelante.

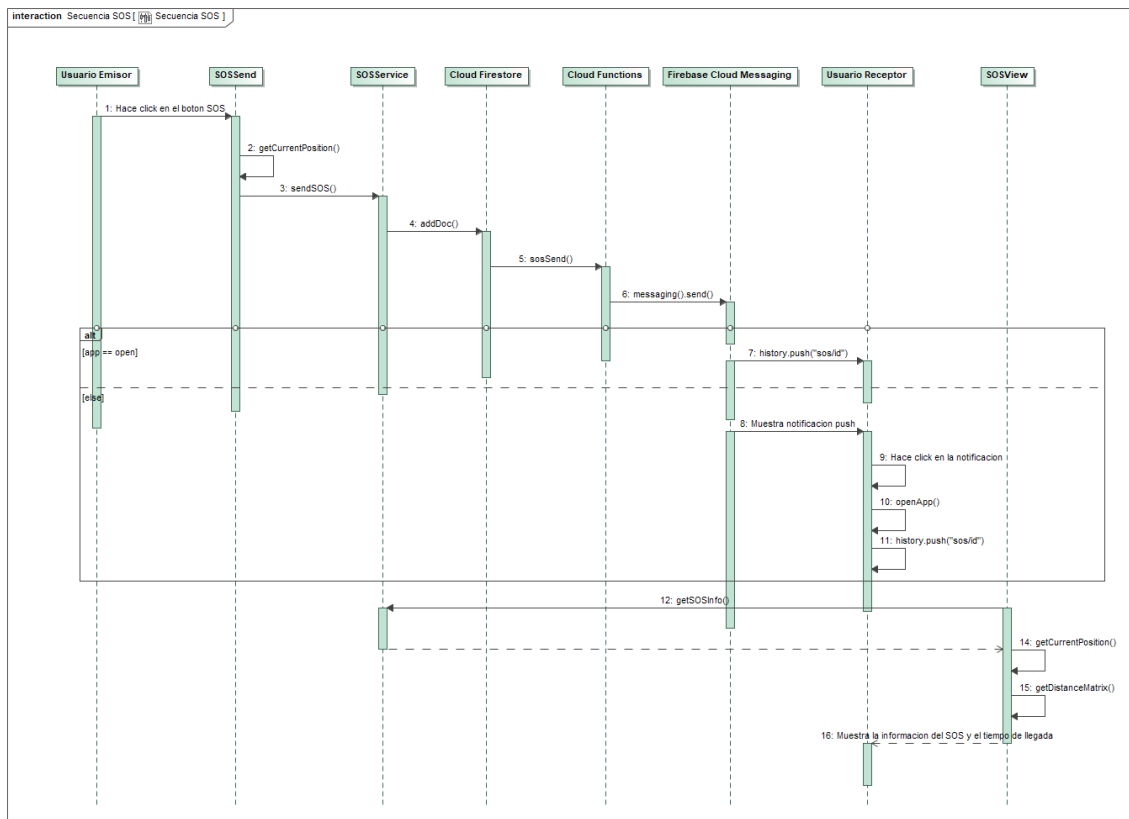


Figura 5.3.4: Diagrama de secuencia de pulsación del SOS

Para acabar, la Figura 5.3.4 muestra el caso del SOS, que consiste en la creación de un documento en Firestore, de forma que se dispare un trigger de Cloud Functions y este notifique a los demás usuarios.

Cuando la aplicación recibe la notificación, si se encuentra abierta, se mostrará inmediatamente la información, en el caso de que este cerrada o en segundo plano, se mostrará una notificación push que, al ser pulsada por el usuario, mostrará la misma información.

Además de mostrar la información básica del mensaje, también se realiza una petición al servicio Distance Matrix de Google Maps y se obtiene el tiempo de llegada del usuario a la señal, tanto a pie como en vehículo.

5.4 Diagrama de Navegación

Por último, se ha realizado un diagrama de navegación de la aplicación, en él se detalla cómo funciona la navegación del sistema y que transiciones será posible hacer.

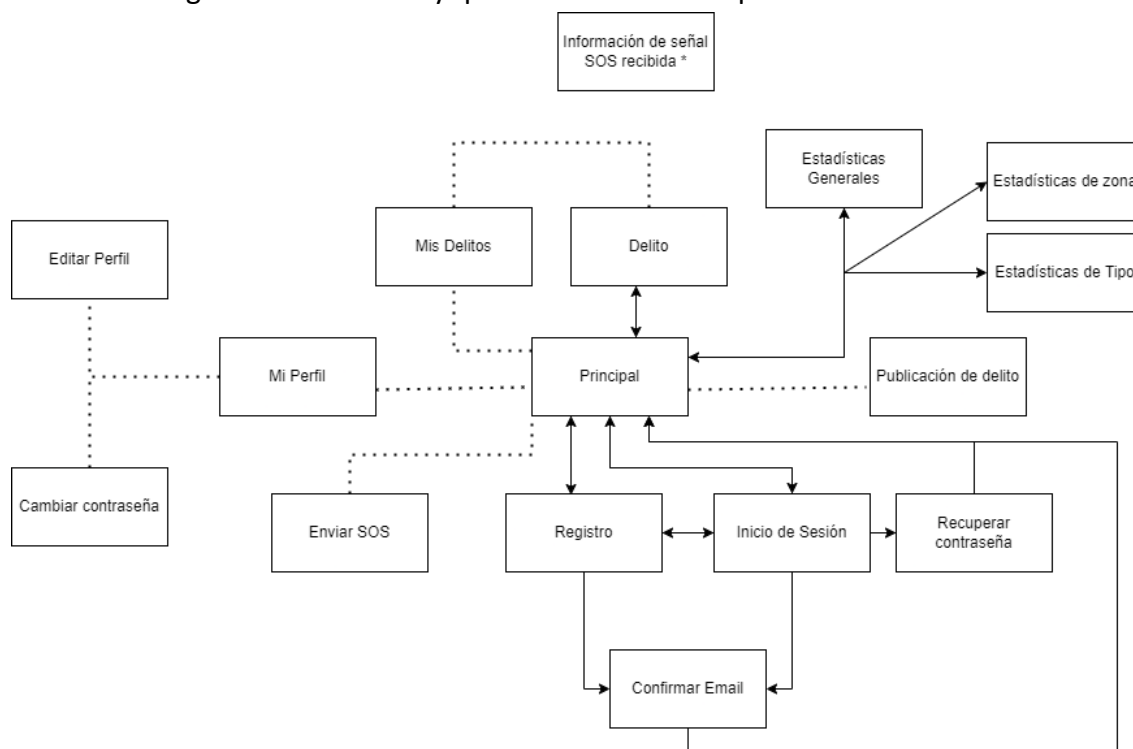


Figura 5.4.1: Diagrama de navegación del sistema

La navegación (Figura 5.4.1) está orientada a un menú principal, este tiene varias opciones, de las cuales se puede acceder a secciones, como, por ejemplo, perfil, estadísticas, delitos y enviar SOS.

Una vez se accede a una sección, el menú desaparece y, en su lugar, se pueda volver hacia atrás, de forma que se vuelva a la página original y, de esta, poder acceder a otra sección, de esta forma, no se generan navegaciones extrañas y permiten al usuario generar un modelo mental que facilite su adaptación a la aplicación.

Como es evidente, no toda la navegación es posible en todo momento, en concreto, el acceso a Mi Perfil, Mis Delitos y Enviar SOS solo estará disponible cuando se haya iniciado sesión, movimientos representados con línea discontinua. De la misma forma, Registro, Inicio de Sesión y Recuperar Contraseña solo están disponibles en el caso contrario. Esto se ha llevado a cabo de esta forma porque se ha estimado que puede ser interesante ofrecer a aquellos

usuarios que, por el motivo que sea, no deseen registrarse, algunos servicios mínimos como la visualización de los delitos y las estadísticas, además de que también recibirán los mensajes de SOS y pueden ayudar, lo que puede motivarles a crear su propio usuario.

Esta pantalla de visionado de información del SOS es accesible desde cualquier punto, ya que el hecho de recibirlo hará que se cambie a esa pantalla. De la misma forma, se puede volver a la pantalla anterior pulsando el botón de retroceso.

6. Desarrollo

En esta sección se van a abarcar los detalles que han supuesto una mayor complejidad o han resultado de mayor interés durante la fase de desarrollo del proyecto.

6.1 Colecciones en Firestore

En primer lugar, se va a abordar el modelo de datos utilizados para gestionar la información de la plataforma en Firestore, para lograrlo, se han definido cinco colecciones que se van a describir a continuación. Además de estas cinco colecciones, se almacena la información del usuario correspondiente, gestionado por Authentication, que mantiene el nombre del usuario, el correo, un booleano que indica si el email esta verificado y, si la imagen del usuario es una propia o por defecto, de forma que, si es una propia, dicha imagen se encuentra en Cloud Storage en la ruta `"/images/{id}.jpg"`.

6.1.1 Colección Profiles

Esta colección tiene como objetivo servir de complemento a la proporcionada automáticamente por Authentication, de forma que se pueda añadir información adicional de forma más simple y eficiente, además de poder ser protegida por reglas de seguridad.

Profiles	
PK	<u>ID</u>
	condiciones : String[] fechaNacimiento : Date sexo : String

Figura 6.1.1: Modelo de la colección Profiles

Esta colección (Figura 6.1.1), bastante simple, almacena en el campo condiciones una colección de strings que representa una serie de condiciones especiales que el usuario cumple y que son de interés para el muestreo de estadísticas, se hablará más adelante sobre ellas.

El resto simplemente son la fecha de nacimiento, para obtener información relevante a la víctima, y de la misma forma, el sexo, que se representa con una "M" para el caso masculino y una "F" para el caso femenino. Simplemente hay que destacar que el ID de este documento es el mismo que asigna automáticamente Authentication al registrar el usuario, para evitar ambigüedades.

6.1.2 Colección Crimes

Crimes	
PK	<u>ID</u>
FK	userid : String position : Map time : Date type : String zone : String

Figura 6.1.2: Modelo de la colección Crimes

La colección Crimes, representada en la Figura 6.1.2, es el corazón de la aplicación, contiene la información asociada a cada delito reportado por los usuarios.

En este caso, el ID es generado automáticamente por Firestore cuando el documento se crea, evitando redundancias. El resto de los campos, empezando por `userid`, es una referencia al perfil del usuario que, en este caso, es la víctima, se ha decidido adoptar esta decisión en lugar de incrustar directamente toda la información del perfil, como se haría en un modelo NoSQL por defecto, porque no se estima necesario, realmente solo se consulta esa información a la hora de actualizar las estadísticas, además que de esta forma, cada vez que se recupere un delito la información de la víctima sería obtenible, algo que no se considera adecuado, por ello simplemente se mantiene el id, recuperando el perfil cuando sea necesario, de esta forma, también es posible comprobar si un delito es del usuario simplemente consultando la coincidencia de sus ids, sin información personal.

Por otro lado, se almacena la posición y la zona, donde la posición es un mapa con dos campos, `lat` y `lng`, latitud y longitud del delito respectivamente y `zone` un string con el identificador

de esta. Puede parecer redundante el hecho de incluir una variable que es directamente deducible de otra, pero esto se ha planteado así para evitar problemas, ya que, si dejamos al usuario marcar la posición y elegir la zona de forma independiente, puede introducir errores de forma muy fácil, y calcular la zona para cada punto cada vez que se recupere es algo inviable, no tiene sentido abordar esta decisión. Para solventar esto se ha desarrollado un algoritmo que se ejecute y asigne la zona cuando un nuevo delito se publica. Este algoritmo será abordado más adelante.

Finalmente, se almacena el instante temporal en el que ha sucedido el delito, y el tipo de delito del que se trata.

6.1.3 Colección Zones

Zones	
PK	<u>ID</u>
	name : String paths : Map[]

Figura 6.1.3: Modelo de la colección Zones

Esta es una colección estática usada para almacenar las distintas zonas de Málaga, simplemente tiene dos campos, el nombre de la zona y una lista con las posiciones, ordenadas, que conforman la zona.

El hecho de que esté ordenada es importante, ya que, evidentemente, el algoritmo para comprobar que un punto está en una zona, necesita todas las aristas que conforman la zona.

Málaga se compone de 11 distritos municipales (Ayuntamiento de Málaga, 2011) que se pueden observar en la siguiente imagen.

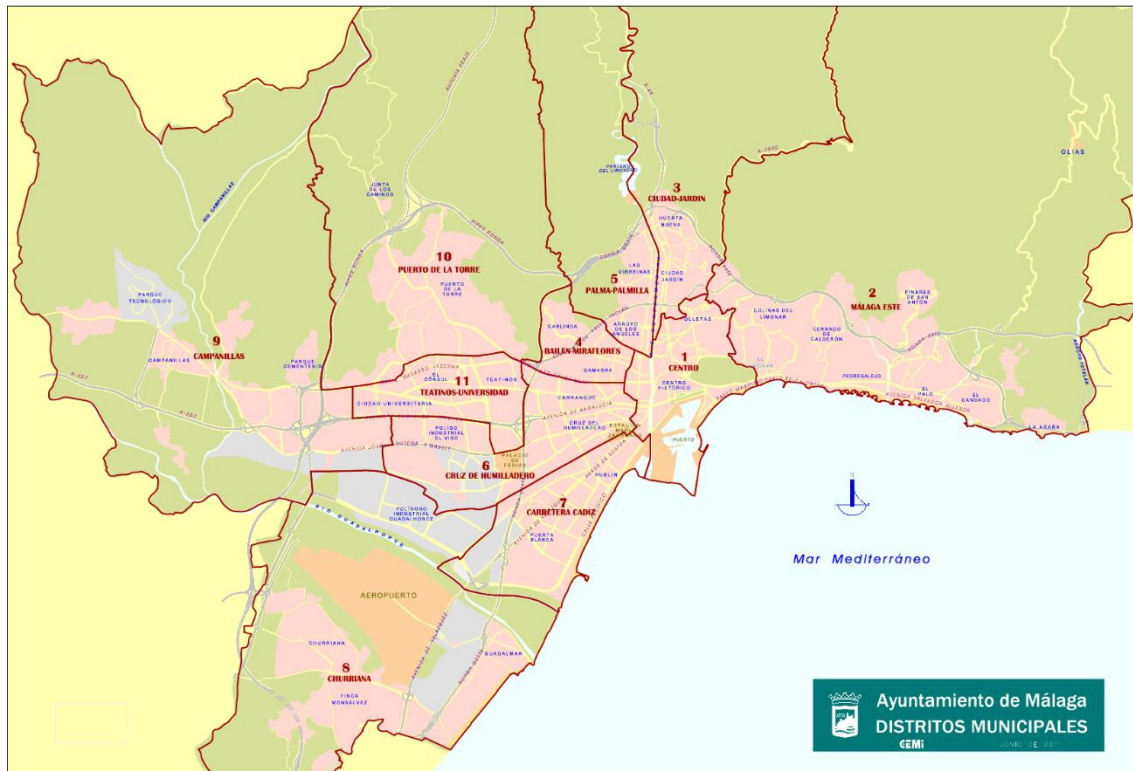


Figura 6.1.4: Mapa de distritos municipales de Málaga

Como se puede observar en la Figura 6.1.4, los límites de estos distritos son bastante irregulares, en un primer estudio, se planteó la posibilidad de simplificar estos polígonos y adaptarlos a cuadriláteros, de forma que, los puntos se generarían a mano y el algoritmo para comprobar a que zona pertenecen sería bastante más simple y eficiente.

Esto, a pesar de ser más barato computacionalmente hablando, se desestimó ya que, en casos como, por ejemplo, la intersección de Teatinos y Cruz de Humilladero, habría claros errores, además de en muchas zonas más.

Tras abandonar esta propuesta, se investigó en los datos abiertos ofrecidos por el ayuntamiento de Málaga y se logró encontrar un recurso donde se definen estos distritos y se pueden obtener sus límites (Ayuntamiento de Málaga, 2022).

Se obtuvieron los datos en formato csv y se les aplicó una simple transformación haciendo uso de Python y la biblioteca Pandas (Pandas Developers, 2022), de forma que se les dio el formato que se define en esta sección y se subieron a Firestore haciendo uso del SDK de Python.

6.1.4 Colección Stats

Stats	
PK	<u>ID</u>
	age : Map
	day : Map
	hour : Map
	month : Map
	year : Map
	sex : Map
?	type : Map
	victim : Map
?	zone : Map
	total : Number

Figura 6.1.5: Modelo de la colección Stats

Para el caso de las estadísticas, como se mencionó anteriormente, se tomó la decisión de mantener documentos que se irán actualizando conforme nuevos delitos sean publicados. Para ello se ha definido un modelo, visible en la Figura 6.1.5, orientado a facilitar esos cambios sin suponer una carga computacional elevada, este consiste en distintos campos, como pueden ser la edad, el día de la semana, el mes, etc. en los que se almacena un mapa en el que las claves son el identificador del campo y la clave es el valor, un número que se aumentará o se disminuirá en función de las características del modelo publicado, para facilitar la comprensión se muestra en la Figura 6.1.6 un ejemplo de cómo se almacenan la edad y el día de la semana, el formato es similar para el resto.

```

▼ age
  0-11: 0
  12-18: 0
  19-26: 3
  27-59: 0
  60-99: 0
▼ day
  0: 2
  1: 0
  2: 0
  3: 1
  4: 0
  5: 0
  6: 0

```

Figura 6.1.6: Ejemplo de los campos edad y día de las estadísticas

Como también se comentó, existían estadísticas generales, por zona y por tipo, pues también se aplica aquí; existe un documento general con todos los campos, un documento por cada zona, sin el campo `zone`, y uno por cada tipo de delito, de la misma forma, sin el campo `type`. De esta forma, cuando se genera un nuevo crimen, se actualiza el documento general, el documento de su zona y el de su tipo. De la misma forma que cuando se elimina un delito también se actualiza de forma oportuna.

Así se logra mantener las estadísticas actualizadas y modularizadas por zona y tipo, para ofrecer un mayor nivel de consulta al usuario.

6.1.5 Colección SOS

SOS	
PK	<u>ID</u>
	date : Date image : String lat : Number lng : Number name : String uid : String

Figura 6.1.7: Modelo de la colección SOS

Por último, como ya se ha mencionado, el envío de una señal SOS (entendido como el usuario pulsando el botón de pánico) tiene como origen la generación de un documento en Firestore. Este documento (Figura 6.1.7) almacena la fecha en la que se ha enviado, un campo imagen

para indicar si el usuario que ha enviado la señal tiene una imagen personalizada, y, por tanto, mostrarla, la posición geográfica desde donde se envió la señal, el nombre y el identificador del usuario.

El motivo de almacenar el identificador es poder distinguir el propio mensaje que envía el usuario y no se le notifique a sí mismo, esto se debe a la implementación basada en suscripción a temas que se ha utilizado.

Como se puede observar, se ha seguido al pie de la letra la recomendación de Firestore y se ha generado una serie de colecciones que constan de muchos documentos simples, de forma que la optimización podrá ser aprovechada, de la misma forma, aquellas colecciones que tienen documentos más grandes, como Stats o Zones, son estáticos y no crecen, de forma que no supondrán un problema de rendimiento.

6.1.6 Seguridad de las colecciones

Pero no es suficiente con almacenar los datos, también es necesario ofrecer una capa de seguridad mediante control de acceso para restringir y limitar el acceso y la modificación de los usuarios a las distintas colecciones.

Para ello, se han generado una serie de reglas de seguridad que detallan que requisitos se ha de cumplir para acceder a los distintos recursos de Firestore.

```
match /profiles/{userId} {  
  allow create, read, update, delete: if request.auth != null && request.auth.uid == userId;  
}
```

Figura 6.1.8: Regla de seguridad para la colección Profiles

La colección con más requisitos de seguridad es Profiles, esta almacena información personal del usuario, alguna incluso catalogada como categoría especial por el Reglamento General de Protección de Datos (RGPD) (Lenguaje Jurídico, 2021), es por ello lo que su acceso debe ser muy restringido, de forma que se ha generado una regla (Figura 6.1.8) que solo

permite acceder, modificar y eliminar esta información al propio usuario, siempre que este autenticado.

```
match /crimes/{crimeId} {  
  allow read : if true;  
  allow create : if request.auth != null && request.resource.data.userid == request.auth.uid;  
  allow delete : if request.auth != null && resource.data.userid == request.auth.uid;  
}
```

Figura 6.1.9: Regla de seguridad para la colección Crimes

En el caso de Crimes (Figura 6.1.9), la información no es tan delicada como el caso anterior, ya que no contiene información personal del usuario, tan solo su identificador. Con objeto de poder ofrecer una experiencia, aunque sea reducida, a todos los usuarios, incluso aquellos que no estén autenticados, el acceso a los crímenes es totalmente libre, cualquiera puede consultarlo sin estar autenticado, sin embargo, la creación y el borrado de delitos requieren ser realizados por la víctima, como es evidente.

```
match /zones/{zoneId} {  
  allow read : if true;  
}  
  
match /stats/{statId} {  
  allow read : if true;  
}
```

Figura 6.1.10: Reglas de seguridad para las colecciones Zones y Stats

La colección Zones contiene información pública que ha sido recogida de los datos ofrecidos por el Ayuntamiento de Málaga, por lo que su acceso no es ningún problema y para el caso de Stats, de la misma forma que en caso anterior, es una experiencia también disponible para usuarios, por lo que también se ofrece abiertamente, eso sí, ambas deben ser inmodificables de forma que, solo sea el propio sistema como administrador (desde Cloud Functions) capaz de realizar modificaciones, en este caso, sobre Stats (Figura 6.1.10).

```
match /sos/{sosId} {  
  allow read : if true;  
  allow create : if request.auth != null && request.resource.data.uid == request.auth.uid;  
}
```

Figura 6.1.11: Regla de seguridad para la colección SOS

Para terminar, la colección SOS (Figura 6.1.11), también se ofrece a usuarios sin autenticar, de forma que cualquiera pueda recibir señales de auxilio, pero su envío requiere autenticación y, que el usuario que ha enviado la petición es el realmente.

6.2 Tipos de Delito y Condiciones de la Víctima

A la hora de que un usuario reporte un delito, más allá de que se ha producido en una localización concreta, a cierta hora, durante un mes del año, se puede extraer bastante más información, por ejemplo, obteniendo que tipo de delito es el que ha sufrido, e incluso, haciendo uso del perfil de la víctima, obtener la edad de la misma y ciertas condiciones que pueden ser directamente causales de que haya sufrido ese delito, por ello se van a documentar tipos de delito y condiciones de la víctima que han sido seleccionados en primera instancia para ser objeto de estadística.

6.2.1 Tipos de Delito

Para ofrecer un listado con los distintos tipos de delito, se han consultado fuentes como EpData (EpData, 2022) y un mapa de criminalidad (Langdon Villamayor, 2020) que se generaba para realizar un estudio de criminalidad sobre él.

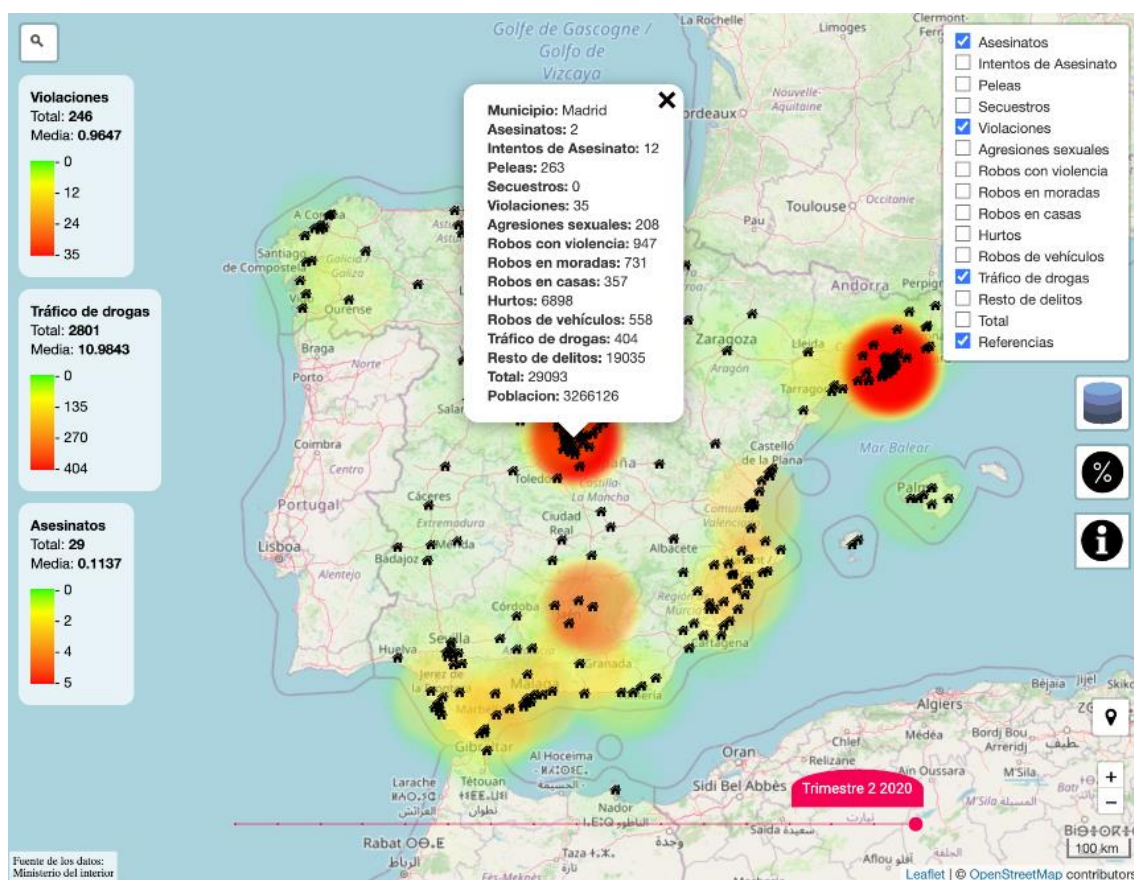


Figura 6.2.1: Mapa de Criminalidad de España

Estos datos fueron extraídos principalmente del Ministerio de Interior (Ministerio del Interior, 2022) y de él se extraen una tipología muy amplia de delitos. Para el caso de este proyecto, no tiene sentido contemplar algunos de ellos por la naturaleza de la aplicación, ya que esta orientada a la movilidad y, sobre todo, a la víctima, por tanto, no tiene sentido tener en cuenta, por ejemplo, robos de vehículos.

Para facilitar el visionado de los delitos en el mapa, se recuperaron una serie de iconos de una biblioteca digital (SVG Viewer, 2022) y se intentó adecuar al máximo su significado. El resultado de la tipología inicial y sus respectivos iconos es el siguiente.

- Hurto: consiste en la sustracción de bienes ajenos sin mediación de la fuerza ni uso de la intimidación (Abbo, 2022).



Figura 6.2.2: Icono de Hurto

- Robo con violencia: se basa en el mismo principio que el hurto, pero en este caso, la persona actúa violentamente para obtener los bienes ajenos (Abbo, 2022).



Figura 6.2.3: Icono de Robo con Violencia

- Pelea: este tipo de delito se basa en el concepto jurídico de riña tumultuaria, definida como la situación en la que un grupo de personas se riñen entre sí, acometiéndose tumultuariamente, haciendo uso de medios o herramientas que pongan en peligro la vida o la integridad de las personas (Wolters Kluwer, 2022). El nombre se ha modificado a pelea para facilitar su entendimiento y está pensado, principalmente, para que lo reporten aquellos ciudadanos que, sin quererlo y por terceros, se vean involucrados en dicha riña.

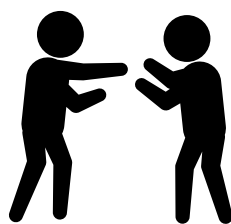


Figura 6.2.4: Icono de Pelea

- **Intento de Asesinato:** de forma similar al anterior, este se basa en el concepto de homicidio en grado de tentativa, definido como todo aquel delito en el que el autor intenta matar a una persona, pero no lo logra por causas ajenas a su voluntad (Redacción Law&Trends, 2022). No se ha incluido el homicidio como delito reportable por causas obvias, pero se estudiará una ampliación de la funcionalidad con objeto de ofrecerlo.

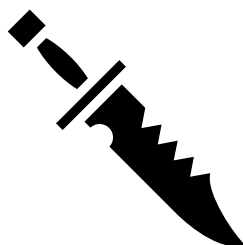


Figura 6.2.5: Icono de Intento de Asesinato

- **Agresión Sexual:** este delito es uno de los más importantes para tener en cuenta para el estudio, se defino como la atentación contra la libertad sexual de otra persona, con violencia o intimidación, sin la producción de acceso carnal y penetración (Vidal Rodríguez, 2022). Para simplificar la tipología de delitos, este caso también contempla aquellos en los que no exista violencia o intimidación.



Figura 6.2.6: Icono de Agresión Sexual

- Violación: este se basa en el anterior, pero, en este caso, si se produce acceso carnal y penetración (Wolters Kluwer, 2022).



Figura 6.2.7: Icono de Violación

- Otros: esta opción se da para el caso en el que el delito sufrido no se adecue a ninguno de los anteriores definidos.



Figura 6.2.8: Icono de Otros

Por último, simplemente hay que aclarar que esta tipología se puede modificar, de forma que resulta sencillo añadir nuevos tipos de delitos y no supone una gran labor de implementación, tan solo añadir las opciones en un subconjunto de los formularios. Quizás se pueda mejorar la implementación añadiendo la tipología en un fichero de configuración o en Firestore, pero como primera instancia se ha decidido optar por esta, se estudiará la posible modificación en siguientes versiones.

6.2.2 Condiciones de Interés

Ya vistos los tipos de delitos, se van a abordar las condiciones de interés de la víctima, más allá de la edad y su sexo, que son cosas evidentes, se han obtenido una serie de condiciones que puedan ser causa de un delito de odio hacia un subconjunto de la población.

De la misma forma que para la tipología de delitos, se van a exponer la serie de condiciones extraídas a raíz del estudio del temario de la asignatura de Victimología del Grado en Criminología de la Universidad de Málaga (Gómez Jiménez, 2019), acompañadas del icono que será mostrado en su perfil personal.

- Comunidad LGBTQ+: el año pasado, tras el triste crimen homóforo que se llevaba la vida de Samuel, el cual fue asesinado al grito de "maricón", se llevaba a cabo un estudio sobre los crímenes de odio hacia el colectivo y se llegó a la conclusión de que el número de delitos, tras una gran caída en 2015, volvía a tener una tendencia alcista (Zuil & Ojeda, 2021), visible en la Figura 6.2.9, es por ello por lo que se ha decidido incluirlo como condición de interés.

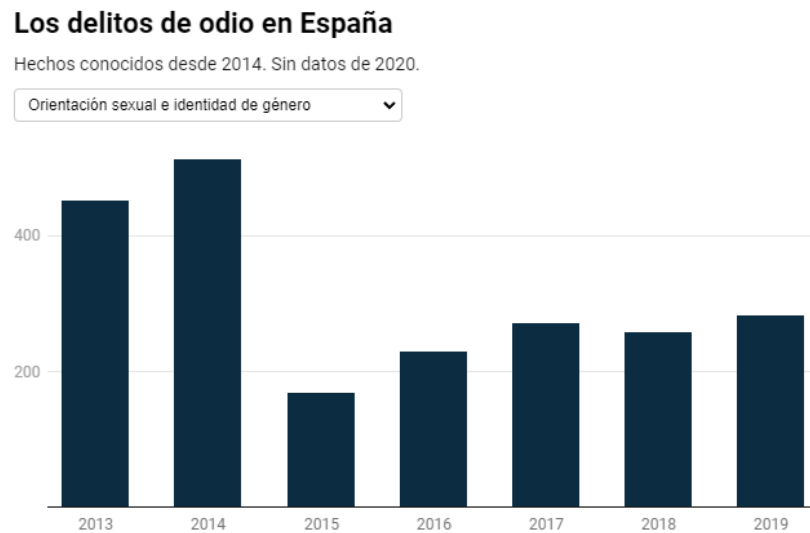


Figura 6.2.9: Delitos de odio en España por orientación sexual e identidad de género desde 2013 hasta 2019



Figura 6.2.10: Icono de Comunidad LGBTQ+

- Discapacidad: de la misma forma que los anteriores, los delitos de odio hacia personas con discapacidad han sufrido un repunte, en este caso del 69,2% en variación al año anterior (López-Fonseca, 2021), como se ve en la Figura 6.2.11, por lo que resulta también interesante incluirlo.

Delitos de odio en España

Hechos conocidos registrados

	2019	2020	% Variación
Antisemitismo	5	3	-40
Aporofobia	12	10	-16,7
Creencias o prácticas religiosas	66	45	-31,8
Persona con discapacidad	26	44	69,2
Orientación sexual e identidad de género	278	277	-0,4
Racismo/xenofobia	515	485	-5,8
Ideología	596	326	-45,3
Discriminación por sexo/género	69	99	43,5
Discriminación generacional	9	10	11,1
Discriminación por enfermedad	8	13	62,5
Antigitanismo	14	22	57,1
Total delitos	1.598	1.334	-16,5
Resto de incidentes	108	67	-38,0
Total delitos e incidentes de odio *	1.706	1.401	-17,9

(*) Todos los delitos han bajado durante la pandemia

Figura 6.2.11: Delitos de odio en España en 2019 y 2020 y su variación



Figura 6.2.12: Icono de discapacidad

- Etnia Minoritaria: es evidente que los casos de delitos de odio por diferencias culturales y sociales son un hecho real, por lo que también se ha incluido.

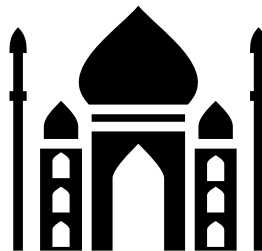


Figura 6.2.13: Icono de Etnia Minoritaria

- Color de piel: aún más evidente que el anterior, las diferencias, ya no sociales ni culturales, si no a nivel de color de piel siguen siendo, tristemente, un problema del día a día para estas personas, es por ello por lo que se añade.

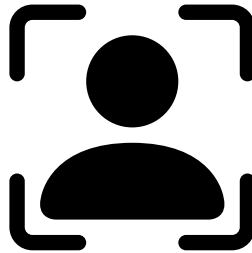


Figura 6.2.14: Icono de Color de Piel

- Empleo de Riesgo: por último, el simple hecho de ejercer un empleo concreto es, en algunas ocasiones, el motivo principal de sufrir un delito, algunos de estos empleos pueden ser el de taxista, fuerzas del orden, juez, incluso profesor, además de un amplio abanico de posibilidades.

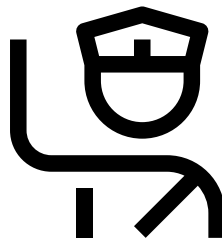


Figura 6.2.15: Icono de Empleo de Riesgo

De la misma forma que pasaba con los tipos de delitos, el proceso de añadir nuevas condiciones es sencillo y no supone una labor compleja de implementación.

6.3 Asignación de Zonas

Como se ha mencionado anteriormente, se tomó la decisión de realizar una asignación de la zona del delito, una vez el mismo sea publicado, para evitar que el usuario pueda introducir errores, ya sea inconscientemente o con malas intenciones.

Estas zonas se obtenían de la fuente de datos abiertos ofrecidos por el Ayuntamiento de Málaga y consisten en una serie de puntos geográficos ordenados que forman el perímetro de la zona.

El resultado de recoger las zonas de esta forma son polígonos complejos e irregulares, por lo que no resulta posible aplicar sobre ellos ningún algoritmo sencillo para comprobar si un punto está en su interior.

Por ello, se inició un proceso de investigación para encontrar un algoritmo geométrico capaz de resolver el problema denominado Point-In-Polygon (PIP) (Wikipedia, 2022), un problema de geometría computacional que consiste en determinar, dado un punto, un plano y un polígono, si el punto está dentro o fuera de los límites del polígono, justamente lo necesario para resolver este problema.

El resultado de esta investigación tuvo como resultado que, para esta casuística, aplicar un algoritmo basado en Ray Casting (Emisión de Rayos) era suficiente para comprobar si un punto está dentro de un polígono.

Este tipo de algoritmo es uno de los más simples a la hora de resolver este problema y, por tanto, de los más eficientes. Se basa en una observación bastante simple, si se traza un rayo desde el punto a comprobar, hasta el infinito, es posible contar cuantas veces atraviesa ese rayo las aristas del polígono.

Si bien es posible contar ese número de coincidencias, se puede determinar que un punto está en el interior del polígono haciendo uso de la regla par-impar (Wikipedia, 2020), es decir, si su rayo lo atraviesa un número impar de veces, estará contenido, en el caso contrario, si lo atraviesa un número par de veces, no estará contenido, se puede observar de forma sencilla en la Figura 6.3.1.

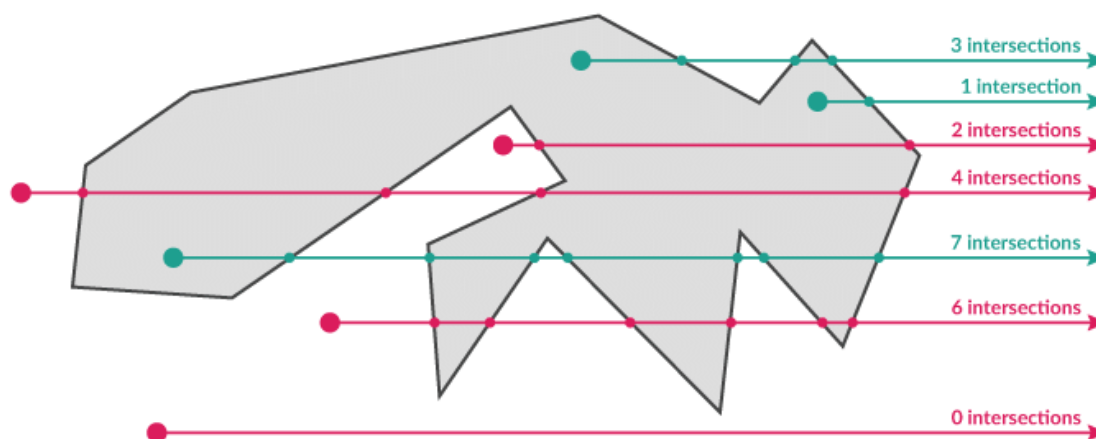


Figura 6.3.1: Ejemplo de Ray Casting para determinar si un punto está dentro de un polígono (Vacek, Atter, Rizo, & Nam, 2017)

Pero no todo son ventajas, este algoritmo se encuentra bastante limitado debido a que, computacionalmente, es imposible la existencia de valores infinitamente precisos, por ello, en los casos en los que un punto está cerca de los límites pueden producirse errores debido al redondeo.

Para este caso de estudio no va a suponer un problema, ya que si al estar muy cercano en un borde, en lugar de asignarse a una zona, se asigna a otra, realmente no supone un error destacable, además de que el propio sistema de geolocalización del dispositivo móvil introduce errores, por lo que resultan admisibles estos errores en el caso de que sucedan.

Una vez vista la viabilidad del algoritmo, se adoptó una solución encontrada (GeeksforGeeks, 2022) que se va a documentar a continuación, mediante pseudocódigo.

Algorithm 1 Algoritmo PIP basado en Ray Casting

```

1: function ISINSIDE(polygon, n, p)    ▷ Devuelve un booleano indicando si el
   punto p esta dentro del poligono polygon
2:   if  $n \leq 3$  then
3:     return false                      ▷ Se necesitan al menos 3 puntos
4:   end if
5:   extreme  $\leftarrow$  Point( $\infty$ , p.y)
6:   count  $\leftarrow$  0
7:   i  $\leftarrow$  0
8:   do
9:     next  $\leftarrow$  (i + 1)%n
10:    if doIntersect(polygon[i], polygon[next], p, extreme) then
11:      if orientation(polygon[i], p, polygon[next]) = 0 then
12:        return onSegment(polygon[i], p, polygon[next])
13:      end if
14:      count  $\leftarrow$  count + 1
15:    end if
16:    i  $\leftarrow$  next
17:  while i  $\neq$  0
18:  return count%2 = 1
19: end function

```

Figura 6.3.2: Pseudocódigo del Algoritmo PIP basado en Ray Casting

Como se puede observar en la Figura 6.3.2, es un procedimiento sencillo, se genera un punto en el infinito (a la hora de desarrollar será un número lo suficientemente grande) y se comprueba cuantas veces atraviesa las aristas del polígono. Existe una condición especial, cuando el punto pertenece a la arista, en este caso, también es verdadero.

Pero como también se puede observar, esta función se compone de otras tres, estas serán documentadas de la misma forma en la Figura 6.3.3 y, algunas de ellas, explicadas, ya que introducen conceptos nuevos como la orientación de 3 puntos.

Algorithm 2 Funciones necesarias

```
1: function ONSEGMENT(p,q,r)      ▷ Devuelve true si el punto q esta en el
   segmento 'pr'
2:   if  $q.x \leq \text{Max}(p.x, r.x) \ \& \ q.x \geq \text{Min}(p.x, r.x) \ \& \ q.y \leq$ 
    $\text{Max}(p.y, r.y) \ \& \ q.y \geq \text{Min}(p.y, r.y)$  then
3:     return true
4:   end if
5:   return false
6: end function
7: function ORIENTATION(p,q,r)    ▷ Devuelve la orientacion de
   los 3 puntos recibidos, 0 si son colineales, 1 sentido de las agujas del reloj,
   2 sentido contrario a las agujas del reloj
8:    $val \leftarrow (q.y - p.y) * (r.x - q.x) - (q.x - p.x) * (r.y - q.y)$ 
9:   if  $val = 0$  then
10:     return 0
11:   end if
12:   return  $(val > 0) ? 1 : 2$ 
13: end function
14: function DOINTERSECT(p1, q1, p2, q2) ▷ Devuelve true si las rectas 'p1q1'
   y 'p2q2' tiene intersección, false en el caso contrario
15:    $o1 \leftarrow \text{orientation}(p1, q1, p2)$ 
16:    $o2 \leftarrow \text{orientation}(p1, q1, q2)$ 
17:    $o3 \leftarrow \text{orientation}(p2, q2, p1)$ 
18:    $o4 \leftarrow \text{orientation}(p2, q2, q1)$ 
19:   if  $o1 \neq o2 \ \& \ o3 \neq o4$  then
20:     return true                                     ▷ Caso General
21:   end if
22:   if  $o1 = 0 \ \& \ \text{onSegment}(p1, p2, q1)$  then
23:     return true ▷ p1, q1 y p2 colineales y p2 esta en el segmento 'p1q1'
24:   end if
25:   if  $o2 = 0 \ \& \ \text{onSegment}(p1, q2, q1)$  then
26:     return true ▷ p1, q1 y q2 colineales y q2 esta en el segmento 'p1q1'
27:   end if
28:   if  $o3 = 0 \ \& \ \text{onSegment}(p2, p1, q2)$  then
29:     return true ▷ p2, q2 y p1 colineales y p1 esta en el segmento 'p2q2'
30:   end if
31:   if  $o4 = 0 \ \& \ \text{onSegment}(p2, q1, q2)$  then
32:     return true ▷ p2, q2 y q1 colineales y q1 esta en el segmento 'p2q2'
33:   end if
34:   return false                                     ▷ No se cumple ningún caso anterior
35: end function
```

Figura 6.3.3: Funciones necesarias para el algoritmo

En primer lugar, la función para comprobar si un punto pertenece a un segmento resulta trivial, simplemente se comprueba que sus valores x e y se sitúan entre los dos límites del segmento.

En cuanto a la función para obtener la orientación de una tripleta de puntos, estos se pueden orientar de 3 formas, colineales, en sentido de las agujas del reloj y en el contrario, como se puede observar en la Figura 6.3.4.

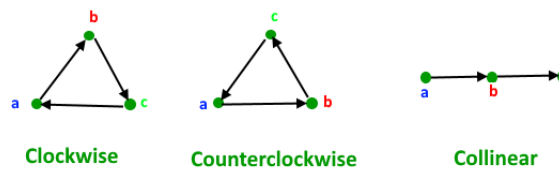


Figura 6.3.4: Posible orientación de una tripleta de puntos (GeeksforGeeks, 2022)

Esto es de utilidad para lograr obtener si dos segmentos tienen intersección, justo la siguiente función. Entonces, dos segmentos $(p1, q1)$ y $(p2, q2)$ tienen intersección si y solo si la orientación de $(p1, q1, p2)$ y $(p1, q1, q2)$ son diferentes y, además, la orientación de $(p2, q2, p1)$ y $(p2, q2, q1)$ también es diferente (GeeksforGeeks, 2022), algunos ejemplos se detallan en la Figura 6.3.5.

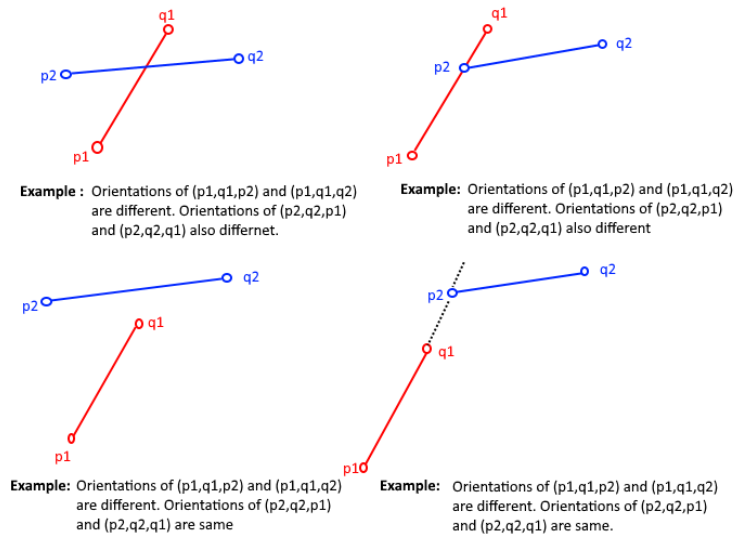


Figura 6.3.5: Ejemplo del caso general de intersección (GeeksforGeeks, 2022)

Además de este caso, existen 4 casos especiales que se detallan a continuación:

- $(p1, q1, p2)$ son colineales y $p2$ pertenece al segmento $(p1, q1)$
- $(p1, q1, q2)$ son colineales y $q2$ pertenece al segmento $(p1, q1)$
- $(p2, q2, p1)$ son colineales y $p1$ pertenece al segmento $(p2, q2)$
- $(p2, q2, q1)$ son colineales y $q1$ pertenece al segmento $(p2, q2)$

Con estos casos, es posible definir en que caso dos segmentos tienen intersección y, por tanto, permiten definir el algoritmo PIP.

El algoritmo se desplegó haciendo uso de Cloud Functions, aprovechando la función que actualiza las estadísticas de forma que, antes de realizar la actualización, se itera a través de todas las zonas almacenadas en Firestore y, si se encuentra una que está en su interior, se le asigna y se detiene la iteración.

Si no se encuentra ninguna zona, se le asigna el valor de zona "OTRA" y no se tiene en cuenta para las estadísticas, de esta forma no se tiene en cuenta información publicada que no pertenezca a las zonas especificadas.

En cuanto a la eficiencia, el algoritmo tiene complejidad $O(n)$ siendo n el número de vértices, por lo que se considera asequible. En una ejecución normal, en la que se asigna la zona y se actualiza los documentos de estadísticas, se consumen entre 3 y 6 segundos, como se puede ver en la Figura 6.3.6.

8:39:03.060 p. ...	newCrime	Function execution started
8:39:09.829 p. ...	newCrime	Function execution took 6770 ms, finished with status: 'ok'
8 jun 2022		
4:53:06.380 p. ...	newCrime	Function execution started
4:53:10.470 p. ...	newCrime	Function execution took 4091 ms, finished with status: 'ok'
14 jun 2022		
7:43:02.973 p. ...	newCrime	Function execution started
7:43:08.376 p. ...	newCrime	Function execution took 5402 ms. Finished with status: ok
7:44:04.749 p. ...	newCrime	Function execution started
7:44:08.299 p. ...	newCrime	Function execution took 3550 ms. Finished with status: ok
15 jun 2022		
9:19:07.676 a. ...	newCrime	Function execution started
9:19:11.594 a. ...	newCrime	Function execution took 3919 ms, finished with status: 'ok'

Figura 6.3.6: Ejecuciones de la función de asignación de zona y actualización de estadísticas

Realmente es un buen resultado, ya que se están realizando una serie de peticiones que son las que realmente consumen el tiempo, se recuperan las zonas, el perfil del usuario, y se actualizan 4 documentos, el del crimen para asignar la zona, y los 3 de las estadísticas (general, zona y tipo), además de la computación del algoritmo PIP, por lo que esa duración es bastante buena.

De todas formas, al ejecutarse de forma independiente al cliente, no fuerza a este a tener que esperar a que la ejecución termine, por lo que el usuario solo tiene que esperar a la generación del documento, la actualización se hace en una fase posterior e invisible para el usuario.

6.4 Recomendador de Zonas

A pesar de que, debido a la dificultad para obtener datos, no tanto sobre crímenes, si sobre sus víctimas, la idea es desarrollar una primera versión del recomendador, sobre la cual, conforme se generen nuevos datos, se le puedan realizar distintos entrenamientos y pruebas para ir comprobando la efectividad de este y poder realizar las modificaciones pertinentes con el objetivo de tener el mejor modelo posible para el momento del despliegue.

En una primera instancia, se planteó la posibilidad de extraer información de los medios de comunicación mediante web scraping (Martí, 2016), pero esta opción se descartó rápidamente, ya que, además de que no todas las noticias contienen información sobre la víctima, el hecho de seleccionar una gran cantidad de noticias de medios de comunicación introduciría un sesgo demasiado grande en el modelo.

Tras estas conclusiones, se inició un proceso de investigación en el que se estudiaron distintos tipos de modelos de aprendizaje automático, incluso, gracias a la participación en prácticas en empresa, se debatió con algún científico de datos sobre las distintas opciones, concluyendo que la más acertada, al menos de forma inicial, era usar el algoritmo K-Nearest Neighbors.

6.4.1 K-Nearest Neighbors

K-Nearest Neighbors (IBM, 2022) es uno de los modelos de aprendizaje supervisado más simples, pero también de los más populares, debido a su buen desempeño. Es utilizado tanto para tareas de clasificación como para regresión sobre la agrupación de un conjunto de datos, además de ser uno de los modelos más utilizados para el desarrollo de sistemas de recomendación en la web.

Se basa en un principio bastante simple, la proximidad, de forma que, se comparan los datos a clasificar o predecir, con una serie de vecinos cercanos y, en función de la similitud con estos, se le asigna un valor.

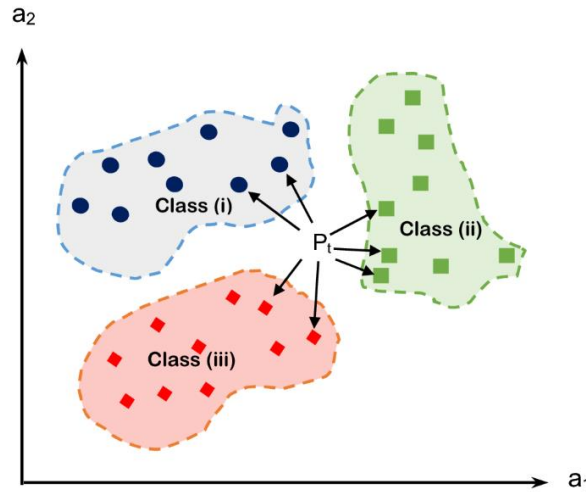


Figura 6.4.1: Ejemplo de K-Nearest Neighbors (Atallah, Badawy, & El-Sayed, 2019)

Es uno de los modelos de la familia conocida como "Lazy Learning", ya que, realmente, el modelo solo almacena el conjunto de datos y no existe una fase de entrenamiento como tal, por lo que todo el proceso de computación se realiza durante la clasificación o la predicción.

Retomando el tema de la proximidad, para poder decidir qué datos son más cercanos a otros, es necesario definir qué datos son más cercanos que a otros, y esto se realiza mediante la selección de una métrica de distancia, las más populares son las siguientes.

- Distancia euclídea: es la más usada, representa la distancia entre 2 puntos tradicional, mediante una línea recta.

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

Figura 6.4.2: Formula de la distancia euclídea

- Distancia Manhattan: otra de las más populares, es la diferencia en valor absoluto entre 2 puntos. Visto como una malla, es el número de movimientos necesarios para llegar desde el punto inicial hasta el final, es muy utilizada para los algoritmos de búsqueda de caminos en inteligencia artificial, sobre todo, de juegos.

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Figura 6.4.3: Formula de la distancia Manhattan

- Distancia de Minkowski: es la generalización de la distancia euclídea y Manhattan, permite generar nuevas métricas modificando el parámetro p .

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i| \right)^{\frac{1}{p}}$$

Figura 6.4.4: Formula de la distancia de Minkowski

Una vez se define la métrica de distancia a utilizar, resulta también importante la selección de un valor k adecuado, este parámetro define la cantidad de vecinos con los que se van a comparar los datos a procesar. Esta elección es realmente importante, ya que valores bajos de este parámetro generarán valores altos de varianza, mientras que valores altos, pueden producir sesgos, de forma que es necesario evitar tanto el overfitting como el underfitting.

La elección de este parámetro no tiene una regla escrita, depende en gran medida de los datos y se basa en la experimentación, aunque se recomienda usar siempre un número impar para evitar empates en la clasificación y usar validación cruzada, en general, realizar mucha experimentación.

Este algoritmo presenta increíbles ventajas, la implementación es realmente simple, además de que existen multitud de bibliotecas que ya ofrecen una versión eficiente implementada, para este proyecto, destaca sobre todo el hecho de que se adapta genial a nuevos datos, al solo necesitar almacenarlos en memoria y no precisa de una fase de entrenamiento, por lo que el hecho de reentrenarlo con los nuevos delitos que se publican no supondría un gran problema.

Pero evidentemente, también tiene una serie de desventajas, el escalado no es bueno, ya que hace un uso intensivo de memoria y, además, suele ser víctima de la denominada maldición de la dimensionalidad, es decir, su rendimiento se ve mermado al tratar con información con muchas dimensiones.

Esta serie de problemas deben ser tenidos en cuenta y, a la hora de realizar pruebas y obtener resultados, se deben tomar las decisiones oportunas sobre el mismo.

Por suerte, existen algunas soluciones aplicables a estos problemas, para el problema de la dimensionalidad, se pueden aplicar técnicas como feature selection (Wikipedia, 2022), que permiten extraer las características más influyentes a la hora de clasificar y poder reducir el número total de las mismas.

Para solventar los problemas de memoria existen estructuras de datos, como Árboles-KD (Hu, 2019), arboles binarios en los que los puntos se dividen en ejes alternos, de forma que cada hoja es un punto k-dimensional. Otra solución es el uso de Ball Trees (Dolatshah, Minaei, & Hadian, 2015), otro tipo de árbol de búsqueda en los que cada nodo es una esfera multidimensional que contiene un subconjunto de los puntos a buscar.

A pesar de estas estructuras de datos, si aún existieran problemas de rendimiento, sería necesario la elección de un modelo diferente a este, en concreto existe la posibilidad de usar sistemas de aproximación, como Locality Sensitive Hashing (LSH).

LSH (Hari, 2018) es un algoritmo basado en hashing que permite aproximar vecinos cercanos similares, de forma que, si dos puntos son cercanos en el espacio, tenderán a tener el mismo hash.

Al contrario que en criptografía, donde se tiende a minimizar el número de colisiones al generar los hashes, en este algoritmo se tiende a maximizarlo, lo que reduce en una gran cantidad, tanto el uso de memoria como de computación.

De todas formas, la experimentación es lo único que puede decidir que decisiones han de ser tomadas, es imposible determinar qué medidas de las comentadas deberán ser ejecutadas, pero es importante contemplar los posibles escenarios y plantear soluciones a los mismos, como se ha realizado aquí.

6.4.2 Desarrollo y Despliegue del Modelo

Una vez documentado el modelo, se ha procedido a el desarrollo de una versión, para poder ponerse en funcionamiento una vez existan los suficientes datos.

Para ello, se ha hecho uso de la biblioteca scikit-learn (scikit-learn developers, 2022), una biblioteca de Python orientada al desarrollo de modelos de aprendizaje automático. De la multitud de clases, se han utilizado KNeighborsClassifier (scikit-learn developers, 2022), un clasificador basado en K-Nearest Neighbors que ofrece las estructuras mencionadas anteriormente, además de estar optimizada para elegir los mejores parámetros (aun así, cuando se disponga de los datos suficientes será necesario experimentar), MinMaxScaler (scikit-learn developers, 2022), para escalar los datos de forma uniforme en el preprocesamiento, y LabelEncoder (scikit-learn developers, 2022), para codificar los strings de las zonas como valores numéricos.

La implementación no supone demasiada dificultad gracias a esta biblioteca que, junto a Pandas (Pandas Developers, 2022), permiten una rápida implementación y ofrecen multitud de herramientas para realizar pruebas y consultar, por ejemplo, matrices de confusión.

Para el despliegue, como se mencionó, se utilizará AI Platform (Google Developers, 2022), lo que permite un despliegue relativamente sencillo, se usarán buckets, como los utilizadas para almacenar las imágenes, para almacenar el modelo y los datos de entrenamiento.

Para almacenar el modelo en forma de fichero se hace uso de Joblib (Joblib developers, 2021), una biblioteca que permite cachear en memoria funciones, de forma que, una vez entrenado el modelo, se almacena como fichero joblib y se puede recuperar su estado de forma eficiente en cualquier momento.

Una vez almacenado el modelo en un bucket con el nombre "model.joblib", ejecutando un script de Python y subiendo los datos de entrenamiento al bucket, se puede entrenar el modelo en la infraestructura de Google, de forma que el modelo almacenado en el bucket se actualiza con el nuevo entrenamiento.

También, desde la consola de AI Platform es posible crear un modelo e indicarle la ubicación del fichero joblib, de forma que se construye automáticamente un punto de acceso desde donde es posible realizar consultas.

El desarrollo y avances en el uso de estas tecnologías permiten que, una vez se puedan integrar la aplicación y el modelo de aprendizaje, haya gran parte del trabajo hecho y, por tanto, la aplicación esté preparada para un procesamiento complejo de datos.

6.5 Diseño de la Interfaz de Usuario

Más allá de la funcionalidad, uno de los principales puntos que definen el éxito o el fracaso de la aceptación por parte de los usuarios finales es la interfaz y la experiencia de usuario, más aún para el desarrollo de aplicaciones móviles. Dada la gran importancia de esta y, a pesar de no ser experto en diseño de interfaces, se ha intentado realizar un diseño lo más amigable posible para el usuario, basándose en la propia experiencia con otras aplicaciones y lo visto en asignaturas como Interfaces de Usuario.

Para lograrlo, se ha hecho uso de la biblioteca de componentes de interfaz ofrecida por Ionic (Ionic, 2022), una gran colección compuesta principalmente por elementos orientados al desarrollo móvil.

El diseño de la aplicación se ha realizado en base a dos componentes principales, `IonGrid`, que permite definir mallas a través de filas y columnas, con `IonRow` e `IonCol` que se adaptan a la pantalla del dispositivo.

El otro se trata de `IonCard`, un componente que permite definir secciones de contenido en formato de carta, esto ha servido de utilidad para mejorar la estética de la aplicación, como se puede observar en la Figura 6.5.1.

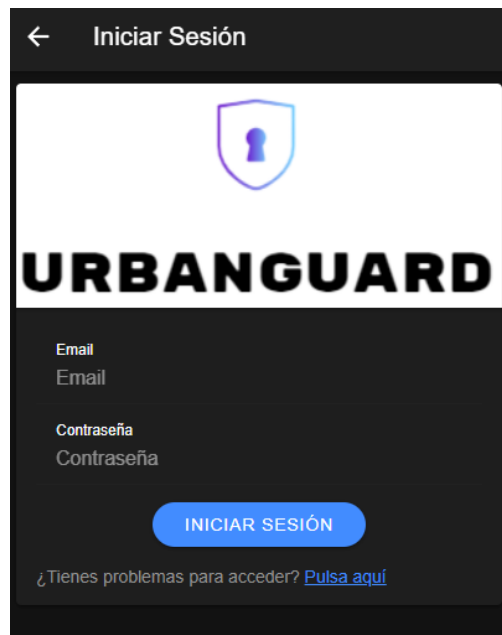
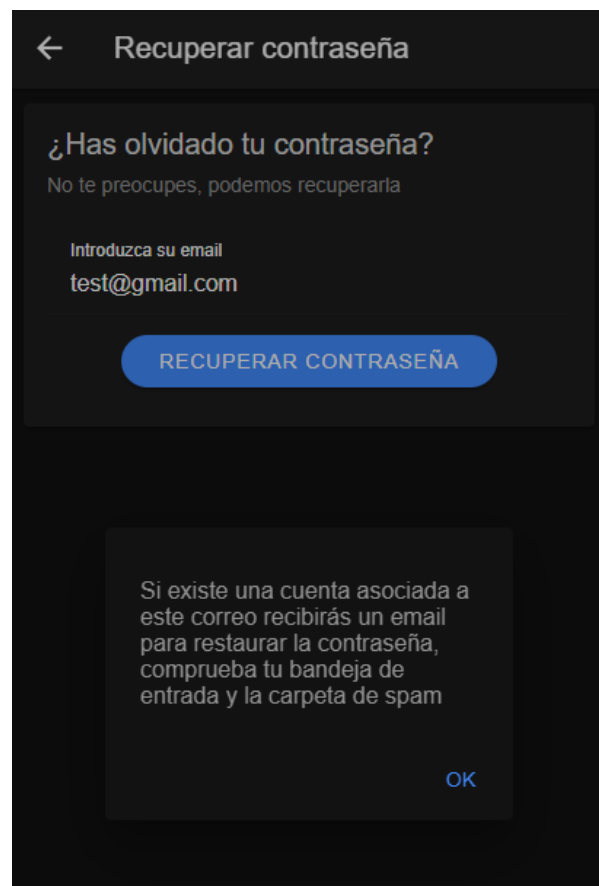
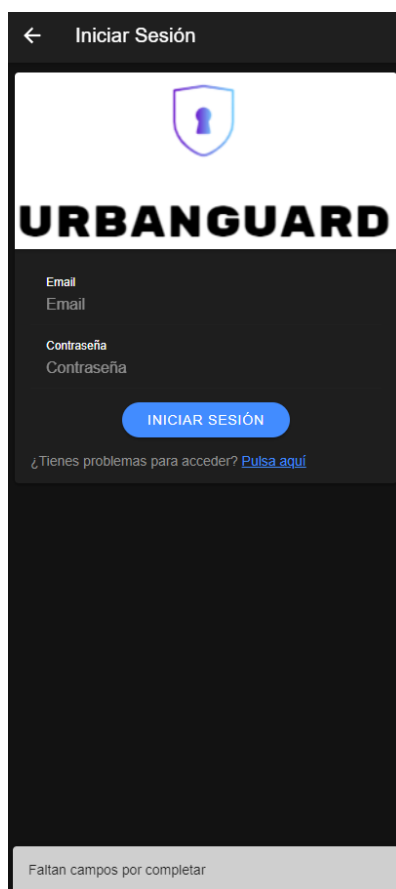


Figura 6.5.1: Ejemplo de uso de IonCard

En cuanto a mecanismos de retroalimentación, con el objetivo de que el usuario pueda recibir respuestas del sistema, se han implementado medidas típicas en el desarrollo móvil a través de los componentes IonToast e IonAlert (Figuras 6.5.2 y 6.5.3).




Figuras 6.5.2 y 6.5.3: Ejemplos de uso de IonToast e IonAlert respectivamente

Como método de entrada para el usuario, en el caso de textos y fechas, se ha hecho uso del componente `IonInput` (Figura 6.5.4); en el caso de selecciones de valores, se ha utilizado `IonSelect` (Figura 6.5.5).

← Registro

Estas a punto de entrar a UrbanGuard
Crea tu cuenta y contribuye a crear una Malaga más segura

IMPORTANTE
Asegurate de introducir tus datos correctamente, solo podrás modificar tu foto de perfil y tu nombre público



SELECCIONAR IMAGEN

Nombre *
Nicolas

Apellidos *
Quintana

Email *
test@uma.es

Fecha de nacimiento *
25/04/2000

Sexo *
Masculino

Contraseña *

Confirmar Contraseña *


Condiciones de interés
Color de piel, Discapacidad

REGÍSTRATE

← Registro

Estas a punto de entrar a UrbanGuard
Crea tu cuenta y contribuye a crear una Malaga más segura

IMPORTANTE
Asegurate de introducir tus datos correctamente, solo podrás modificar tu foto de perfil y tu nombre público



Condiciones de interés

SELECCIONAR

Nombre *
Nicolas

Apellidos *
Quintana

Email *
test@ur

Fecha de nacimiento *
25/04/2000

Sexo *
Masculi

Contraseña *

Confirmar Contraseña *

Condiciones de interés
Elige varios

REGÍSTRATE

Figuras 6.5.4 y 6.5.5: Ejemplos de `IonInput` e `IonSelect` respectivamente

Para la navegación se ha desarrollado un menú personalizado (Figura 6.5.6) a raíz del componente base `IonMenu`, tal como se comentaba anteriormente, se ha orientado la navegación de forma que, al cambiar de sección, desaparezca el menú y aparezca un menú de retroceso. De esta forma, se evitan rutas de navegación infinitas y se mejora la facilidad del usuario a desarrollar un modelo mental que le permita optimizar su interacción con la aplicación. Además, se han acompañado todas las opciones de un icono con el mismo objetivo anterior, mejorar la experiencia.

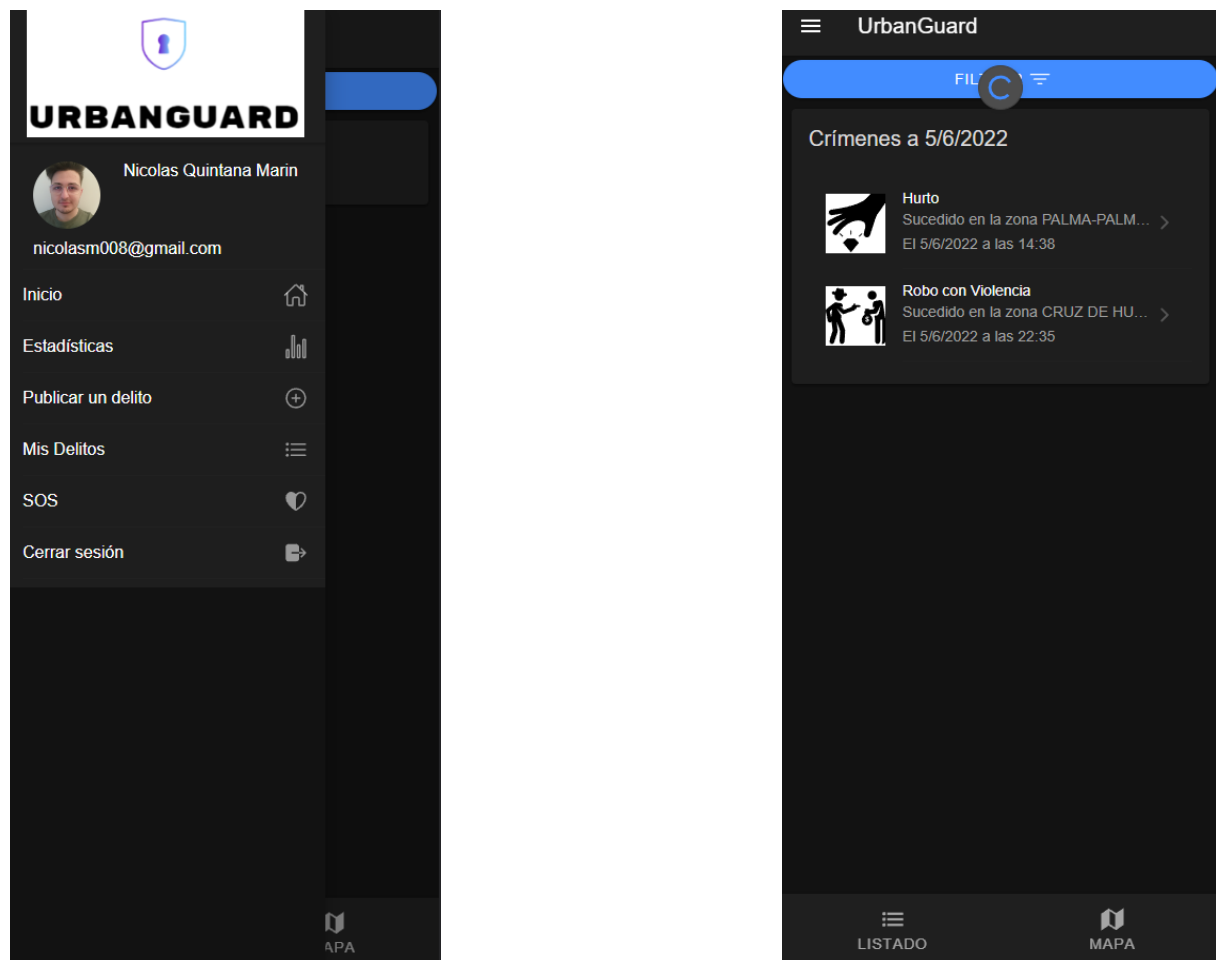


Figura 6.5.6 y 6.5.7: Menú de la aplicación y Ejemplo de IonRefresher

Para refrescar contenido en la pantalla de inicio y poder consultar si existen nuevos delitos, en lugar de introducir el típico botón "Actualizar" que además de ocupar espacio de pantalla, algo crucial en las interfaces móviles, no es demasiado estético, se ha implementado a través de un gesto de deslizamiento hacia abajo mediante el componente IonRefresher (Figura 6.5.7).

Además, para permitir al usuario ser consciente de que se está realizando algún proceso o alguna bajada o subida de datos, que requiere de un tiempo de respuesta algo más elevado, como puede ser por ejemplo el caso de las imágenes, se han incluido elementos para indicarlo, en concreto, se ha hecho uso de IonLoading e IonSpinner, como se puede observar en la Figura 6.5.8.

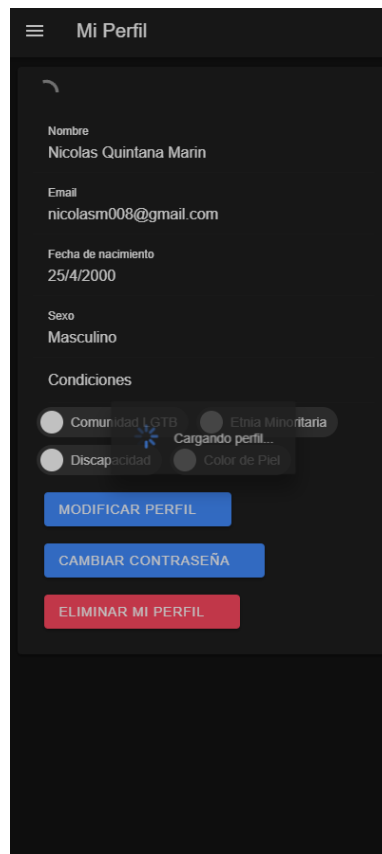
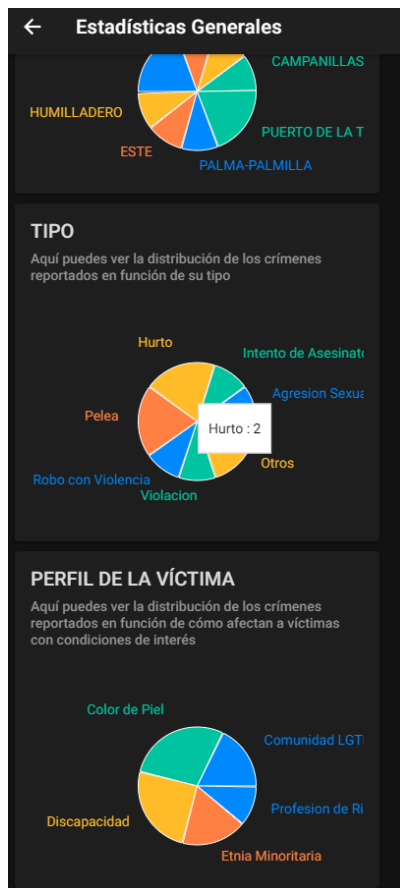


Figura 6.5.8: Ejemplo de carga de pantalla

El diseño de las pantallas ha ido orientado un principal objetivo, pantallas simples que ocupen toda la visión del usuario, de forma que no sea necesario deslizar hacia abajo para ver más contenido. En algunos casos, como es obvio, esto es totalmente imposible, por ejemplo, en el caso de los listados y las estadísticas.

Para el visionado de las estadísticas, como ya se mencionó anteriormente, se ha hecho uso de la biblioteca Recharts (Recharts Group, 2021), de la que se han utilizado dos componentes principales BarChart y PieChart. Debido al gran número de valores, por ejemplo, en las zonas, es inviable que se visualicen a la perfección todos los valores en la pantalla, por lo que se le ha permitido al usuario pulsar sobre una zona de la gráfica y ver sus valores (Ver Figura 6.5.9).



Figuras 6.5.9 y 6.5.10: Ejemplos de Estadísticas y Mapas

La integración de los mapas se ha realizado con una biblioteca de React, en concreto, React-Google-Maps/Api (Lyakhov & Klar, 2022), basado principalmente en el componente GoogleMap, además de marcadores, Marker, de los cuales se han personalizado tanto su aspecto como su comportamiento al ser pulsados, y Polygon, usado para resaltar la zona filtrada, como se puede observar en la Figura 6.5.10.

7. Conclusiones y Líneas Futuras

La realización de este Trabajo Fin de Grado ha permitido combinar todos los conocimientos adquiridos durante la realización del grado y enfocarlo a un caso práctico real, de forma que estos conocimientos se han vuelto efectivos y han tenido un impacto real a la hora de mejorar el rendimiento durante este proyecto.

Como resultado, se ha generado una aplicación Android basada en una plataforma en la nube como Firebase que permite a los ciudadanos de Málaga publicar y consultar los delitos que ellos mismos pueden reportar, además de consultar una serie de estadísticas sobre los mismos y tener una vía de socorro rápido en caso de necesitarlo. El principal objetivo y deseo de este proyecto es que permita reducir las tasas de criminalidad y, sobre todo, evitar algún mal mayor o, al menos, prevenirlo gracias a las señales de SOS.

Además, se ha realizado un aprendizaje de tecnologías como React, NodeJS, TypeScript, Firebase, Python y Pandas y se ha comenzado a manejar otras como scikit-learn para el desarrollo de modelos de aprendizaje computacional y Google Cloud Platform para su despliegue, todas estas son tecnologías que no se aprenden explícitamente durante el trascurso del grado y que su conocimiento será de provecho tanto para futuros proyectos personales como para la incorporación al mundo laboral.

En general ha sido una experiencia reconfortante, el comienzo de un desarrollo siempre es muy duro, y más aun cuando no se conocen completamente las tecnologías, pero la sensación de avanzar, ver las primeras funcionalidades y, finalmente, manejar la aplicación final y poder

mostrarla a los demás es una experiencia fantástica que motiva a seguir trabajando en la misma.

Como se ha mencionado, esta aplicación, a pesar de ser totalmente funcional y desplegable, no representa la totalidad de ideas que existen sobre la misma, por ello se plantean una serie de líneas futuras a abarcar una vez finalice el mismo.

- Adaptar la aplicación al cumplimiento del RGPD: en primer lugar, habría que cumplir todos los requisitos exigidos por el RGPD ya que, además, la aplicación hace uso de datos personales de categoría especial.
- Compilación en iOS y despliegue en las tiendas de aplicaciones: la aplicación se ha compilado para Android ya que, al tener un móvil personal del mismo tipo, las pruebas en un dispositivo físico solo eran posibles con este. El lado bueno es que haber elegido una tecnología como Ionic, que permite desarrollos multiplataforma, hace realmente fácil compilar una versión para iOS siguiendo una guía de la propia documentación. Tras esto, se publicaría en las tiendas de ambos sistemas operativos.
- Desarrollar un sistema de pruebas automático: las pruebas se han realizado de forma manual, sería necesario automatizar dichas pruebas y la posibilidad de generar informes, para comprobar realmente la validez del sistema.
- Mejora de la IU: a pesar de que se ha trabajado en la interfaz de usuario, esta es bastante mejorable, por ello, una de las líneas a seguir es adquirir una mayor formación en cuanto a diseño de interfaces y lograr mejorarla, sobre todo para ofrecer una mejor experiencia de usuario y llegar a un mayor público.
- Despliegue de la fase 2: una vez la aplicación esté puesta en producción y recoja los suficientes datos, se podría iniciar la fase de experimentación con el modelo de aprendizaje automático y su consecuente integración con la aplicación.
- Cambio en los mapas con nuevo plugin: durante el desarrollo de la aplicación y cuando esta parte ya estaba implementada, se ha publicado un plugin oficial que permite la integración directa de Capacitor con Google Maps de forma nativa (Vidas, 2022), sin hacer uso de bibliotecas de React, lo que mejora su eficiencia, uno de los pasos a seguir sería realizar la sustitución de la biblioteca por el nuevo plugin.

- Implementación de un sistema de moderación: aunque la idea de esta aplicación persigue buenos propósitos puede usarse para malas causas, como realizar reportes falsos en determinadas zonas para que los ciudadanos no acudan a ella y los negocios se puedan ver afectados, para evitar esto, será necesario implementar algún sistema que permita reportar delitos como falsos y se realice una verificación de este, o un sistema de votación, es una idea que se debe estudiar aún.
- Integración con las fuerzas del orden: uno de los mejores futuros escenarios es ofrecer esta plataforma al Ayuntamiento de Málaga y que se integre como aplicación oficial, de forma que sea mantenida, por ejemplo, por la Policía Nacional, lo que puede permitirle recibir las alertas de SOS en tiempo real, ver las zonas más peligrosas y delitos que acaben de suceder, además de facilitarle distribuir las unidades de una forma más efectiva. La verificación de los delitos también sería más eficiente.
- Ampliación de la funcionalidad: en algún punto del desarrollo existió la idea de implementar un sistema de match para el movimiento de personas, de forma que la aplicación permitiera indicar tu intención de moverte hacia un punto y encontrará perfiles que quieran hacer un movimiento similar, para evitar realizarlos en solitario, sobre todo a altas horas de la noche. Se pretendía acompañar con un modelo de mensajería para poder contactar con los compañeros elegidos, la falta de tiempo ha hecho esto imposible, por lo que se pretende desarrollar una vez se terminé con el proyecto.
- Expansión a otras ciudades: la aplicación, a pesar de estar orientada a Málaga, permite añadir nuevas zonas con facilidad, de la misma forma, se podrían añadir ciudades y agrupar cada zona con su ciudad, así se podría expandir y utilizar en cualquier ciudad del mundo.

Bibliografía

- Abbo. (2022). *Delito de Hurto: concepto, tipos, penas y diferencia con Robo*. Obtenido de ConceptosJuridicos: <https://www.conceptosjuridicos.com/hurto/>
- Abbo. (2022). *Robo con violencia: concepto, regulación y penas para este delito*. Obtenido de ConceptosJuridicos: <https://www.conceptosjuridicos.com/robo-con-violencia/>
- Alcácer, V., & Cruz-Machado, V. (2019). Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems. *Engineering Science and Technology, an International Journal*, 900-912.
- Amazon Web Services. (2022). *Informática sin servidor: Amazon Web Services*. Obtenido de Amazon Web Services: <https://aws.amazon.com/es/serverless/>
- Atallah, D. M., Badawy, M., & El-Sayed, A. (27 de Septiembre de 2019). *Intelligent feature selection with modified K-nearest neighbor for kidney transplantation prediction*. Obtenido de SpringerLink: <https://link.springer.com/article/10.1007/s42452-019-1329-z>
- Atlassian. (2021). *Trello*. Obtenido de <https://trello.com/es>
- Ayuntamiento de Málaga. (Junio de 2011). *Mapa Distritos Municipales Málaga*. Obtenido de Ayuntamiento de Málaga: https://www.malaga.eu/recursos/ayto/distritos/distritosd_municipales.html
- Ayuntamiento de Málaga. (2022). *Aplicaciones*. Obtenido de Datos abiertos Ayto. Málaga: <https://datosabiertos.malaga.eu/aplicaciones>
- Ayuntamiento de Málaga. (2022). *Sistema de Información Cartográfica - Distrito Municipal*. Obtenido de Ayuntamiento de Málaga: <https://datosabiertos.malaga.eu/dataset/sistema-de-informacion-cartografica-distrito-municipal>
- Cadavid Navarro, D., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 30-39.
- Capacitor. (2022). *Camera Capacitor Plugin*. Obtenido de Capacitor: <https://capacitorjs.com/docs/apis/camera>

Capacitor. (2022). *Capacitor: Cross-platform Native Runtime for Web Apps*. Obtenido de Capacitor: <https://capacitorjs.com/docs>

Capacitor. (2022). *Geolocation Capacitor Plugin*. Obtenido de Capacitor: <https://capacitorjs.com/docs/apis/geolocation>

Capacitor. (2022). *Push Notifications Capacitor Plugin*. Obtenido de Capacitor: <https://capacitorjs.com/docs/apis/push-notifications>

Dassault Systèmes. (2022). *MagicDraw*. Obtenido de <https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/>

Díaz Piraquive, F. N., González Crespo, R., Pérez Castillo, J. N., & Medina García, V. H. (2015). Motivos de fracaso en los proyectos de Tecnologías de Información y Comunicaciones. *13th LACCEI Annual International Conference*, (págs. 1-11). Santo Domingo.

Dolatshah, M., Minaei, B., & Hadian, A. (2 de Noviembre de 2015). *Ball*-tree: Efficient spatial indexing for constrained nearest-neighbor search in metric spaces*. Obtenido de ResearchGate: https://www.researchgate.net/publication/283471105_Ball-tree_Efficient_spatial_indexing_for_constrained_nearest-neighbor_search_in_metric_spaces

EpData. (2022). *EpData*. Obtenido de <https://www.epdata.es/>

EpData. (2 de Junio de 2022). *Todos los Datos de crimen en España hoy: asesinatos, robos, secuestros y otros delitos*. Obtenido de EpData: <https://www.epdata.es/datos/crimen-espana-hoy-asesinatos-robos-secuestros-otros-delitos/4/espana/106>

GeeksforGeeks. (13 de Enero de 2022). *How to check if a given point lies inside or outside a polygon?* Obtenido de GeeksforGeeks: <https://www.geeksforgeeks.org/how-to-check-if-a-given-point-lies-inside-a-polygon/>

GeeksforGeeks. (3 de Mayo de 2022). *How to check if two given line segments intersect?* Obtenido de GeeksforGeeks: <https://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/>

GitHub. (2022). *Git Hub*. Obtenido de <https://github.com/>

Gómez Jiménez, A. (2019). *Victimología - Tema 2*. Obtenido de StuDocu: <https://www.studocu.com/es/document/universidad-de-malaga/victimologia/tema-2-apuntes-2/9944043>

Google. (2022). *Servicios de Cloud Computing*. Obtenido de Google Cloud: <https://cloud.google.com/?hl=es>

Google Developers. (2022). *Agrega Firebase al proyecto de JavaScript*. Obtenido de Firebase: <https://firebase.google.com/docs/web/setup?hl=es>

Google Developers. (2022). *AI Platform*. Obtenido de Google Cloud: <https://cloud.google.com/ai-platform/docs/technical-overview?hl=es-419>

Google Developers. (2022). *Android Studio*. Obtenido de <https://developer.android.com/studio>

Google Developers. (2022). *Cloud Firestore*. Obtenido de Firebase: <https://firebase.google.com/docs/firestore>

Google Developers. (2022). *Cloud Functions for Firebase*. Obtenido de Firebase: <https://firebase.google.com/docs/functions>

Google Developers. (2022). *Cloud Storage for Firebase*. Obtenido de Firebase: <https://firebase.google.com/docs/storage>

Google Developers. (2022). *Firebase*. Obtenido de <https://firebase.google.com/>

Google Developers. (2022). *Firebase Authentication*. Obtenido de Firebase: <https://firebase.google.com/docs/auth>

Google Developers. (2022). *Firebase Cloud Messaging*. Obtenido de Firebase: <https://firebase.google.com/docs/cloud-messaging>

Hari, S. (5 de Julio de 2018). *Locality Sensitive Hashing for Similar Item Search*. Obtenido de Towards Data Science: <https://towardsdatascience.com/locality-sensitive-hashing-for-music-search-f2f1940ace23>

Hu, Y. (18 de Julio de 2019). *KD-Tree*. Obtenido de <https://yashen.github.io/post/kd-tree/>

Iberdrola. (2022). *'Smart cities': la revolución tecnológica llega a las ciudades*. Obtenido de Iberdrola: <https://www.iberdrola.com/innovacion/smart-cities>

IBM. (2022). *K-Nearest Neighbors Algorithm*. Obtenido de IBM: <https://www.ibm.com/topics/knn>

Ionic. (2022). *Image Picker*. Obtenido de Ionic: <https://ionicframework.com/docs/native/image-picker>

Ionic. (2022). *Introduction to Ionic*. Obtenido de Ionic: <https://ionicframework.com/docs>

Ionic. (2022). *UI Components*. Obtenido de Ionic: <https://ionicframework.com/docs/components>

JetBrains. (2022). *IntelliJ IDEA*. Obtenido de <https://www.jetbrains.com/es-es/idea/>

JGraph Ltd. (2021). *Diagram Software and Flowchart Maker*. Obtenido de <https://www.diagrams.net/>

Joblib developers. (2021). *Joblib: running Python functions as pipeline jobs*. Obtenido de <https://joblib.readthedocs.io/en/latest/>

Langdon Villamayor, Á. (2020). *MAPA DEL CRIMEN DE ESPAÑA*. Obtenido de EntreDatos: <https://entredatos.es/mapa-del-crimen-en-espana/>

Lenguaje Jurídico. (11 de Junio de 2021). *¿Qué son las categorías especiales de datos personales?* Obtenido de Lenguaje Jurídico: <https://www.lenguajejuridico.com/diccionario-juridico/derecho-digital/proteccion-de-datos-personales/categorias-especiales-de-datos-personales/>

Llamas, J. (12 de Mayo de 2021). *Era digital*. Obtenido de Economipedia: <https://economipedia.com/definiciones/era-digital.html>

López-Fonseca, Ó. (28 de Julio de 2021). *Los delitos de odio repuntan hasta alcanzar cifras superiores a las de antes de la pandemia*. Obtenido de El País: <https://elpais.com/espana/2021-07-28/los-delitos-de-odio-repuntan-hasta-alcanzar-cifras-superiores-a-las-de-antes-de-la-pandemia.html>

Lyakhov, A., & Klar, U. (1 de Junio de 2022). *React Google Maps Api Style Guide*. Obtenido de <https://react-google-maps-api-docs.netlify.app/>

Martí, M. (8 de Abril de 2016). *Qué es el Web scraping? Introducción y herramientas*. Obtenido de sitelabs: <https://sitelabs.es/web-scraping-introduccion-y-herramientas/>

MDN. (8 de Diciembre de 2020). *Aplicaciones Web Progresivas*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/Progressive_web_apps

Meta Platforms. (2022). *React*. Obtenido de <https://es.reactjs.org/>

Microsoft. (2022). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com/>

Ministerio del Interior. (2022). *Hechos conocidos por comunidades autónomas, tipología penal y periodo*. Obtenido de PORTAL ESTADÍSTICO DE CRIMINALIDAD: <https://estadisticasdecriminalidad.ses.mir.es/publico/portalestadistico/portal/datos.html?type=pcaxis&path=/Datos1/&file=pcaxis>

Pandas Developers. (2022). *pandas - Python Data Analysis Library*. Obtenido de <https://pandas.pydata.org/>

Recharts Group. (2021). *Recharts*. Obtenido de <https://recharts.org>

Redacción Law&Trends. (16 de Mayo de 2022). *Homicidio en grado de tentativa*. Obtenido de Law&Trends: <https://www.lawandtrends.com/noticias/penal/homicidio-en-grado-de-tentativa-1.html>

Remix. (2022). *React Router*. Obtenido de <https://reactrouter.com/>

Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional.

scikit-learn developers. (2022). *KNeighborsClassifier*. Obtenido de scikit-learn: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

scikit-learn developers. (2022). *LabelEncoder*. Obtenido de scikit-learn: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>

scikit-learn developers. (2022). *MinMaxScaler*. Obtenido de scikit-learn: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

scikit-learn developers. (2022). *scikit-learn*. Obtenido de <https://scikit-learn.org/stable/index.html>

Silva, S., & Pereira, D. (8 de Octubre de 2021). *Capacitor FCM*. Obtenido de GitHub: <https://github.com/capacitor-community/fcm>

Software Freedom Conservancy. (2022). *Git*. Obtenido de <https://git-scm.com/>

SVG Viewer. (2022). *SVG Viewer - View, save, and optimize SVGs*. Obtenido de <https://www.svgviewer.dev/>

Vacek, L., Atter, E., Rizo, P., & Nam, B. (Junio de 2017). *sUAS for Deployment and Recovery of an Environmental Sensor Probe*. Obtenido de ResearchGate: https://www.researchgate.net/publication/317957001_sUAS_for_Deployment_and_Recovery_of_an_Environmental_Sensor_Probe

Vidal Rodríguez, G. (19 de Abril de 2022). *El delito de agresión sexual en el Código Penal*. Obtenido de <https://www.gersonvidal.com/blog/delito-agresion-sexual/>

Vidas, T. (29 de Abril de 2022). *Announcing the Capacitor Google Maps Plugin*. Obtenido de Ionic: <https://ionicframework.com/blog/announcing-the-capacitor-google-maps-plugin/>

Wikipedia. (16 de Abril de 2020). *Regla par-impar*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Regla_par-impar

Wikipedia. (2 de Junio de 2022). *Feature selection*. Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Feature_selection

Wikipedia. (9 de Junio de 2022). *Point in polygon*. Obtenido de Wikipedia:
https://en.wikipedia.org/wiki/Point_in_polygon

Wikipedia. (1 de Mayo de 2022). *Transformación digital*. Obtenido de Wikipedia, la enciclopedia libre: https://es.wikipedia.org/wiki/Transformaci%C3%B3n_digital

Wolters Kluwer. (2022). *Riña tumultuaria*. Obtenido de La Ley:
https://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAA AAAAEAMtMSbF1jTAAAUMTQwMTtbLUouLM_DxblwMDCwNzAwuQQGZapUt-ckhlQaptWmJOcSoAQJHm4jUAAAA=WKE

Wolters Kluwer. (2022). *Violación*. Obtenido de La Ley:
https://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAA AAAAEAMtMSbF1jTAAAUMTQwMTtbLUouLM_DxblwMDCwNzAwuQQGZapUt-ckhlQaptWmJOcSoAQJHm4jUAAAA=WKE

Zuil, M., & Ojeda, D. (6 de Julio de 2021). *Así están creciendo los delitos de odio en toda España*. Obtenido de El Confidencial: https://www.elconfidencial.com/espana/2021-07-06/delitos-de-odio-agresiones-homofobia-espana-aumento_3168175/

Apéndice A: Manual de Usuario

En este apéndice se expone una guía de como hacer uso de las distintas funcionalidades de la aplicación.

Inicio de Sesión

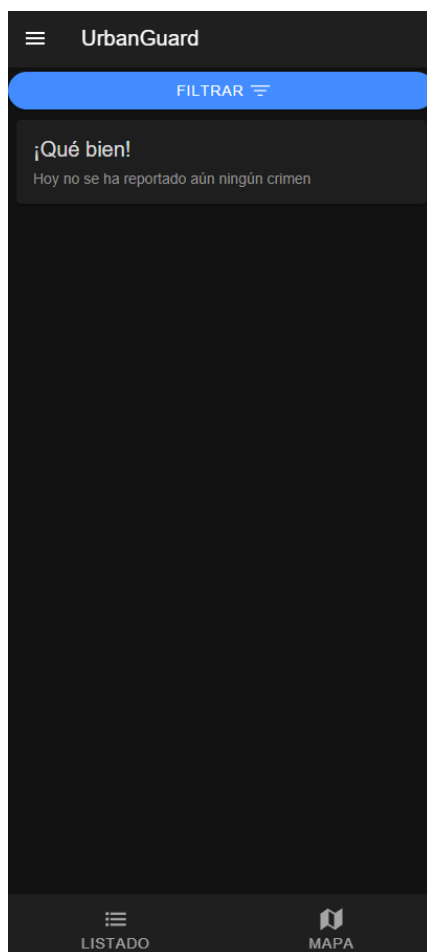
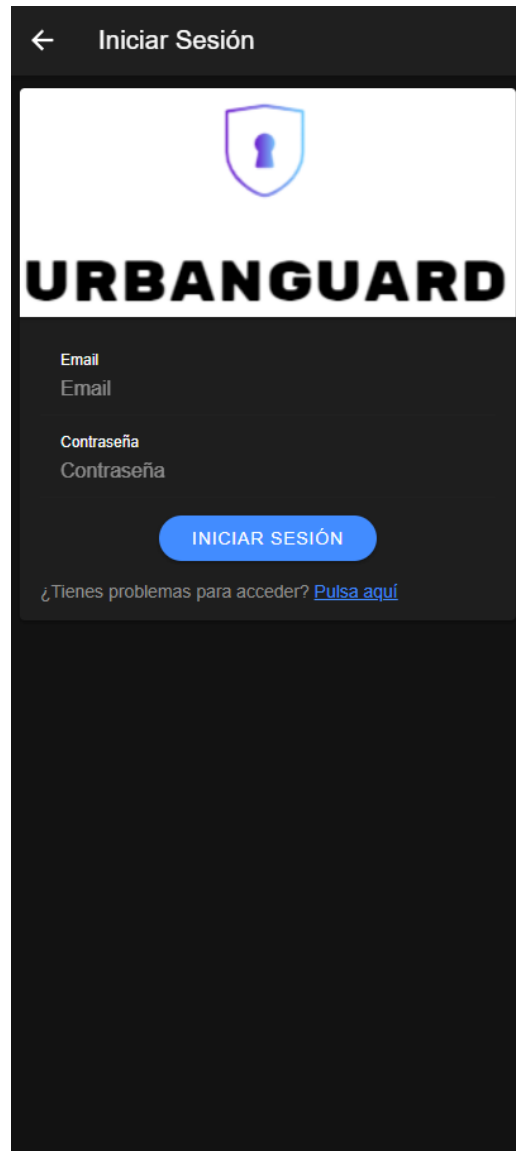
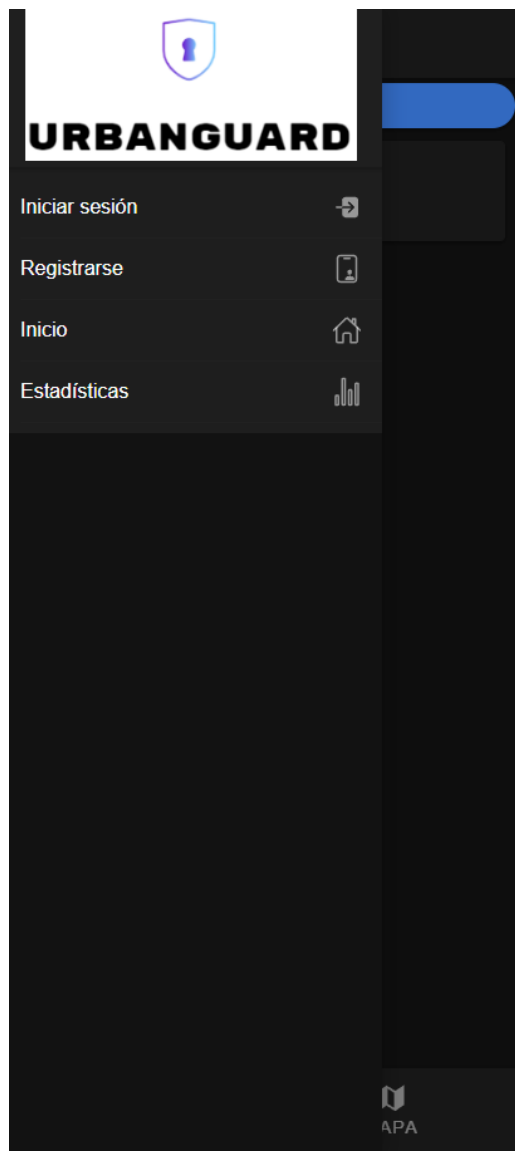


Figura A.1: Pantalla de Inicio

En primer lugar, abrimos la aplicación y, desde la pantalla inicial (Figura A.1), pulsamos el icono "≡" y se desplegará el menú de la aplicación (Figura A.2).



Figuras A.2 y A.3: Menú e Inicio de Sesión

Una vez desplegado pulsamos "Iniciar Sesión" y se mostrará la pantalla de iniciar sesión (Figura A.3), se introducen los datos y, al pulsar "Iniciar Sesión" se realizará el inicio de sesión.

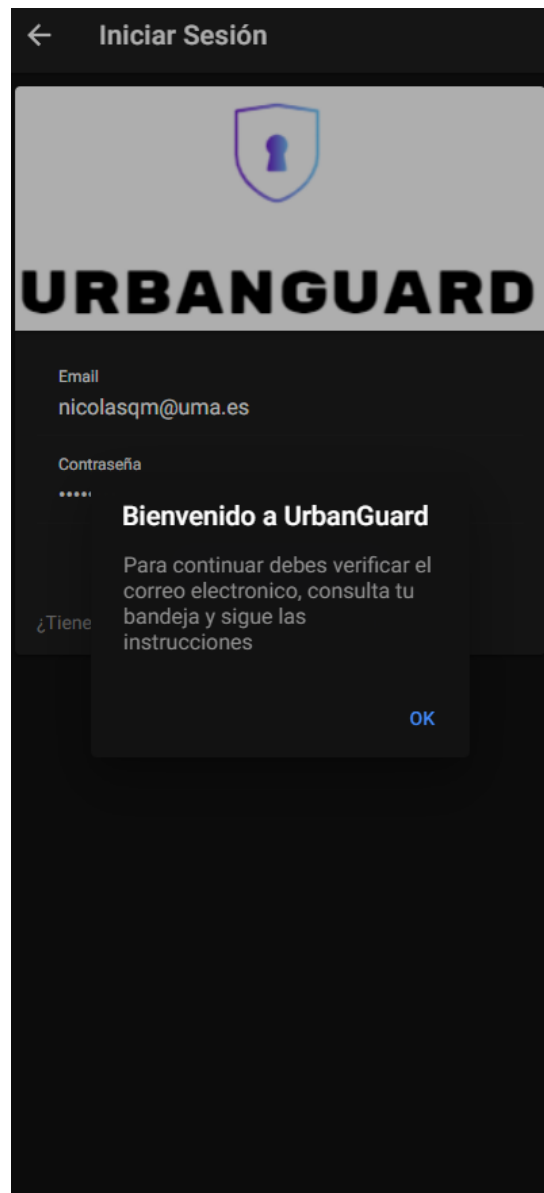
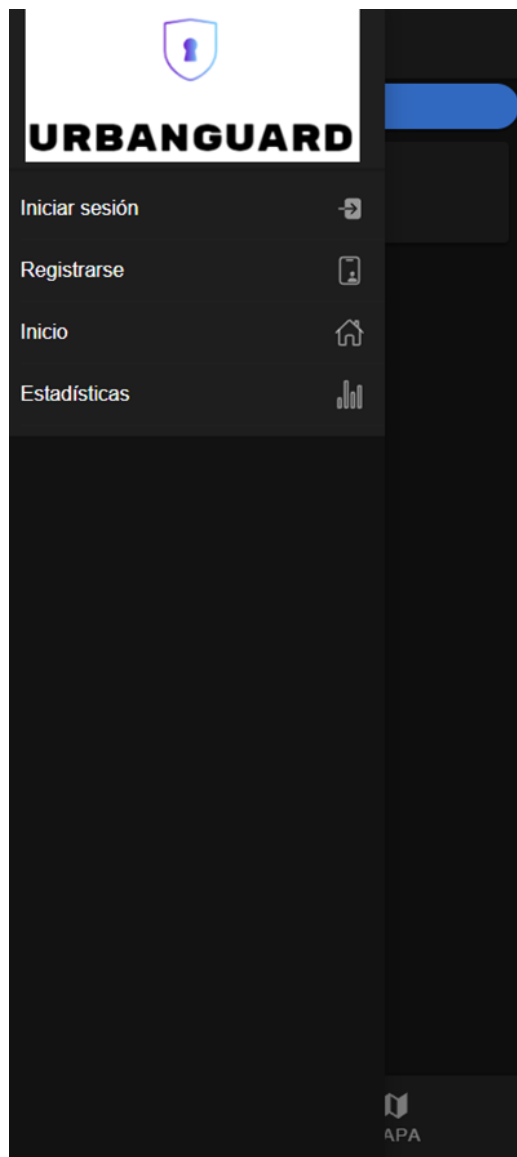


Figura A.4: Aviso de verificación de email

Si el email no esta verificado, se mostrará el siguiente mensaje y será necesario consultar el email para verificarlo (Ver Figura A.4).

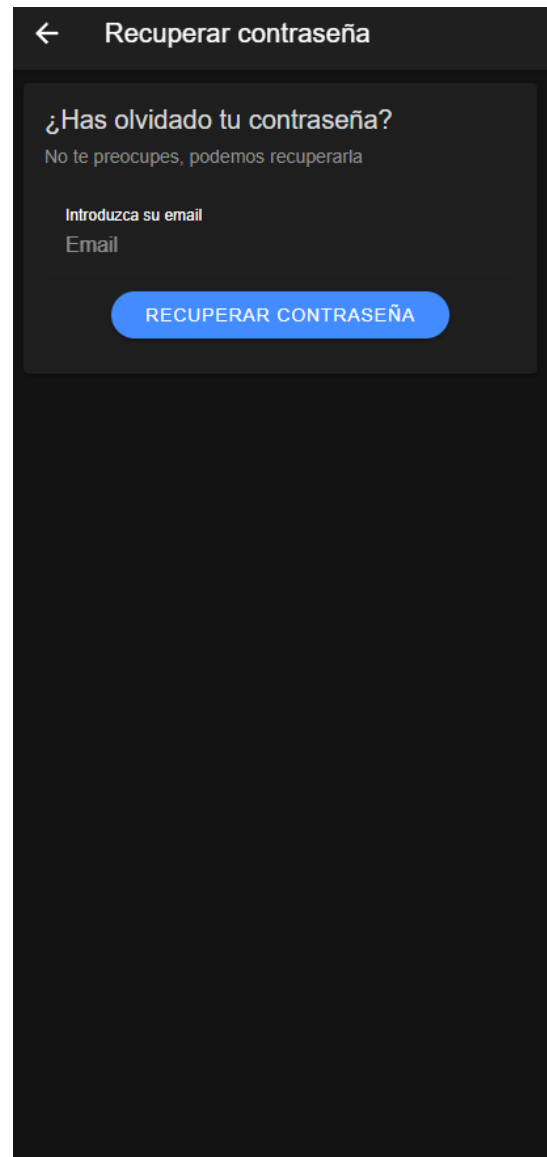
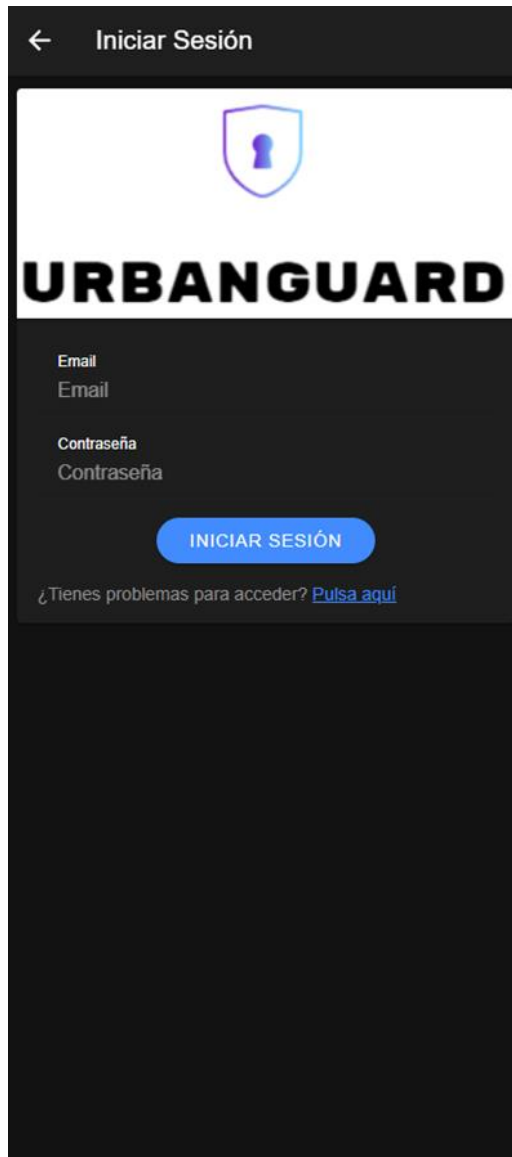
Registro

This screenshot shows the registration screen of the UrbanGuard app. At the top, there is a back arrow and the title "Registro". Below this, there is a message: "Estas a punto de entrar a UrbanGuard" followed by "Crea tu cuenta y contribuye a crear una Malaga más segura". An "IMPORTANT" section follows, stating: "Asegurate de introducir tus datos correctamente, solo podrás modificar tu foto de perfil y tu nombre público". Below this is a profile picture placeholder and a blue button labeled "SELECCIONAR IMAGEN". The form contains several fields: "Nombre *" (Nombre), "Apellidos *" (Apellidos), "Email *" (Email), "Fecha de nacimiento *" (dd/mm/aaaa) with a calendar icon, "Sexo *" (Elige Uno) with a dropdown arrow, "Contraseña *" (Contraseña), "Confirmar Contraseña *" (Confirmar Contraseña), and "Condiciones de interés" (Elige varios) with a dropdown arrow. At the bottom right, there is a large blue button labeled "REGÍSTRATE".

Figuras A.5 y A.6: Menú y Pantalla de Registro

Para el registro, volvemos a abrir el menú y esta vez hacemos click en "Registrarse" (Figura A.5), se abrirá la pantalla de registro (Figura A.6), cumplimentados todos los campos obligatorios indicados con un * y condiciones de interés, si procede. Podemos elegir la imagen desde la galería o desde la propia cámara. Después pulsamos "Regístrate" y se creará el usuario, tras esto, es necesario iniciar sesión.

Recuperar contraseña



Figuras A.7 y A.8: Inicio de Sesión y Recuperación de Contraseña

Para recuperar la contraseña, desde el inicio de sesión (Figura A.7), pulsamos en la zona indicada para recuperar la contraseña, lo que abrirá dicha pantalla (Figura A.8). Introducimos el email y, al pulsar sobre "Recuperar Contraseña", si el email existe, un correo electrónico de restablecimiento de contraseña será enviado. De todas formas, se muestra un mensaje diciendo que, en el caso de existir, se enviará un correo a la dirección.

Modificar Perfil

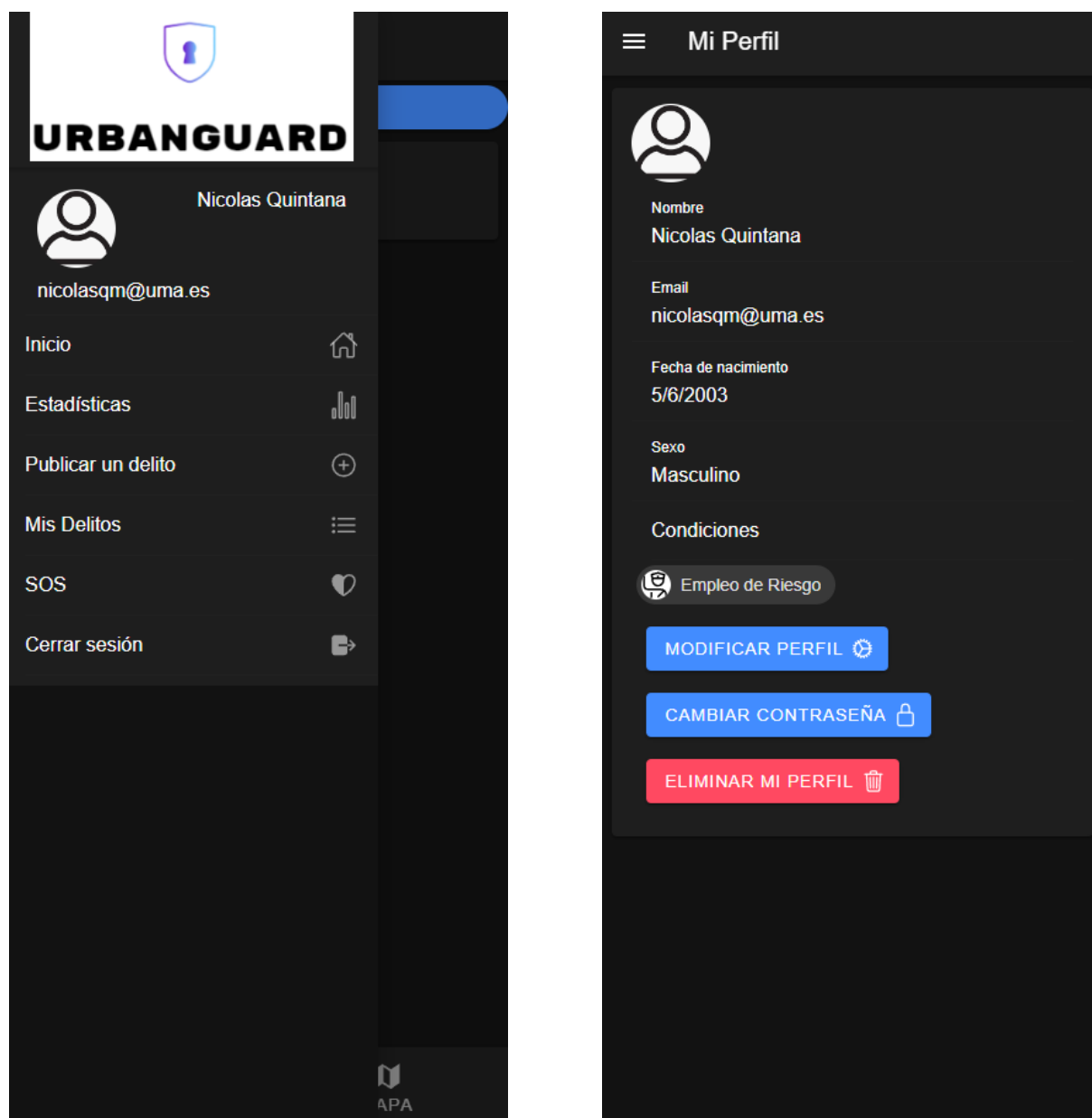


Figura A.9 y A.10: Menú y Perfil del Usuario

Habiendo previamente iniciado sesión en la aplicación, pulsamos en cualquier punto de la sección en donde se muestra nuestra foto de perfil, email y nombre (Figura A.9) y se mostrará la información de nuestro perfil (Figura A.10). Desde nuestro perfil, pulsamos "Modificar Perfil" y se mostrará la pantalla de modificación de perfil (Figura A.11).

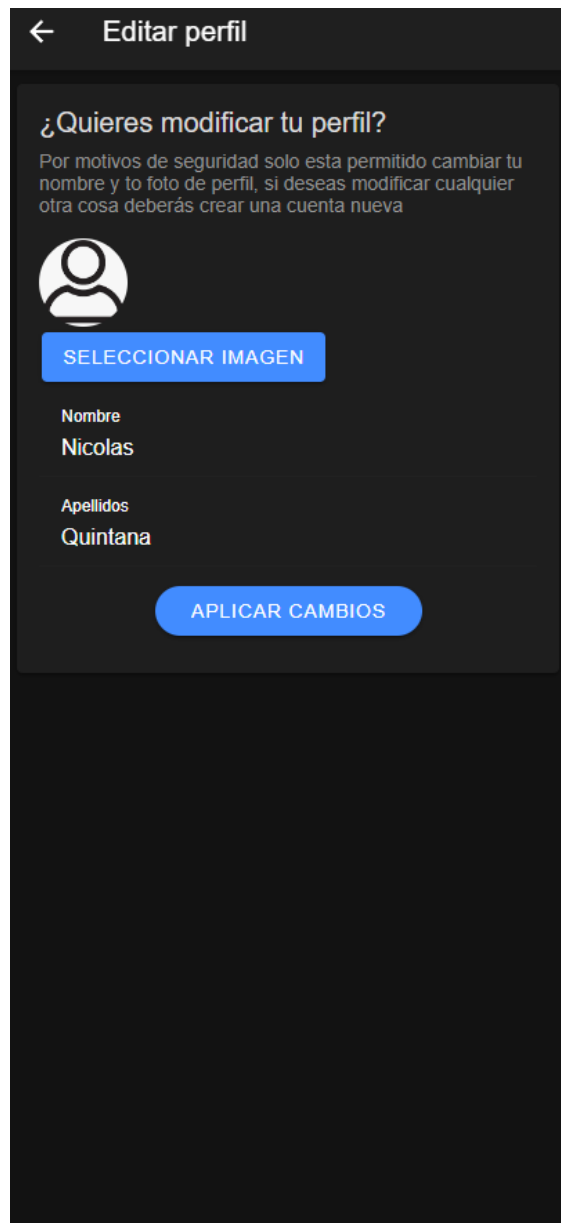
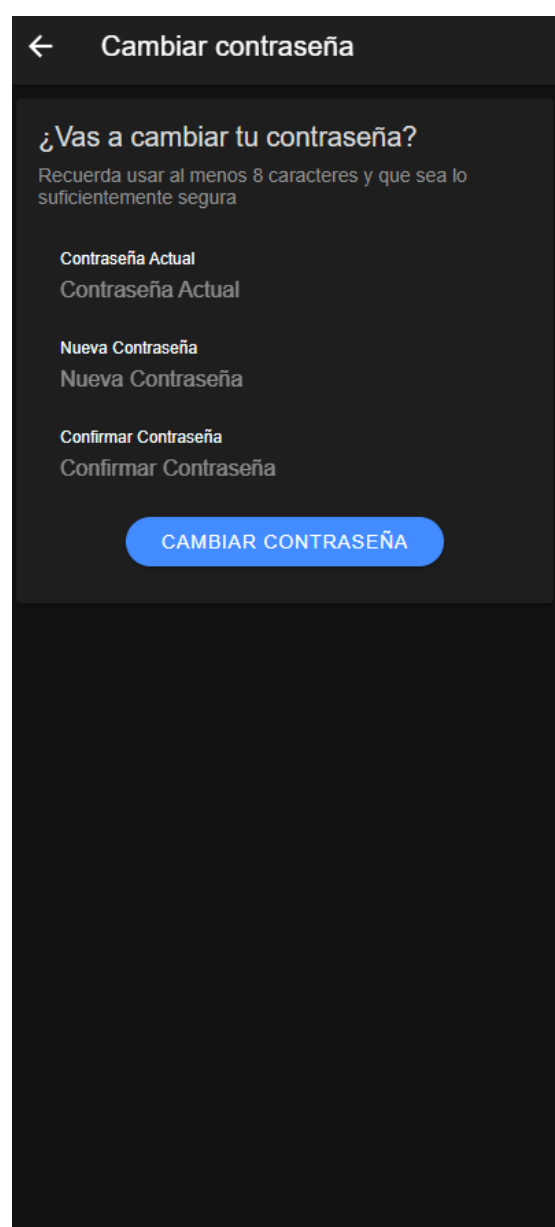
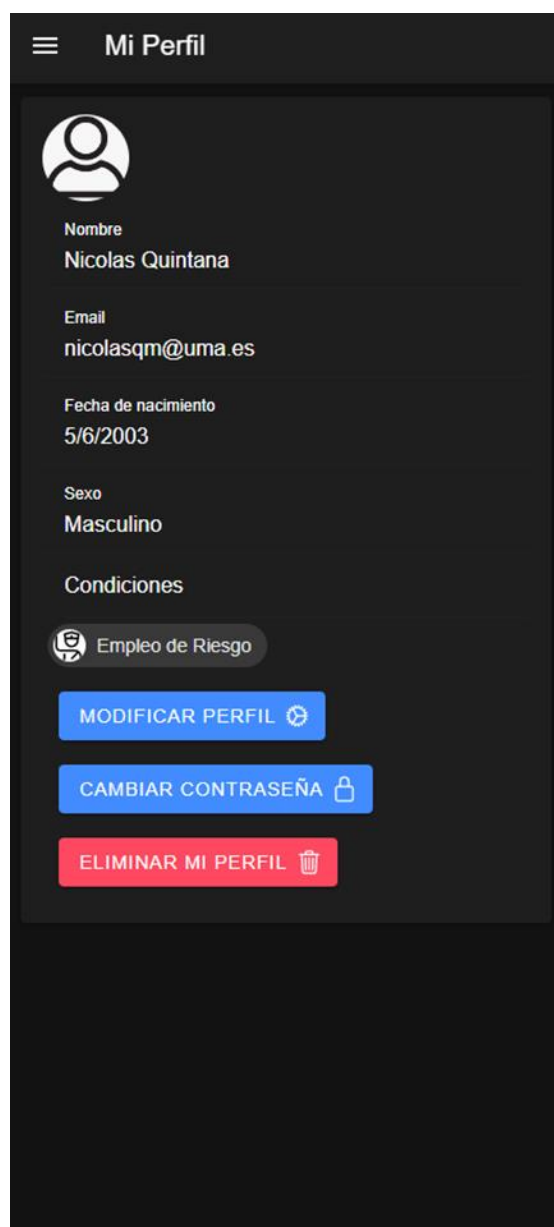


Figura A.11: Modificar Perfil

Una vez en la pantalla podemos, de la misma forma que en el registro, modificar la imagen desde la galería o la cámara y modificar el nombre y apellidos, una vez acabado, al pulsar "Aplicar Cambios" se actualizará el perfil.

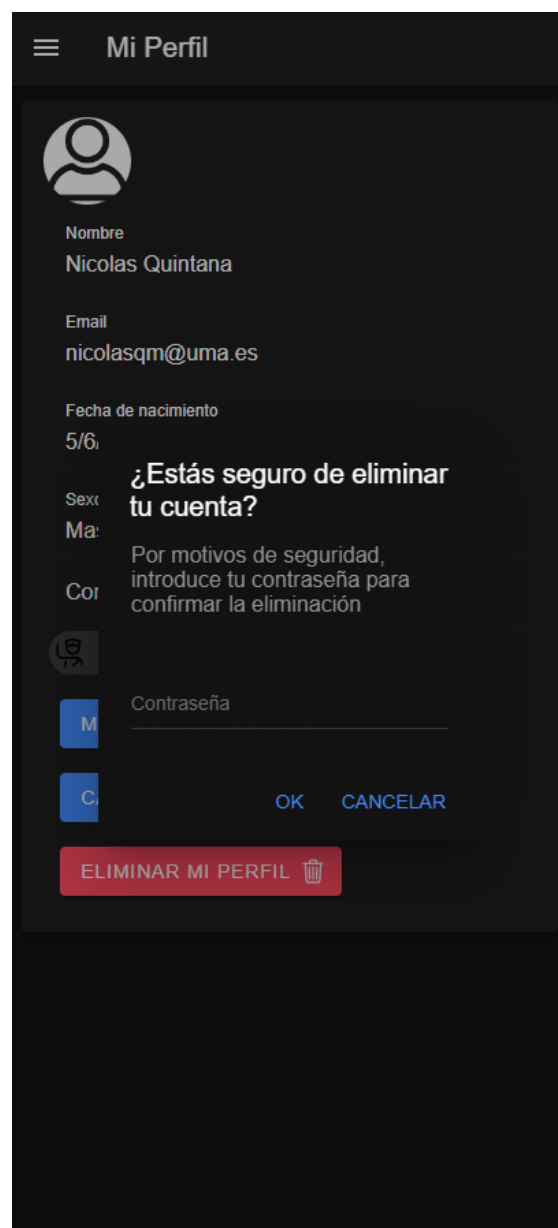
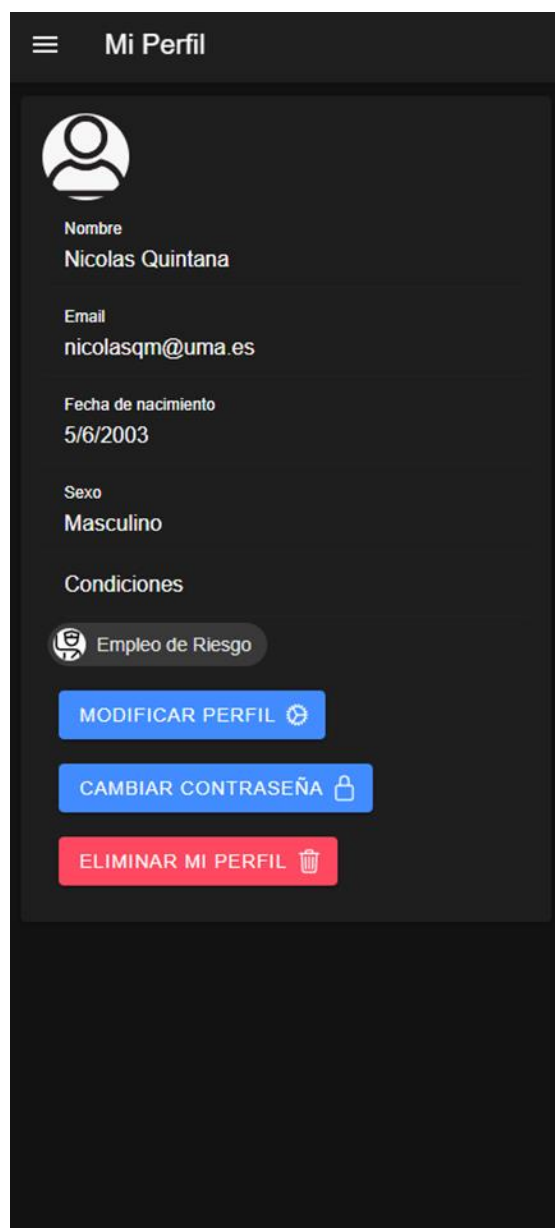
Cambiar Contraseña



Figuras A.12 y A.13: Perfil y Cambio de Contraseña

Para cambiar la contraseña, desde el perfil del usuario (Figura A.12), pulsamos "Cambiar Contraseña" y se mostrará la pantalla de cambio de perfil (Figura A.13), tras esto, es necesario introducir la contraseña actual y la nueva, que debe tener al menos 8 caracteres, tras pulsar "Cambiar Contraseña", se cambiará la contraseña del usuario.

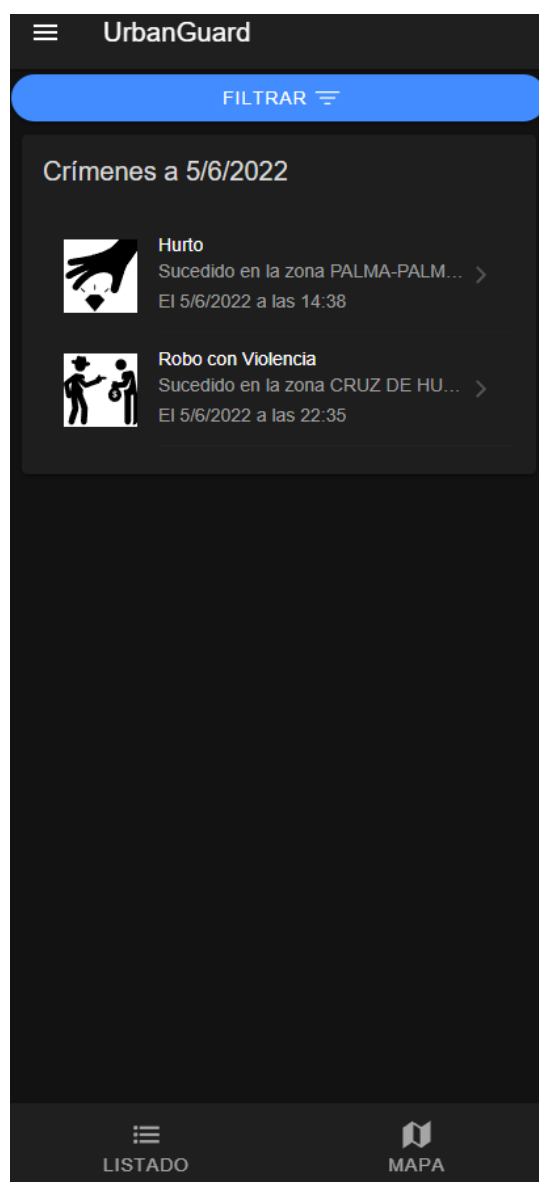
Eliminar Perfil



Figuras A.14 y A.15: Perfil Y Eliminación de Perfil

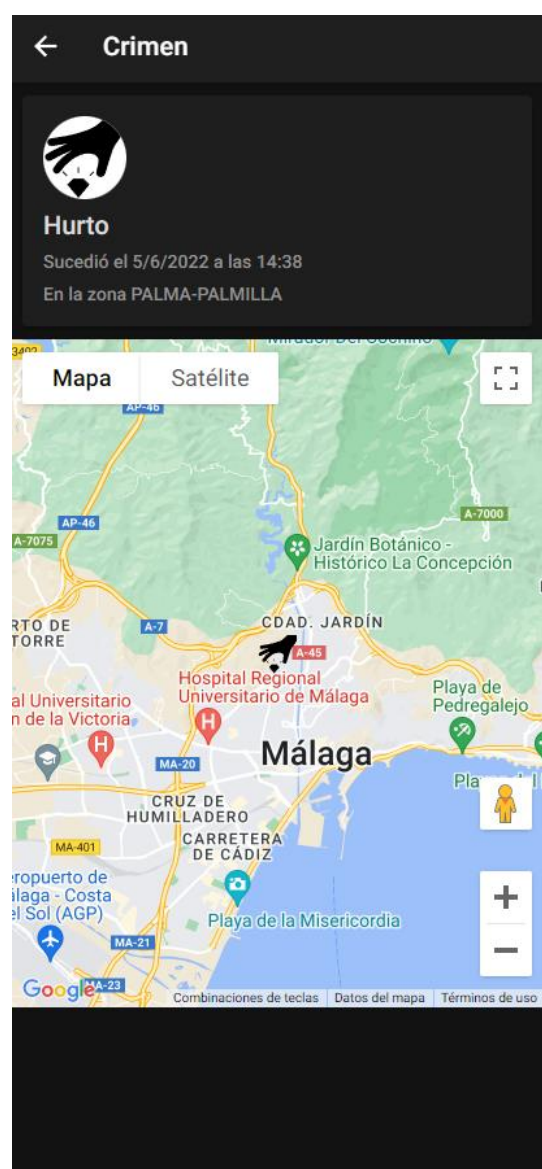
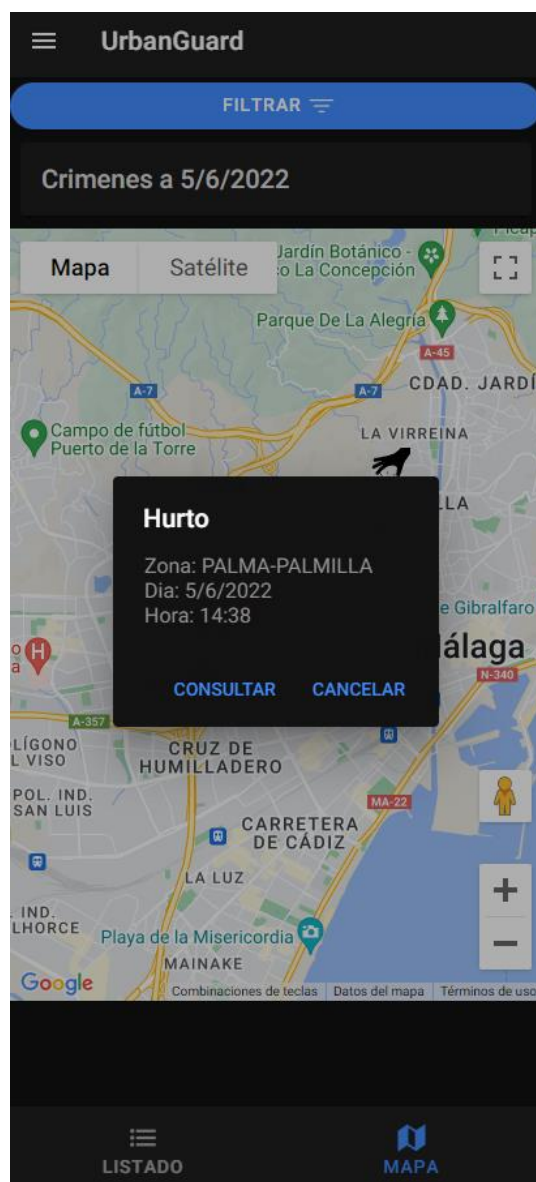
Para la eliminación del perfil, también desde el propio perfil (Figura A.14), pulsamos "Eliminar mi Perfil", se nos mostrará un diálogo (Figura A.15) que solicita la contraseña para eliminar el perfil, por motivos de seguridad. Tras introducirla y pulsar "OK", si la contraseña es correcta, se eliminará el usuario y se cerrará la sesión.

Filtrar y Visualizar Delitos



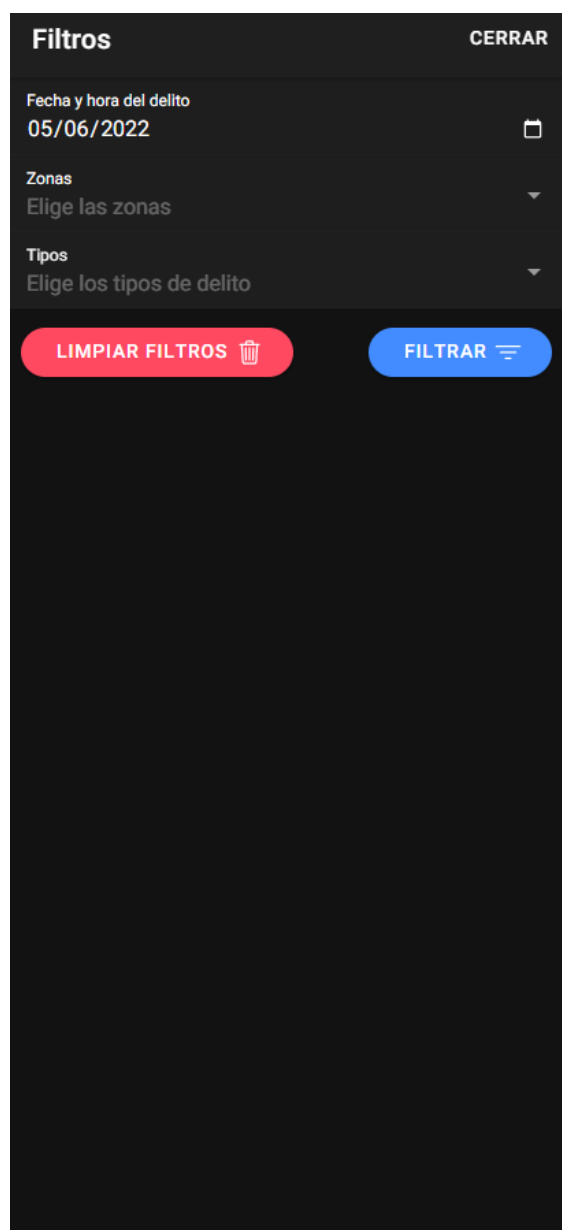
Figuras A.16 y A.17: Vista de Delitos en Listado y en Mapa

Los delitos pueden ser visualizados tanto en un listado, como en un mapa, haciendo uso del menu tabular de la zona inferior (Ver Figuras A.16 y A.17).



Figuras A.18 y A.19: Detalles de Perfil en Mapa y Vista Completa

También es posible visualizar un delito en concreto haciendo click en el objeto de la lista o, haciendo click en el icono del mapa y pulsando en consultar en el dialogo de información que aparece (Ver Figuras A.18 y A.19).



Figuras A.20 y A.21: Filtrado y Resultado de Filtrado

Para realizar el filtrado, se puede pulsar "Filtrar" y se ofrecen una serie de campos como la fecha, la zona y el tipo de delito, así como una opción para limpiar los filtros (Figura A.20). Al seleccionar una zona, la misma se muestra en el mapa para permitir al usuario visualizar la misma (Figura A.21).

Publicar un Delito

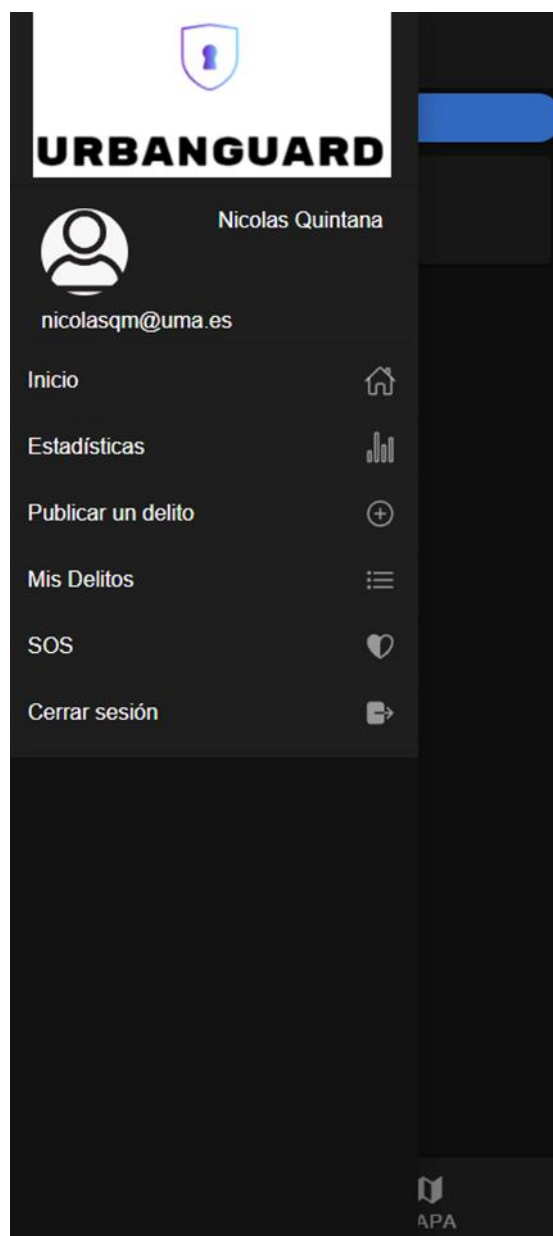


Figura A.22: Menú

Para publicar un nuevo delito, simplemente pulsamos desde el menú (Figura A.22) en "Publicar un Delito".

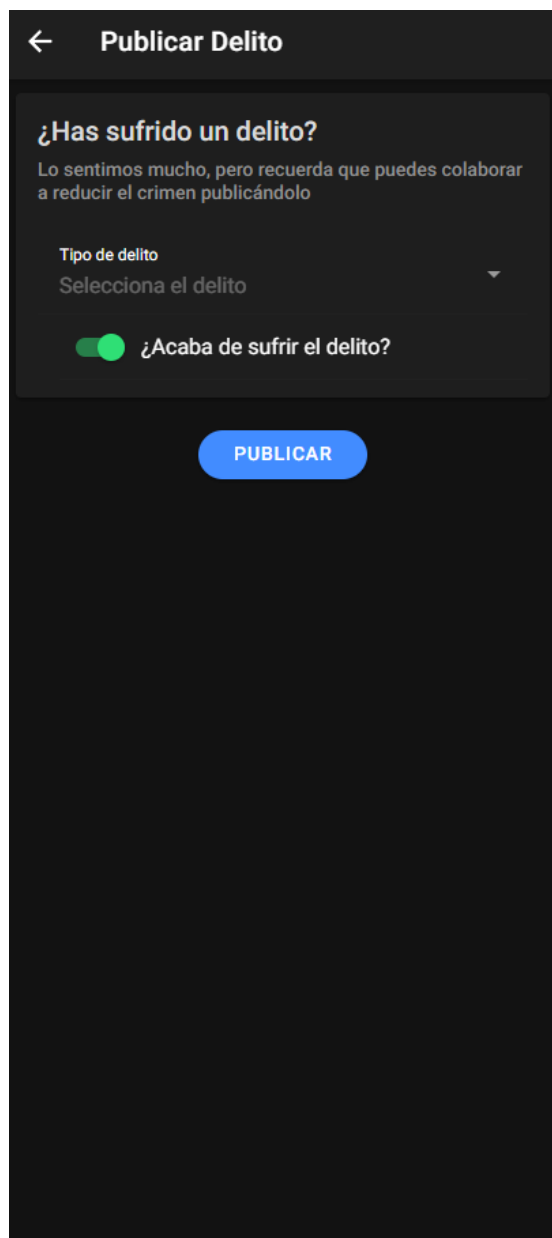


Figura A.23 y A.24: Publicación de un Delito actual o no actual

Esto hará que se muestre la vista de publicación de delito (Figura A.23), es necesario indicar el tipo de delito y se puede alternar entre acabar de sufrir el delito, de forma que se recoge la ubicación del usuario y la fecha y hora, o es posible indicar esta información de forma manual (Figura A.24). Tras pulsar "Publicar", se publicará el delito.

Eliminar Delito

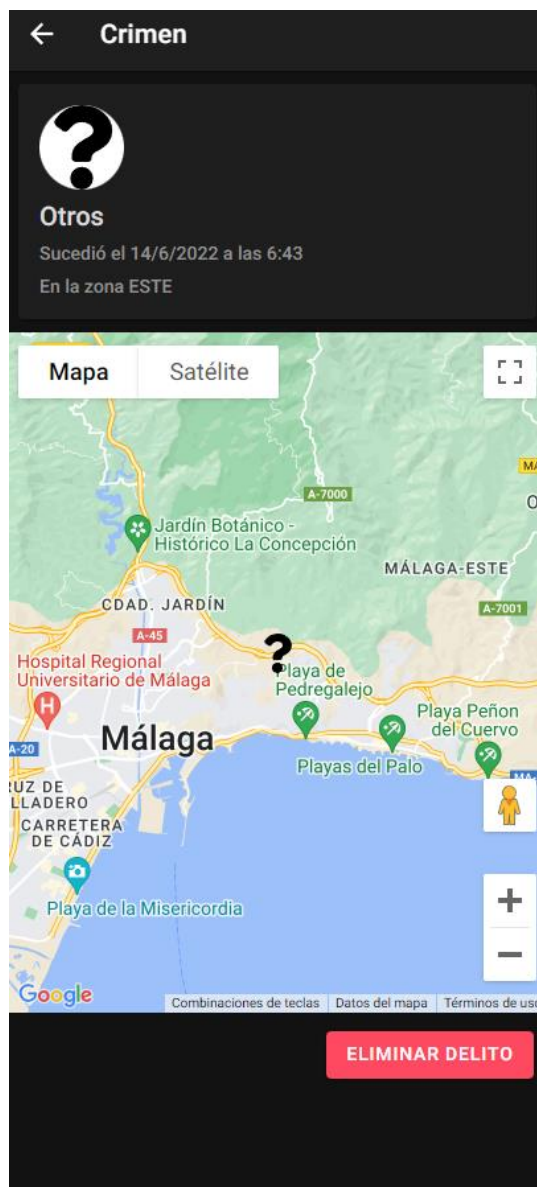


Figura A.25: Detalles de Delito

Para eliminar un delito, simplemente hay que acceder a la vista detallada del delito (Figura A.25) y hacer click en "Eliminar Delito", solo visible si eres el usuario que publicó el mismo.

Visualizar Estadísticas

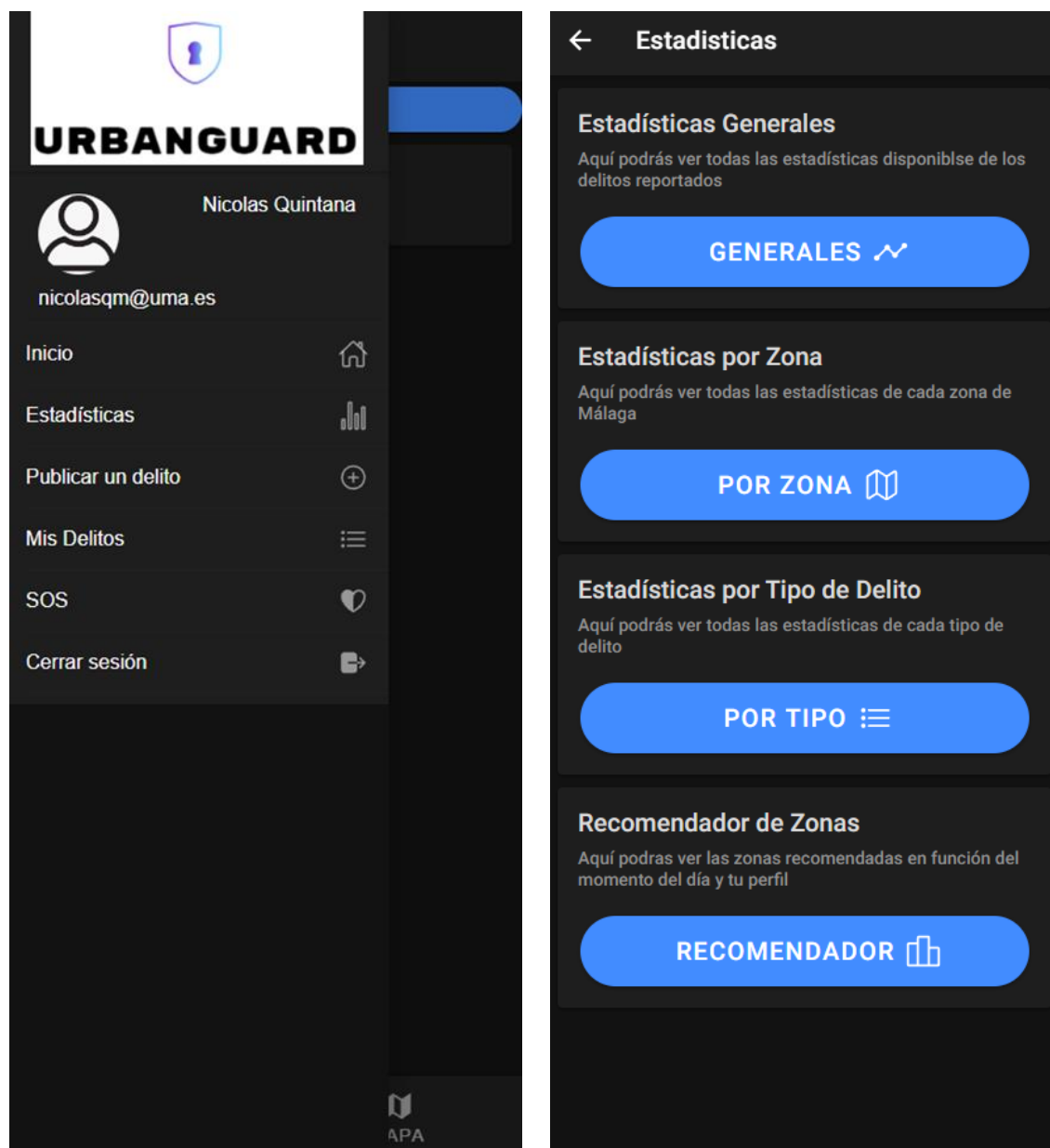


Figura A.26 y A.27: Menú y Selección de Estadísticas

Para visualizar las estadísticas, pulsamos, dentro del menú (Figura A.26), en "Estadísticas", esto mostrará la pantalla de selección (Figura A.27), de forma que podemos elegir entre estadísticas generales o por zona y delito, el recomendador aún no está integrado.

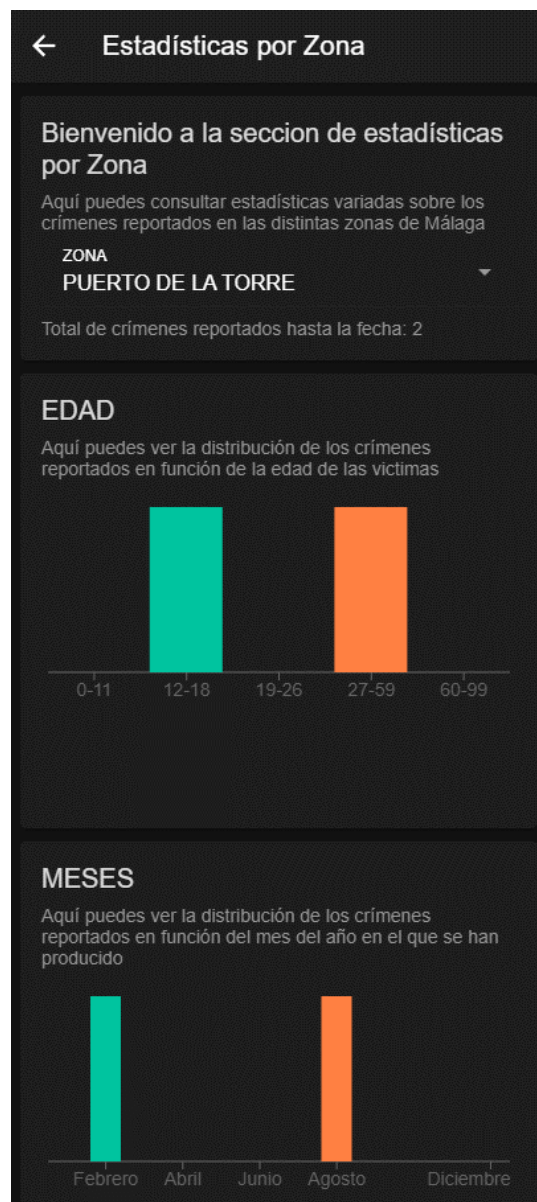


Figura A.28 y A.29: Estadísticas Generales y Específicas

De esta forma, podemos consultar las estadísticas generales (Figura A.28) y las específicas (Figura A.29), en estas últimas, podemos seleccionar la zona o el tipo de delito que queramos consultar.

Envío de SOS

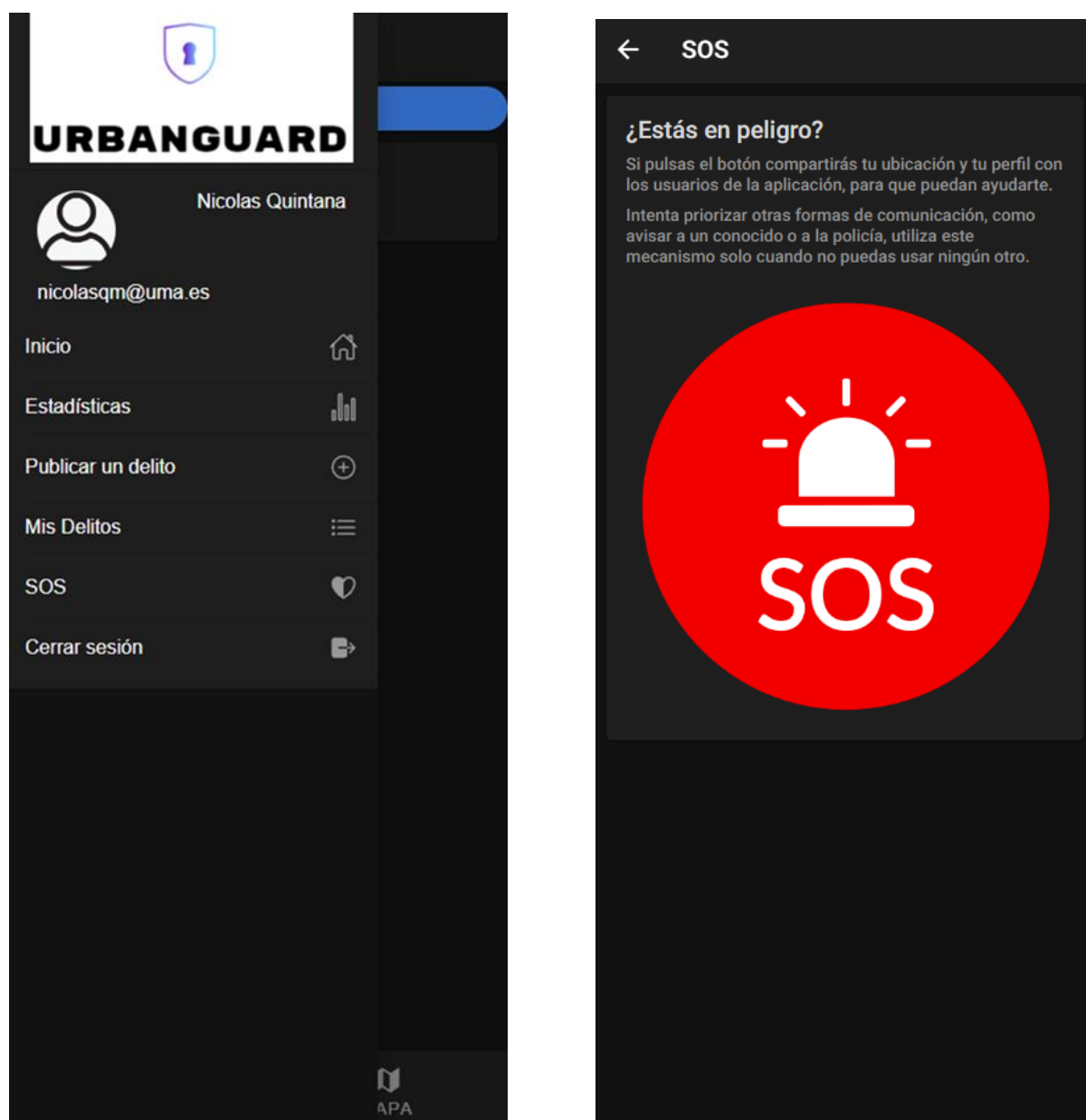


Figura A.30 Y A.31: Menú y Envío de SOS

Para el envío de señales de auxilio, desde el menú (Figura A.30), pulsamos "SOS", lo que nos muestra la pantalla con el botón de emergencia (Figura A.31), que al pulsarlo, enviará la señal al resto de usuarios.

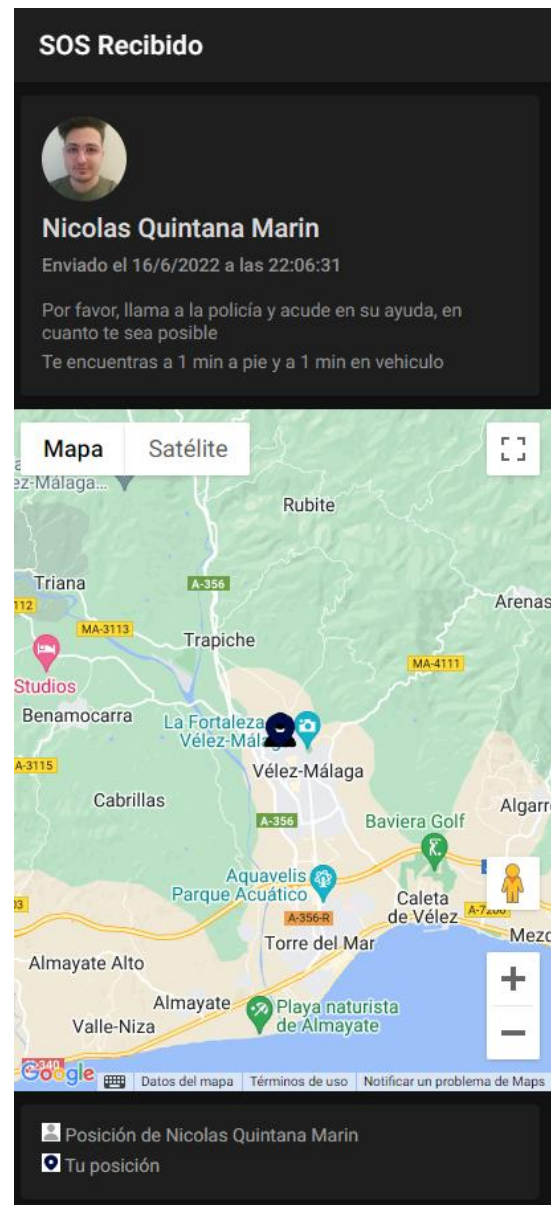
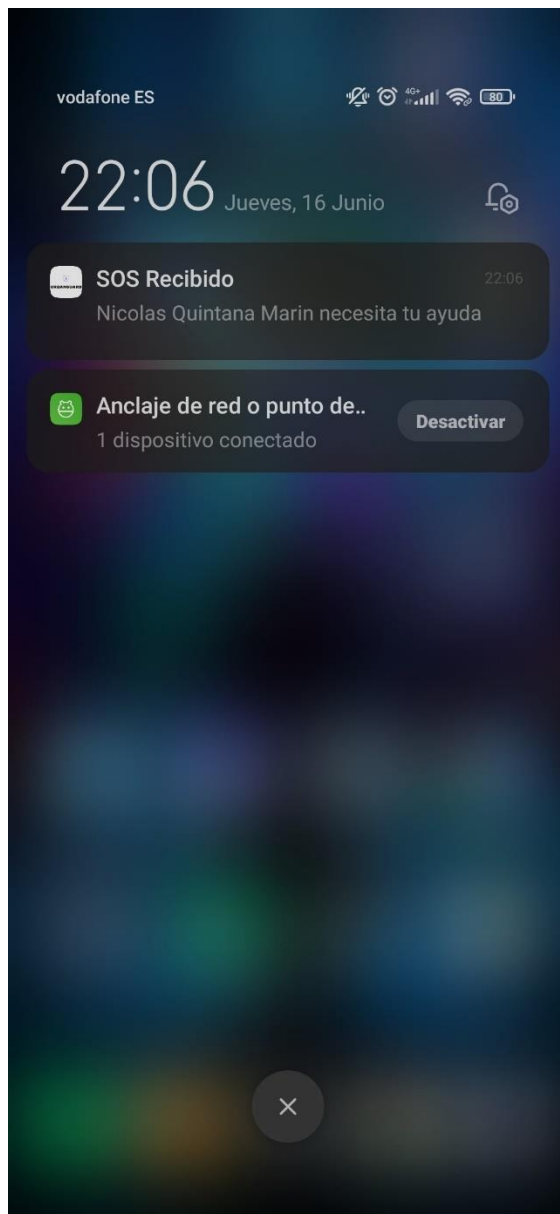


Figura A.32 y A.33: Notificación recibida y Panel de Información del SOS

El usuario recibe la señal y, al pulsar sobre ella (Figura A.32), le muestra la información de la señal, en resumen, el usuario, el momento en el que se envió, su ubicación y la de la señal y el tiempo de llegada a pie y en vehículo (Figura A.33). En el ejemplo, al realizarse desde el mismo punto, paracticamente coinciden las ubicaciones.

Apéndice B: Manual de Instalación

En este último apartado, se incluirán las instrucciones para instalar la aplicación y desplegar las funciones de Cloud Functions

Instalación en Android

Para instalar la aplicación en Android basta con mover el fichero .apk a nuestro sistema y proceder con la instalación automatizada por el sistema operativo, como se indicaba en los requisitos, la versión mínima de Android es 8.

Instalación en local para desarrollo

Para instalar el proyecto de forma local y poder desplegarlo en servidor local de desarrollo, es necesario instalar previamente:

- Ionic CLI
- NodeJS
- Npm
- Repositorio con el código

Una vez instalado todo, desde la carpeta raíz del proyecto ejecutamos el comando `npm install`, lo que instalará todas las dependencias necesarias. Tras esto, basta con ejecutar `ionic serve` para desplegar el servidor de desarrollo con la aplicación funcionando.

Adicionalmente, se puede generar el proyecto de Android Studio con un simple comando, `ionic capacitor add android`, esto generará la estructura del proyecto con todos los ficheros necesarios.

Despliegue de las funciones

Para desplegar las Cloud Functions, son necesarios:

- NodeJS
- Npm
- Firebase CLI
- Repositorio con el código

Tras configurar el CLI con la guía oficial de Firebase (disponible en https://firebase.google.com/docs/cli?hl=es#install_the_firebase_cli), simplemente, desde la carpeta raíz del proyecto, ejecutamos el comando `firebase deploy --only functions`, esto iniciará el proceso de empaquetamiento, subida y despliegue de las funciones.



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga