

---

# PROYECTO ALERT

---

**GERALD LIMA MENDIA**

**Grado en Ingeniería Informática**

**Facultad de Informática**



**TRABAJO DE FIN DE GRADO**

**Universidad Complutense de Madrid**

**Curso 2018-2019**

**DIRECTOR:**

**ANTONIO SARSA CABEZUELO**



## AGRADECIMIENTOS

En primer lugar, quiero dar las gracias a mi tutor y coordinador del proyecto, Antonio Sarasa Cabezuelo, por haber puesto su confianza en mí y por haberme guiado durante el periodo y las fases de este trabajo.

También quiero agradecer a Marta Frías y Javad Amoozegar, coordinadores del sector de desarrollo TIC en “Acción Contra El Hambre”, por ser pacientes en las reuniones y por haberme facilitado de cierta información sobre las actividades y funciones que llevan ellos y sus compañeros en el campo de trabajo, de esta forma me han puesto en perspectiva para preparar este proyecto.

Y por último dar las gracias a mi familia por estar siempre pendientes de mi labor e inquietudes, por el apoyo recibido en todo momento.

## RESUMEN

Alert es un proyecto que surgió a partir de una idea inicial como propuesta de proyecto en colaboración con una ONG, y que posteriormente se amoldó a las necesidades de esta organización externa (cliente) para cubrir una pequeña parte de los muchos proyectos que tienen en mente y algunos ya puestos en marcha. En particular, se pretende facilitar las tareas de recogida de información por parte de los miembros de dicha organización, cuyos puntos de control pueden estar localizados en distintos puntos del planeta.

Consiste en una aplicación para plataformas móvil que mejora y asiste a la transferencia y el suministro de información a portales y plataformas de datos organizacionales nuevos o existentes. Este sistema se alimentará de forma manual con los datos introducidos por el usuario final, o de forma automática mediante un script que analizará datos extraídos de una red social. Además, cuenta con un sistema de consulta de datos estadísticos de carácter público o privado.

Este sistema ayudará a la preparación y pronta o inmediata respuesta después de un incidente o desastre.

**Palabras clave:** *ONG, aplicación, script, incidente, alerta*

## ABSTRACT

Alert is a project in collaboration with an NGO. This project has been adapted to the needs of the client, in order to cover a small part of the many projects they have in mind (some of them have been already launched). This idea will help the members of the NGO, whose control points may be located in different parts of the planet, to facilitate the tasks of collecting information.

It is an application for mobile platforms which improves and assists the data flow. The information is addressed to new and updated organizational platforms, so that this system can be fed manually with the data given by the final user, or automatically through a script that will analyse data extracted from a social network. In addition, it has a public or private statistical data of all this information.

This statistical data will help to prepare a better and immediate response after an incident or disaster.

**Key Words:** *NGO, application, script, issue, alert*

# ÍNDICE

AGRADECIMIENTOS.....	3
RESUMEN .....	4
ABSTRACT .....	5
ÍNDICE DE FIGURAS .....	7
1 INTRODUCCIÓN.....	9
1.1 Motivación .....	9
1.2 Objetivos .....	10
1.3 Estructura de la memoria.....	11
1 INTRODUCTION .....	13
1.4 Motivation .....	13
1.5 Objectives.....	13
1.6 Structure of the report.....	14
2 ESTADO DEL ARTE .....	16
2.1 Herramientas Similares .....	16
3 ESPECIFICACIÓN DE LA APLICACIÓN .....	18
3.1 Historia inicial.....	18
3.2 Casos de uso.....	19
3.2.1 Módulo Gestión de usuarios .....	19
3.2.2 Módulo Gestión de alertas.....	20
3.3 Descripción detallada de los Casos de Uso .....	21
3.3.1 Módulo Gestión de usuarios .....	21
3.3.2 Módulo Gestión de alertas.....	24
4 TECNOLOGÍAS EMPLEADAS.....	28
4.1 Javascript (JS) .....	28
4.2 HTML5 y CSS3.....	28
4.3 Angular .....	28
4.4 Ionic.....	28
4.5 NPM.....	29
4.6 Firebase .....	29
4.7 Visual Studio Code.....	29
5 MODELO DE DATOS.....	30
5.1 Base de Datos no relacional .....	30
6 ARQUITECTURA DEL SISTEMA.....	34
6.1 Aplicación móvil .....	35
6.2 Aplicación web .....	35

6.3	Base de datos .....	35
7	DISEÑO E IMPLEMENTACIÓN .....	36
7.1	Diseño.....	36
7.2	Implementación .....	36
8	CONCLUSIONES Y TRABAJO FUTURO .....	50
8.1	Conclusiones.....	50
8.2	Trabajo futuro .....	51
8	CONCLUSIONS AND FUTURE WORK.....	53
8.1	Conclusions .....	53
8.2	Future work.....	54
	BIBLIOGRAFÍA.....	56
	ANEXOS .....	57
	ANEXO I – GUÍA DE USO .....	57
	APP MÓVIL .....	57
	APLICACIÓN WEB .....	72
	ANEXO II – GUÍA DE INSTALACIÓN .....	79
	Instalación de la app móvil.....	79
	Instalación de la aplicación web.....	79

## ÍNDICE DE FIGURAS

Figura 1:	Diagrama de casos de uso del actor "administrador" .....	19
Figura 2:	Diagrama de casos de uso del actor "usuario" .....	20
Figura 3:	Importación de algunas librerías de código abierto para un proyecto Ionic-angular – Angular repo, angularfire2 y Firebase .....	30
Figura 4:	Métodos principales de operaciones CRUD con Firebase-Firestore.....	31
Figura 5:	Colección de Usuarios y un Documento de un usuario concreto.....	32
Figura 6:	Colección de Alertas y una serie de Documentos y sus datos.....	32
Figura 7:	Arquitectura que sigue el sistema. Aplicación móvil y web .....	34
Figura 8:	Diagrama de módulos de la aplicación móvil.....	35
Figura 9:	Detalle de la vista del Menú principal .....	37
Figura 10:	Fragmento de código implementado del Menú principal.....	38
Figura 11:	Detalles de algunas vistas del formulario de alerta – parte1.....	39
Figura 12:	Detalles de algunas vistas del formulario de alerta – parte2 .....	39
Figura 13:	Fragmento de código de la implementación del formulario de alertas.....	40
Figura 14:	Detalle de la vista de Login y parte de la implementación de su modelo.....	41
Figura 15:	Fragmento de código de la implementación HTML de Login .....	42
Figura 16:	Vistas de la funcionalidad – Modificar perfil de usuario.....	42
Figura 17:	Fragmento de código del modelo – Modificar perfil.....	43

Figura 18: Fragmento de código HTML – Modificar perfil .....	44
Figura 19: Vistas de la lista de alertas del usuario y la edición de una alerta .....	45
Figura 20: Fragmento de código HTML de las alertas del usuario.....	46
Figura 21: Fragmento de código del modelo de alertas del usuario, editar y actualizar la alerta .....	46
Figura 22: Gráficos de las Alertas más recientes, filtradas por la cantidad introducida .....	47
Figura 23: Gráficos de la frecuencia con la que un usuario utiliza la app .....	47
Figura 24: Gráficos de las Alertas que se han registrado entre las fechas elegidas.....	48
Figura 25: Gestión de todos los usuarios en la base de datos .....	49
Figura 26: Gestión de todas las alertas en la base de datos .....	49
Figura 27: Pantalla – Iniciar sesión.....	57
Figura 28: Pantalla – Crear cuenta .....	58
Figura 29: Pantalla – Menú principal .....	59
Figura 30: Pantalla – Mapa con localización GPS .....	59
Figura 31: Pantalla – Más opciones.....	60
Figura 32: Pantalla – Guía de usuario .....	60
Figura 33: Pantalla – Barra lateral de opciones del menú .....	61
Figura 34: Pantalla – Formulario de alerta (slide_1) .....	62
Figura 35: Pantalla – Formulario de alerta (slide_2).....	62
Figura 36: Pantalla – Formulario de alerta (slide_3) .....	63
Figura 37: Pantalla – Formulario de alerta (slide_4).....	63
Figura 38: Pantalla – Formulario de alerta (slide_5).....	64
Figura 39Pantalla – Formulario de alerta (slide_6).....	64
Figura 40: Pantalla – Formulario de alerta (slide_7) .....	65
Figura 41: Pantalla – Formulario de alerta (slide_8).....	66
Figura 42: Pantalla – Formulario de alerta (slide_9).....	67
Figura 43: Pantalla – Formulario de alertas (slide_9). Progreso de carga de imagen .....	67
Figura 44: Pantalla – Formulario de alerta (slide_10) .....	68
Figura 45: Pantalla – Modificar perfil.....	69
Figura 46: Pantalla – Visualizar Alertas .....	70
Figura 47: Pantalla – Modificar alerta.....	71
Figura 48: Pantalla – Iniciar sesión.....	72
Figura 49: Pantalla – Ver y editar perfil.....	73
Figura 50: Pantalla – Menú lateral .....	73
Figura 51: Pantalla – Selección de una las distintas opciones.....	74
Figura 52: Pantalla – Interacción sobre las gráficas.....	74
Figura 53: Pantalla – Exportar la gráfica actual localmente y en formato imagen.....	75
Figura 54: Pantalla – Tabla de gestión de usuarios (vista general) .....	75
Figura 55: Pantalla – Tabla de gestión de usuarios(vista seleccionar usuario).....	76
Figura 56: Pantalla – Tabla de gestión de usuarios (más opciones).....	76
Figura 57: Pantalla – Tabla de gestión de usuarios (con filtro de).....	77
Figura 58: Pantalla – Tabla de gestión de usuarios (orden por columnas).....	77
Figura 59: Pantalla – Tabla de gestión de alertas.....	78

# **1 INTRODUCCIÓN**

Como primer capítulo de este documento se va a exponer la principal motivación que ha llevado a realizarse y desarrollarse este proyecto, así también se van a quedar reflejadas sus objetivos.

## **1.1 Motivación**

En la actualidad existe aún un número considerable de países o zonas de países del tercer mundo en el que sus habitantes pasan una situación desesperante de necesidad, sin los recursos más básicos como es el agua y alimentos. Pues bien, existen organismos no gubernamentales que intentan reducir por todos los medios que tienen esta agravante situación que están viviendo las personas que tuvieron la mala fortuna de nacer en países no muy prósperos.

Una de las propuestas de proyecto que se consideró y se puso sobre la mesa, de algún modo, era el hecho de aportar algo a estos organismos no gubernamentales, algo que les pueda ayudar a realizar su trabajo y desempeño con esos países que necesitan de cualquier tipo de apoyo.

De esta forma este sistema haría que el trabajo de estos organismos sea un poco más fácil a la hora de tomar acciones preventivas para ciertos reportes de alertas que puedan producirse en zonas críticas del mundo.

## 1.2 Objetivos

El principal objetivo de este proyecto era desarrollar un sistema que permita gestionar los datos más pertinentes de reporte de incidencias para abordar los problemas centrales de facilitar el flujo de información en misiones (capital y bases), sede y delegaciones.

De tal forma, se definieron los siguientes objetivos específicos:

- Implementar una aplicación que englobe y permita la gestión de alertas por parte de profesionales en las distintas misiones que puedan tener.
- Desarrollar un sistema que sea capaz de extraer, analizar y reportar información con respecto a incidencias, alertas reportadas en las redes sociales.
- Crear un sistema donde se pueda visualizar los datos estadísticos más relevantes sobre la situación de reportes, perpetradores, víctimas, localizaciones, etc.

### **1.3 Estructura de la memoria**

El proyecto tiene una memoria que se estructura de la siguiente forma:

#### **Capítulo 1 – Introducción**

Primeramente, en el capítulo uno se realiza una breve introducción del proyecto y la motivación que ha llevado a su desarrollo. Además, se especifica cual es la estructura de memoria que sigue el actual documento.

#### **Capítulo 2 – Estado del arte**

En el siguiente capítulo dos se describen las distintas herramientas que estén en el mercado hoy en día y que tengan funcionalidades semejantes a esta herramienta.

#### **Capítulo 3 – Especificación de la aplicación**

Las funcionalidades más relevantes del sistema se exponen en el capítulo tres, los cuales están estructurados en distintos módulos, que son: Alertas, Gestores, Reportadores, visualización de datos. Y se incluyen algunos diagramas de casos de uso, tanto de parte de la aplicación móvil como de la parte web.

#### **Capítulo 4 – Tecnologías empleadas**

En el capítulo cuatro se dan a conocer las tecnologías que se ha utilizado para el desarrollo de este sistema, así como sus principales características.

#### **Capítulo 5 – Modelo de datos**

En el siguiente capítulo cinco se establece la estructura relacional que tiene el sistema, este modelo no relacional y su implementación quedan reflejados en unos esquemas y/o figuras que se exponen una visión global.

#### **Capítulo 6 – Arquitectura del sistema**

En el capítulo seis se describe los patrones, reglas y la organización estructural que sigue el sistema.

## **Capítulo 7 – Diseño e Implementación**

La parte de diseño del sistema, patrones de diseño empleados, herramientas de desarrollo y la funcionalidad quedan reflejados en el capítulo siete. Se detallan la relación entre los distintos módulos y las librerías utilizadas.

## **Capítulo 8 – Conclusión y Trabajo futuro**

Este último capítulo resume las conclusiones sobre el trabajo, teniendo en cuenta la experiencia del desarrollador mediante el proceso de creación, pruebas y funcionalidad del producto final. Además, se detallan las posibles mejoras que podrían incorporarse en el proyecto con la finalidad de potenciar sus funcionalidades y uso por parte de la experiencia de usuario.

## **Bibliografía, Anexos I y II**

Finalmente, la bibliografía y los anexos se incluyen al final de la memoria.

# 1 INTRODUCTION

In the first chapter, we will see the inner drive of this project and a brief overview about the main aims.

## 1.4 Motivation

Nowadays, there are still a huge amount of undeveloped countries where people are not able to get the basic resources to survive. Fortunately, there are also a few non-governmental organizations that try to help with this situation by all means.

One of the project proposals was contribute somehow to these non-governmental organizations, something that can help them carry out their work and performance with those countries that need any kind of support.

In this way, this application would make the work of these organizations a little easier when they have to take preventive actions for certain alert reports that may occur in critical areas of the world.

## 1.5 Objectives

The main objective of this project was to develop a system that allows managing the most relevant incident report data to address the issues of facilitating the flow of information in missions, headquarters and delegations.

Because of this, the following goals were required:

- Develop an application which allows the management of alerts by professionals in the different missions they may have.
- Create a system that is able to extract, analyse and report information regarding incidents and alerts reported on social networks.
- Set up a system where you can visualize the most relevant statistical data on the status of reports, victims, locations, etc.

## **1.6 Structure of the report**

The project's documentation is structured as follows:

### **Chapter 1 – introduction**

First of all, in the first chapter we can see a brief overview of the project and the motivation which has driven to the development. Besides, it specifies the structure of the documentation

### **Chapter 2 - State of the art**

The chapter two describes the different tools that are on the market today and that have similar features to this application.

### **Chapter 3 - Application Specification**

The most relevant features of the system are presented in chapter three, which are structured in different modules, which are: Alerts, Managers, Reporters, data visualization. Besides, some mobile and web Diagrams of Use Cases are included.

### **Chapter 4 – Technologies**

Chapter four describes the technologies that have been used for the development of this system, as well as its main features.

### **Chapter 5 - Data Model**

The following chapter establishes the relational structure that the system has. This non-relational model and its implementation are reflected in some schemes and / or figures that expose a global vision.

### **Chapter 6 - System Architecture**

Chapter six describes the patterns, rules and structural organization that the system follows.

## **Chapter 7 - Design and Implementation**

The part of system design, design patterns used, development tools and functionality are reflected in chapter seven. The relationship between the different modules and the libraries used is detailed.

## **Chapter 8 - Conclusion and Future Work**

The last chapter summarizes the conclusions about the work, taking into account the experience of the developer through the process of creation, testing and functionality of the final product. In addition, the possible improvements that could be incorporated into the project are detailed in order to enhance its functionalities and use by the user experience.

## **Bibliography, Annexes I and II**

Finally, the bibliography and annexes are included at the end of this document.

## 2 ESTADO DEL ARTE

En este apartado se va a considerar algunas características más importantes de las herramientas que existen actualmente en el mercado y que son semejantes al software desarrollado en este proyecto.

El proyecto tiene una herramienta en el que básicamente integra un sistema de gestión de información tanto de usuarios y alertas que. Alertas que reportan los usuarios y que estos datos serán gestionados por un administrador.

Siguiendo este ámbito, en la actualidad existen algunas herramientas de gestión de datos y solicitudes, y reportes de incidencias que se pueden asemejar a las características del sistema desarrollado.

### 2.1 Herramientas Similares

#### **Alert Cops [1]**

Es una herramienta de reporte de avisos e incidencias que lo puede utilizar un ciudadano que quiera notificar a la policía nacional, para que ellos puedan actuar acorde la situación. Principales características:

- Envío de alertas sobre acciones de vandalismo, agresiones, robos, etc.
- Contacto a través de un chat para reportar algún delito.
- Recepción de avisos en la localización actual del usuario.
- Mapa GPS con la situación actual del usuario.

#### **Disaster Alert [2]**

Se trata de un sistema de detección de riesgos de desastres que informa a los usuarios sobre los desastres que se ha producido en muchas partes del mundo. Principales características:

- Actualización constante de alertas de desastres que están activos.
- Alertas de peligros según la localización.
- Mapa interactivo que muestra diferentes peligros que puedes encontrarte en ciertos países del mundo.
- Mapeo de superposiciones de la densidad de la población en un determinado país.

### **Spotlight – Incident Reporting [3]**

Se trata de un sistema de reporte de todo tipo de alertas de incidentes en tiempo real, como reporte de personas desaparecidas, atención médica para un grave accidente ocurrido, etc. Entre las características principales destacan:

- Notificaciones instantáneas sobre las alertas suscritas.
- Generación y visualización de documentos de reportes de incidentes.
- Compartir la localización actual dentro un chat grupal.
- Interactuar en un chat grupal para compartir documentos o imágenes sobre incidentes que ocurrieron.

### 3 ESPECIFICACIÓN DE LA APLICACIÓN

En este capítulo se van a englobar y desglosar los distintos casos de uso que se han especificado para esta herramienta. Pero antes se hará una breve introducción de la historia inicial.

#### 3.1 Historia inicial

“Alert” es una herramienta en la que los usuarios podrán informarse, reportar (manualmente) una incidencia y visualizar datos sobre las principales catástrofes naturales, epidemias, conflictos armados, etc., que se hayan producido o estén próximas a producirse.

Desde la interfaz el usuario tendrá acceso a su lista de alertas, mostradas en orden alfabético ascendentemente, en las que podrá gestionar sus alertas ya creadas. Además, podrá gestionar también su perfil y dar de alta una nueva alerta.

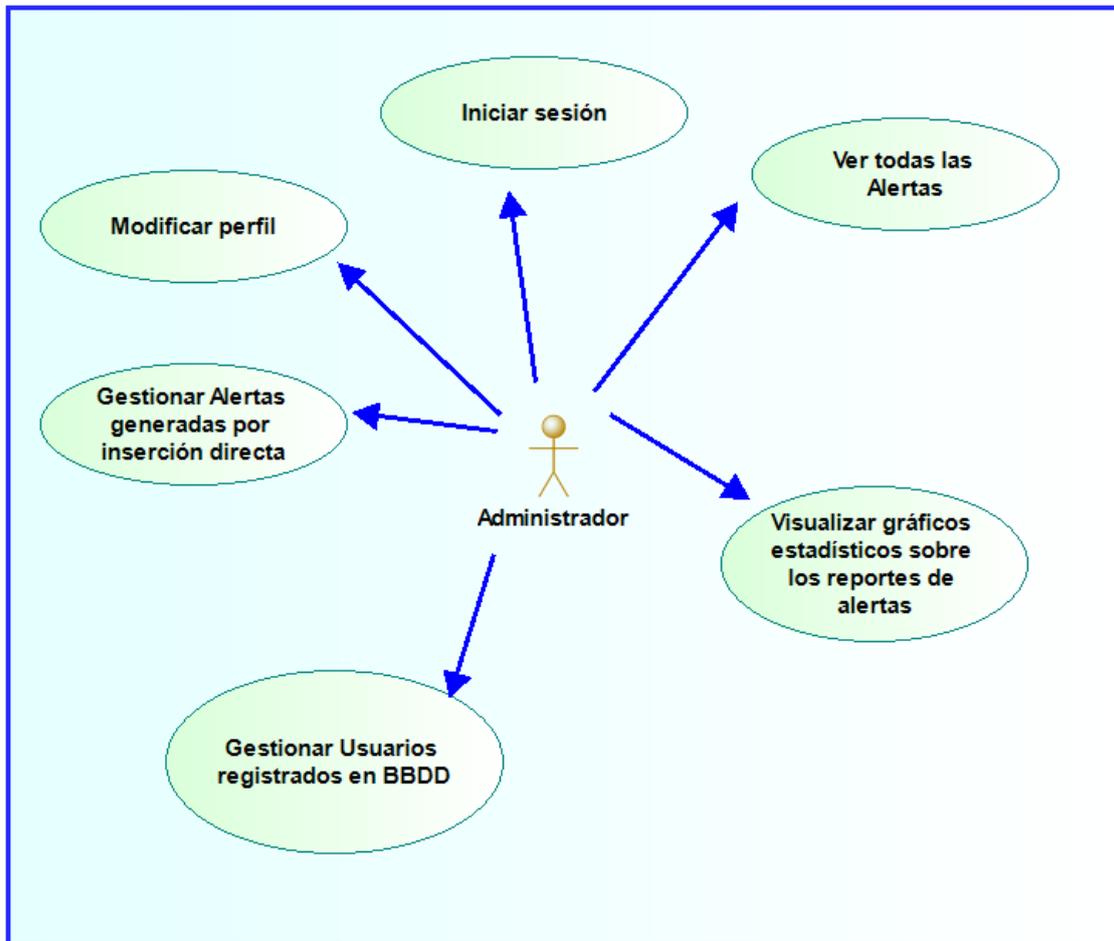
Por otro lado, existe interfaz web en el que se podrá acceder para obtener datos estadísticos de lo que está ocurriendo en la base de datos.

De igual manera, el administrador podrá activar la opción de generar alertas desde su panel de acciones, los cuales provienen de las redes sociales (más específicamente de twitter), y que se guardarán en la base de datos, así tendrán como responsable de los reportes al administrador.

## 3.2 Casos de uso

### 3.2.1 Módulo Gestión de usuarios

Toda la funcionalidad con respecto a la gestión de los usuarios se encuentra en este módulo, y para tener una visualización general de los casos de uso se ilustra la siguiente figura:



*Figura 1: Diagrama de casos de uso del actor "administrador"*

Lista de casos de uso

- 1.1. UC – Mostrar información de la App
- 1.2. UC – Mostrar Ayuda
- 1.3. UC – Mostrar Contacto
- 1.4. UC – Registrar/Alta Usuario
- 1.5. UC – Iniciar sesión
- 1.6. UC – Modificar perfil
- 1.7. UC – Dar de baja a un usuario (web)

### 3.2.2 Módulo Gestión de alertas

Las acciones que se pueden realizar con las alertas y su gestión se describen en este módulo. Como se ilustra en la figura 2, se ve a grandes rasgos las acciones de los actores que intervienen en cada caso de uso.

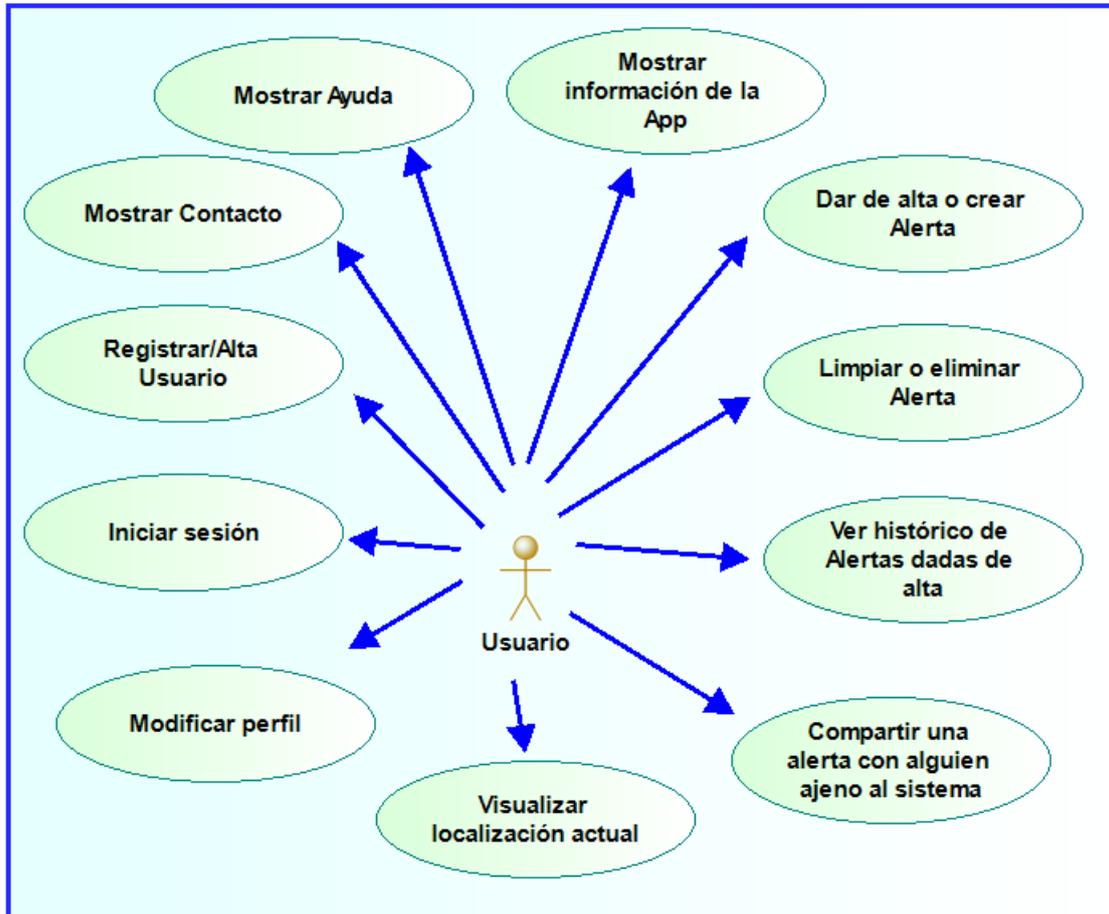


Figura 2: Diagrama de casos de uso del actor "usuario"

#### Lista de casos de uso

- 1.1. UC – Dar de alta una alerta
- 1.2. UC – Ver histórico de Alertas
- 1.3. UC – Limpiar o eliminar las Alertas
- 1.4. UC – Compartir una alerta con alguien ajeno al sistema
- 1.5. UC – Eliminar cualquier alerta (web)
- 1.6. UC – Ver gráficos estadísticos del estado de la base de datos (web)

### 3.3 Descripción detallada de los Casos de Uso

#### 3.3.1 Módulo Gestión de usuarios

##### 1.1. UC – Mostrar información de la App

Actores:

Usuario

Objetivo:

Visualizar la información de la aplicación.

Entrada:

Seleccionar el panel "Más opciones".

Seleccionar el enlace "Acerca de".

Salida:

Información detallada sobre la aplicación.

Secuencia normal:

- I. El usuario accede a la aplicación.
- II. El usuario selecciona el panel inferior "Más opciones".
- III. Se muestra la respectiva información.

##### 1.2. UC – Mostrar Ayuda

Actores:

Usuario

Objetivo:

Visualizar información de ayuda sobre la aplicación.

Entrada:

Seleccionar el panel "Más opciones".

Seleccionar el enlace "Guía de uso".

Salida:

Opciones en una especie de guía con información útil de la aplicación para su correcto uso.

Secuencia normal:

- I. El usuario accede a la aplicación.
- II. El usuario selecciona el panel inferior "Más opciones".
- III. El usuario selecciona el enlace "Guía de uso".
- IV. Se muestra información de con una guía útil de la aplicación.

##### 1.3. UC – Mostrar Contacto

Actores:

Usuario

Objetivo:

Visualizar información de contacto.

Entrada:

Seleccionar el panel "Más opciones".

Visualizar el apartado "Contacto".

Salida:

Detalles de la información de contacto para el soporte o para sugerencias de mejora de la aplicación.

Secuencia normal:

- I. El usuario accede a la aplicación.
- II. El usuario selecciona el panel inferior "Más opciones".
- III. El usuario visualiza el apartado "Contacto".
- IV. Se muestra opciones de contacto para contactar con el gestor de la aplicación.

**1.4. UC – Registrar/Alta Usuario**

Actores:

Usuario

Objetivo:

Registro de un nuevo usuario.

Entrada:

Seleccionar "registrarse".

Datos personales del usuario: nombre completo, contraseña, e-mail, teléfono, ciudad.

Salida:

El usuario se agrega en la BBDD y ya accede a la aplicación.

Secuencia normal:

- I. El usuario selecciona el botón " Registrarse".
- II. El usuario rellena los campos obligatorios necesarios para el registro.
- III. Se registra como nuevo usuario (cliente) en la BBDD.
- IV. Se muestra un mensaje de bienvenida.

Secuencia alternativa:

- II. A.) El nombre del usuario ya se encuentra en la base de datos y se le notifica que no puede crear una nueva cuenta con el nombre de usuario introducido porque este está registrado por otro usuario, y podrá volver a introducir nuevamente los datos requeridos.
- II. B.) El formato de algún campo obligatorio no es correcto y se le notifica que no puede finalizar el proceso hasta que corrija los errores.

### **1.5. UC – Iniciar sesión**

Actores:

Usuario, Administrador.

Objetivo:

Acceder a la aplicación.

Entrada:

Rellenar los campos de inicio de sesión con los datos de autenticación (nombre de usuario/mail y contraseña).

Salida:

Acceso a la aplicación (pantalla principal).

Secuencia normal:

- I. El usuario selecciona el botón "Iniciar sesión".
- II. El usuario introduce los campos de nombre de usuario y contraseña.
- III. Accede a la página de inicio de la aplicación.

Secuencia alternativa:

- II. A.) Algún dato es incorrecto y se le notifica que no puede acceder a la aplicación hasta corregir los errores de credenciales.

### **1.6. UC – Modificar perfil**

Actores:

Usuario, Administrador.

Objetivo:

Cambiar información del perfil de usuario.

Entrada:

Seleccionar el menú "Perfil".  
Seleccionar "editar perfil"  
Nuevos datos para reemplazar los anteriores.

Salida:

Datos actualizados del usuario.

Secuencia normal:

- I. El usuario selecciona el menú "Perfil".
- II. El usuario selecciona la opción "Editar perfil".
- III. El usuario introduce los nuevos datos que quiere cambiar en los respectivos campos.
- IV. El usuario selecciona el botón "Actualizar" para finalizar los cambios.
- V. Se muestra un mensaje de cambios realizados correctamente.

Secuencia alternativa:

- V. A.) El formato de algún campo no es correcto y se le notifica que no puede finalizar el proceso hasta que corrija los errores.

### **1.7. UC – Dar de Baja a un usuario (web)**

Actores:

Administrador

Objetivo:

Dar de baja a un usuario registrado.

Entrada:

Seleccionar del menú lateral "Gestión de contenidos".

Seleccionar la opción "Gestión de usuarios".

Seleccionar un usuario de la tabla de usuarios.

Salida:

El usuario queda dado de baja en la BBD de la aplicación.

Secuencia normal:

- I. El administrador selecciona del menú lateral "Gestión de contenidos".
- II. El administrador selecciona "Gestión de usuarios"
- III. El administrador selecciona a un usuario de la tabla de usuarios.
- IV. El administrador selecciona el botón "Eliminar perfil".
- V. Se muestra un mensaje de advertencia sobre la baja del usuario.
- VI. El administrador selecciona el botón "Aceptar" para finalizar el proceso.
- VII. Los datos del usuario seleccionado son eliminados de la BBDD.

### 3.3.2 Módulo Gestión de alertas

#### **1.1. UC – Dar de alta una Alerta**

Actores:

Usuario

Objetivo:

Añadir manualmente una nueva alerta a la BBDD de la aplicación.

Entrada:

Seleccionar del menú lateral "Home".

Seleccionar el icono "Alert report".

Datos de la nueva alerta: nombre, descripción, nivel de impacto, localización, etc.

Seleccionar la opción "enviar alerta".

Salida:

Se crea una nueva alerta para la aplicación.

Secuencia normal:

- I. El usuario selecciona el menú "Home".
- II. El usuario selecciona la opción "Alert report".
- III. Se muestra un formulario con una serie de slides para rellenar los datos de la nueva alerta.

- IV. El usuario rellena los campos del formulario (Nombre, HQ al que pertenece el usuario, nivel de impacto, personas afectadas, localización, etc.)
- V. El usuario selecciona la opción "Enviar alerta".
- VI. Se muestra un mensaje de confirmación de la nueva alerta añadida.
- VII. Se redirige al usuario al menú principal "Home".

Secuencia alternativa:

- V. El usuario selecciona la opción "cancelar" y vuelve al menú principal de alertas.

### **1.2. UC – Ver histórico de Alertas**

Actores:

Usuario

Objetivo:

Visualizar alertas propias del usuario ordenadas alfabéticamente.

Entrada:

Seleccionar el menú "Mis alertas".

Salida:

Lista de todas las alertas de una determinada categoría.

Secuencia normal:

- I. El usuario selecciona el menú "Mis alertas".
- II. Se muestra un listado, con el nombre y tipo de alerta, de todas las alertas que el usuario ha creado.

### **1.3. UC – Limpiar o eliminar las Alertas**

Actores:

Usuario

Objetivo:

Eliminar una o más alertas que ha creado el usuario.

Entrada:

Seleccionar el menú "Mis alertas".

Seleccionar una alerta específica.

Seleccionar la opción "eliminar".

Salida:

La lista de alertas personales del usuario queda reducida únicamente a las que éste prefiere.

Secuencia normal:

- I. El usuario selecciona el menú "Mis alertas".
- II. Se muestra un listado de todas las alertas del usuario.

- III. El usuario selecciona una alerta específica.
- IV. El usuario selecciona la opción "eliminar" para limpiar la alerta de su lista.
- V. Se muestra un mensaje de advertencia sobre la acción a realizar.
- VI. El usuario selecciona el botón "ok" y finaliza el proceso.

#### **1.4. UC – Compartir una alerta con alguien ajeno al sistema**

Actores:

Usuario

Objetivo:

Compartir una alerta fuera de la aplicación.

Entrada:

Seleccionar el menú "Mis alertas".

Seleccionar una alerta específica.

Seleccionar la opción "Compartir".

Salida:

Una alerta es compartida con una aplicación externo al sistema.

Secuencia normal:

- I. El usuario selecciona el menú "Mis alertas".
- II. Se muestra un listado de todas las alertas del usuario.
- III. El usuario selecciona la opción "Compartir".
- IV. Se muestra un desplegable con las posibles aplicaciones con las que puede compartir una alerta.
- V. El usuario selecciona una de las aplicaciones listadas.
- VI. Sale de la plataforma principal para interactuar con la aplicación seleccionada antes y realizar la acción de compartir la alerta

#### **1.5. UC – Eliminar cualquier alerta (web)**

Actores:

Administrador

Objetivo:

Eliminar una alerta de la base de datos.

Entrada:

Seleccionar del menú lateral "Gestión de contenidos".

Seleccionar la opción "Gestión de alertas".

Seleccionar una alerta de la tabla de alertas de la BBDD.

Salida:

La alerta queda eliminada de la base de datos.

Secuencia normal:

- I. El administrador selecciona del menú lateral "Gestión de contenidos".
- II. El administrador selecciona "Gestión de alertas"
- III. El administrador selecciona una alerta de la tabla de alertas.

- IV. El administrador selecciona el botón "Eliminar alerta".
- V. Se muestra un mensaje de advertencia sobre la acción realizada.
- VI. El administrador selecciona el botón "Aceptar" para finalizar el proceso.
- VII. Los datos de la alerta seleccionada son eliminados de la BBDD.

#### **1.6. Ver gráficos estadísticos del estado de la base de datos (web)**

Actores:

Administrador

Objetivo:

Visualizar gráficos estadísticos del contenido y el estado actual de la base de datos.

Entrada:

Seleccionar el menú lateral izquierdo una de las opciones de reporte de visualización de alertas o usuarios.

Rellenar o seleccionar una las opciones que solicita la pantalla.

Salida:

Se visualiza un gráfico estadístico sobre los datos actuales de la base de datos, según el tipo de filtro y entradas recibidas.

Secuencia normal:

- I. El administrador selecciona del menú lateral una de las opciones de visualización de gráficos.
- II. Se muestra una pantalla descriptiva con opciones para introducir unos datos solicitados o escoger uno de los parámetros de entrada.
- III. Se muestra el reporte de un gráfico estadístico según el tipo de parámetros que se ha introducido.

## 4 TECNOLOGÍAS EMPLEADAS

Las herramientas tecnológicas que hoy en día se emplean para el desarrollo de aplicaciones son muy variadas y de distinta índole. En este caso se va a describir las que se han utilizado para el desarrollo de este proyecto.

### 4.1 Javascript (JS)

Lenguaje de programación interpretado, basado en prototipos, es imperativo y estructurado, utilizado principalmente para el desarrollo orientado a objetos y eventos, es débilmente tipado y muy dinámico. La forma más habitual de uso es para el lado front-end (lado cliente) pero también se puede utilizar para el lado del servidor.

### 4.2 HTML5 y CSS3

La última versión del lenguaje de marcado para desarrollar páginas web es HTML5, gracias al cual se puede establecer estructuralmente el contenido de una página web.

El lenguaje de estilos CSS se utiliza para describir cómo se representa el contenido de una página web, mediante reglas simples y compuestas que modifican la forma que los elementos de un documento son presentados.

### 4.3 Angular

Angular es un framework de javascript muy completa, creado y mantenido por Google, y es usado generalmente para el desarrollo front-end de aplicaciones web y móvil, y también como parte de una aplicación full-stack. Es de código abierto y está organizado en estructuras de componentes, módulos y servicios. Pensado para crear soluciones All-in-one gracias a sus propios módulos y otros externos disponibles (Routing, HTTP, RxJS, etc).

Utiliza un tipo de patrón MCV, más concretamente el patrón Modelo-Vista Vista-Modelo (MVVM) y toda la funcionalidad TypeScript.

### 4.4 Ionic

Ionic Framework es una herramienta de código abierto integrada totalmente con Angular, soporta Vue y React, proporciona un conjunto de módulos orientado al desarrollo de aplicaciones web, móvil y de escritorio. Está baso en estándares web (THML, CSS y Javascript), tiene componentes web independientes (ES5, ES6) y reusables para un sistema web progresivo (PWA).

#### 4.5 NPM

Node Packaged Manager (NPM) es una herramienta de gestión de paquetes y otras herramientas software para Javascript. Mediante una interfaz de línea de comandos denominado "npm" actúa como mediador entre el Node Package Registry (una extensa base de datos online de Javascript, con paquetes públicos y/o privados, incluido toda la metainformación) y el entorno de desarrollo de una aplicación.

#### 4.6 Firebase

Firebase es una plataforma ubicada en la nube para el desarrollo de aplicaciones en diferentes plataformas (Android, IOS, web), está integrada con Google Cloud Platform con la finalidad de guardar y sincronizar datos en la nube en tiempo real.

Provee de varios servicios: Análítica (Firebase Analytics), desarrollo, mensajería en la nube y autenticación (Firebase cloud Messaging and Auth), base de datos en tiempo real (Realtime Database), Almacenamiento y base de datos NoSQL (Firebase Storage and Firestore), etc. [4]

Además, entre otras características destacan; la creación de proyectos sin la necesidad de un servidor, y posee una infraestructura escalable para cualquier tipo de aplicaciones ya que dispone de herramientas multiplataforma.

#### 4.7 Visual Studio Code

Es una herramienta de edición de código fuente redefinido y optimizado para la creación, compilación y depuración de aplicaciones web. Está desarrollada por Microsoft para plataformas como Windows, Linux y macOS. Incluye control integrado de Git [5], comprensión IntelliSense, lint y refactorización de código, dispone de cobertura de lenguajes de programación con resaltado de sintaxis y coincidencias de signos de puntuación de codificación.

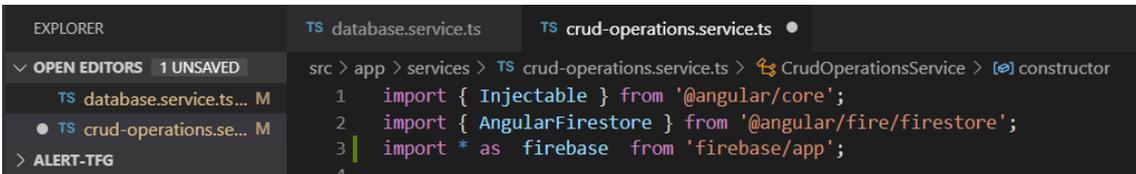
## 5 MODELO DE DATOS

Uno de los puntos más importantes del diseño de un sistema es la elección y representación de la persistencia de los datos e información. En este capítulo se va a representar y explicar el modelo de datos del sistema.

### 5.1 Base de Datos no relacional

Como se esperaba que la cantidad de datos sea masiva se ha optado por utilizar una base de datos no relacional (NoSQL), así que ha elegido usar los servicios de la nube de Firebase-Firestore [6], ya que ofrece funciones sincronizadas y en tiempo real, que a diferencia de MongoDB no soporta la agregación de datos sino que la base de datos es una estructura enorme de nodos y objetos JSON. Estos datos escalables, con una relación estática y unidireccional, se anidan de forma que los objetos secundarios están bajo los primarios.

Todas las operaciones CRUD [7] se realizan utilizando la librería disponible de AngularFireStore(@angular/fire/firestore) en Ionic, como se muestra en las siguientes imágenes:



```
EXPLORER
  OPEN EDITORS 1 UNSAVED
    TS database.service.ts... M
    TS crud-operations.se... M
  ALERT-TFG

TS database.service.ts
TS crud-operations.service.ts

src > app > services > TS crud-operations.service.ts > CrudOperationsService > constructor
1 import { Injectable } from '@angular/core';
2 import { AngularFireStore } from '@angular/fire/firestore';
3 import * as firebase from 'firebase/app';
4
```

Figura 3: Importación de algunas librerías de código abierto para un proyecto Ionic-angular – Angular repo, angularfire2 y Firebase

```
src > app > services > TS crud-operations.service.ts > CrudOperationsService
15
16 /* Users */
17
18 createUser(record) {
19   return this.firestore.collection('users').add(record);
20 }
21
22 readUsers() {
23   return this.firestore.collection('users').snapshotChanges();
24 }
25
26 readUser(record: string) {
27   return this.firestore.collection('users').doc(record).snapshotChanges();
28 }
29
30 updateUser(recordID, record) {
31   return this.firestore.doc('users/' + recordID).update(record);
32 }
33
34 deleteUser(recordID) {
35   this.firestore.doc('users/' + recordID).delete();
36 }
37
38
39 /* Alerts */
40 readAlerts(user) {
41   return this.firestore.collection('users').doc(user).collection('myAlerts').snapshotChanges();
42 }
43
44 updateAlert(user, recordID, record){
45   this.firestore.collection('users').doc(user).collection('myAlerts').doc(recordID).update(record);
46   this.firestore.collection('alerts').doc(recordID).update(record);
47 }
48
49 deleteAlert(user, recordID) {
50   this.firestore.collection('users').doc(user).collection('myAlerts').doc(recordID).delete();
51   this.firestore.collection('alerts').doc(recordID).delete();
52 }
53
54
```

Figura 4: Métodos principales de operaciones CRUD con Firebase-Firestore

Así la información de la base de datos se almacena en documentos identificados por el id "usuarios" y una sucesión de colecciones y otros documentos, alternando uno con otro.

En la siguiente figura se pueden ver las colecciones y documentos de la base de datos.

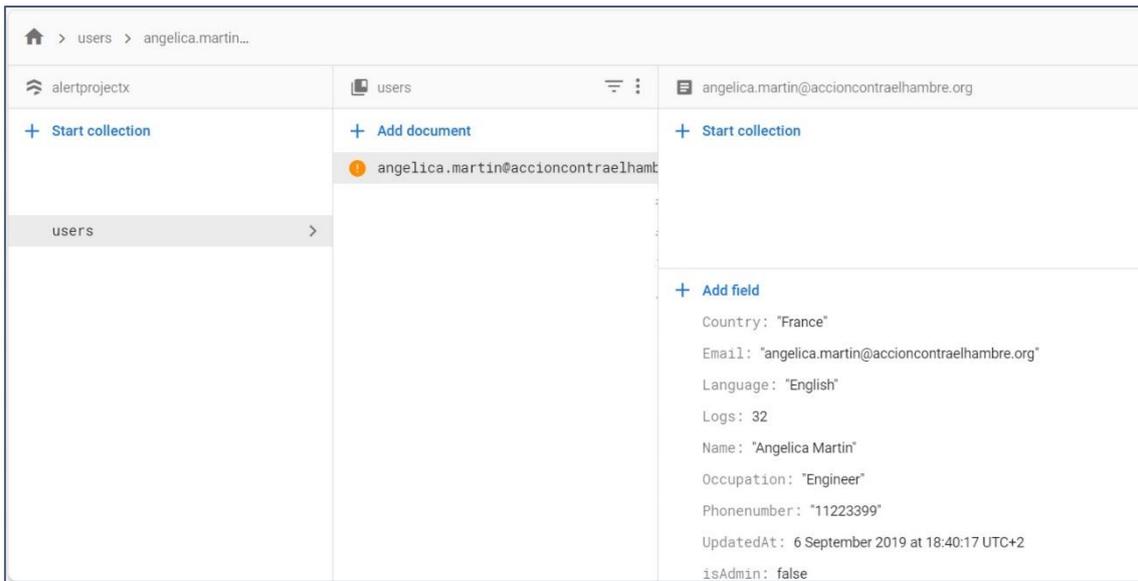


Figura 5: Colección de Usuarios y un Documento de un usuario concreto

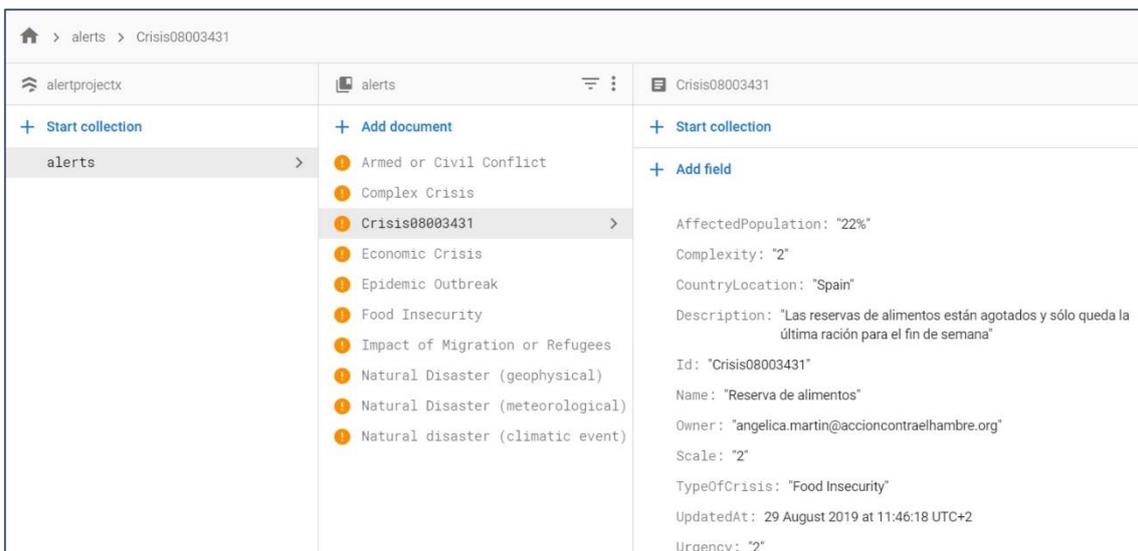


Figura 6: Colección de Alertas y una serie de Documentos y sus datos

Las principales colecciones son los "Usuarios", "Alertas" y "Localizaciones".

## Usuario

La colección que contiene la información de un usuario tiene los siguientes campos:

- Email: (cadena) Contiene el correo electrónico del usuario y es la vez el id de su documento.
- Nombre: (cadena) Contiene el nombre completo del usuario.
- País [HeadQuarter]: (cadena) Contiene el país donde trabaja el usuario.
- Ciudad: (cadena) Contiene la ciudad donde reside.

- Género: (cadena) Contiene el género del usuario.
- Profesión: (cadena) Contiene la profesión a la que se dedica el usuario.
- Lenguaje: (cadena) Contiene el lenguaje preferido del usuario.
- Teléfono: (entero) Contiene el número de contacto del usuario.
- CreadoEn: (timestamp) Contiene la fecha en la que el usuario se ha dado de alta.
- EsAdmin: (boolean) Contiene un valor boolean "false" o "true" dependiendo si es administrador o no.

## Alerta

- Id: (cadena) Es la clave de su documento.
- Nombre: (cadena) Contiene el nombre de la alerta.
- Localización: (cadena) El lugar del que se reporta por el usuario que creó la alerta.
- País [Head Quarter]: (cadena) Localización a la que pertenece el usuario que creó la alerta.
- Población afectada (cadena) Es el porcentaje de la población afectada.
- Complejidad: (entero) Nivel de complejidad de la alerta.
- Escala: (entero) Nivel de escala o balanza.
- Urgencia: (entero) Nivel de urgencia.
- CreadoEn: (TimeStamp) Fecha en la que se creó la alerta.
- ActualizadoEn: (TimeStamp) Fecha de última actualización.
- Propietario: (cadena) Es el Id del usuario quien creó la alerta.
- Descripción: (cadena) Contiene la descripción de la alerta.

## Imágenes

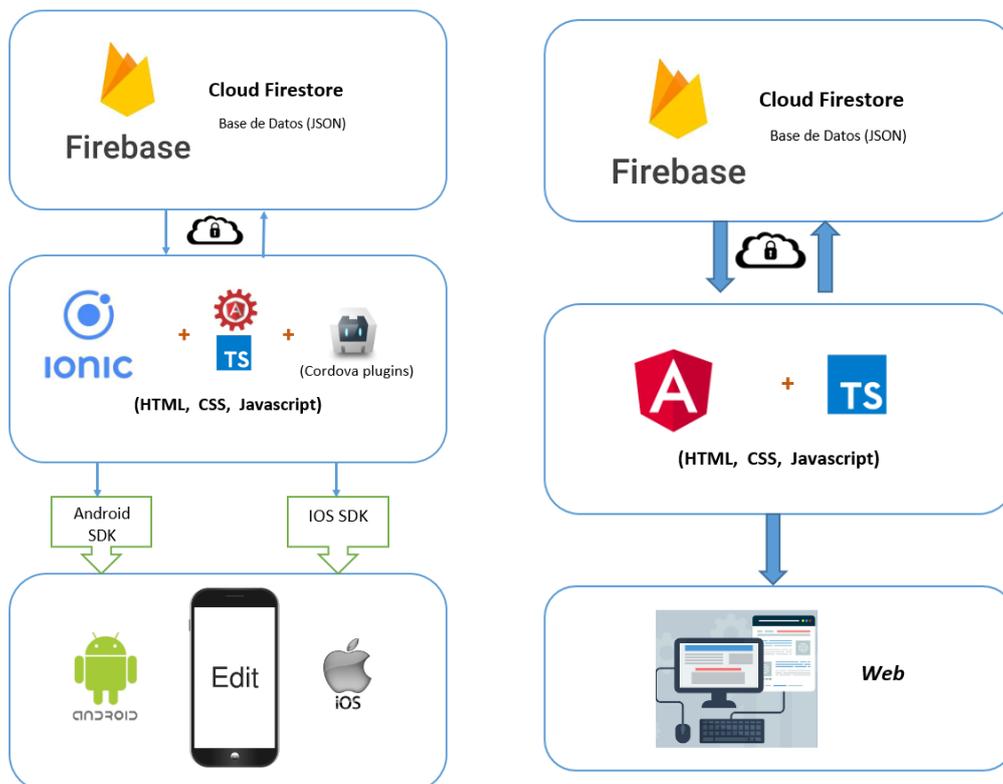
- Dirección: (cadena) La ruta donde está almacenada la imagen.
- Nombre: (string) Nombre del fichero de imagen.
- Tamaño: (entero) Contiene el tamaño de la imagen

## 6 ARQUITECTURA DEL SISTEMA

En este capítulo se describe la arquitectura del sistema que está compuesto por varios módulos en los cuales se utiliza tecnologías de un amplio abanico.

Así de forma global, esta arquitectura se conforma de una parte como aplicación móvil (para la gestión de contenidos), otra de un cliente web (para la consulta de datos y una parte del ámbito administrativo), y una base de datos no relacional para el almacenamiento de todos los datos.

El diseño de la arquitectura del proyecto con las tecnologías empleadas, se muestran en la siguiente *figura 7*. A la izquierda el sistema móvil y a la derecha el sistema web.



*Figura 7: Arquitectura que sigue el sistema. Aplicación móvil y web*

Ahora bien, los módulos que componen el modelo, servicios y componentes del proyecto están estructurados de la siguiente forma. Ver *Figura 8*.

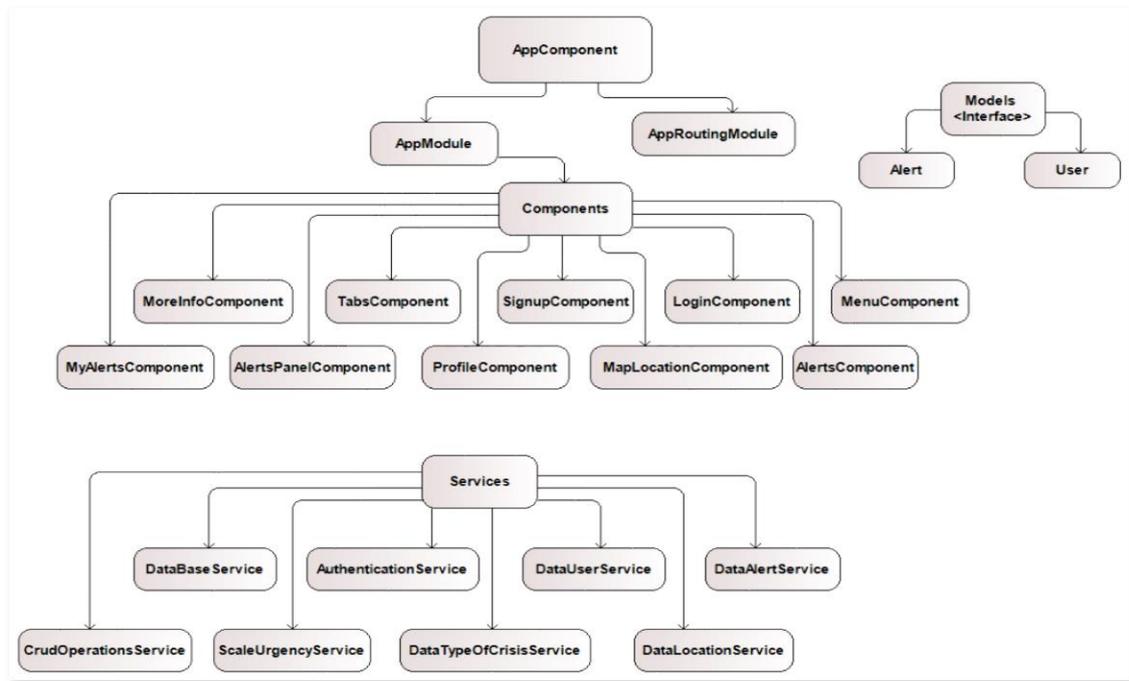


Figura 8: Diagrama de módulos de la aplicación móvil

## 6.1 Aplicación móvil

El framework que se utiliza para el desarrollo de la herramienta móvil es Ionic 4, éste usa un patrón del tipo Vista-Controlador ya que está basado en Angular. Las consultas que se realizan directamente a la base de datos (operaciones CRUD) están situadas en la parte de servicios de la aplicación. Las páginas utilizadas en el proyecto Ionic-Angular están divididas en módulos, así cada página es un módulo, y estos utilizan una técnica llamada Lazy-Loading para optimizar el rendimiento a la hora de cargar los datos de la aplicación.

## 6.2 Aplicación web

Para el desarrollo de la herramienta web se ha empleado Angular 6, éste utiliza un patrón de diseño modelo-vista vista-modelo (MVVM) puesto que la forma de sincronizar los datos entre la vista y el modelo es muy dependiente, y todo gracias a característica base de angular que es "two-way data binding"

## 6.3 Base de datos

Como base de datos del sistema se establece uno del tipo no relacional que se encuentra en el lado del servidor (Firebase), y se utilizará para el almacenamiento de todos los datos (datos del usuario, alertas, localizaciones, Timestamp, etc.)

## 7 DISEÑO E IMPLEMENTACIÓN

Toda la parte de las principales funcionalidades del sistema se reflejará en este capítulo, desde la parte del desarrollo de aplicación móvil hasta la parte de desarrollo web. Además de las reglas de diseño empleadas para la representación de los datos.

### 7.1 Diseño

Así pues, para la capa de presentación se utilizó las reglas formales de personalización CSS según el manual de estilos de diseño desarrollada por Google; "Material Design" [8], ya que el desarrollo en HTML y javascript en Angular [9] lo requieren para una atractiva representación de ésta capa.

Puesto que la aplicación está pensada para personas con poca experiencia en el campo de la informática se ha tratado de utilizar algunos principios de experiencia de usuario, como son la **Navegación** (la forma en la que se navega a través de los contenidos de la herramienta por medio de las diferentes pantallas de la interfaz), la **Simplicidad** (utilizar el mínimo y de elementos necesarios en la interfaz para que se pueda interactuar en ella sin otros elementos que puedan perjudicar la función primaria de la aplicación), el **producto útil y visible** (proporcionar una ayuda contextual con el uso de colores de resaltado con el que se pueda estar familiarizado).

### 7.2 Implementación

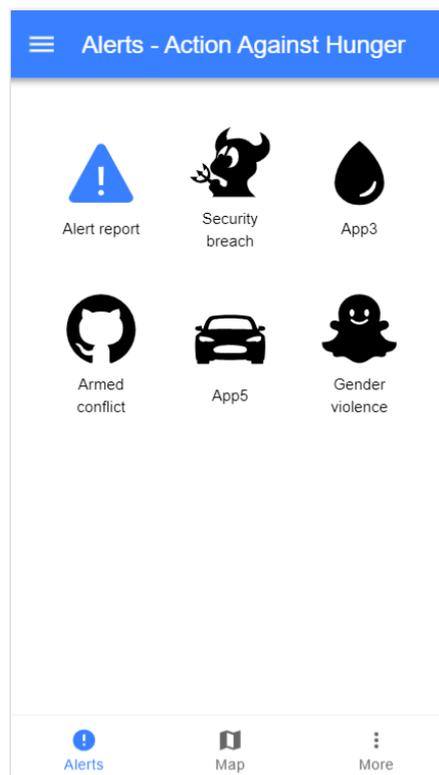
Funcionalidades de la aplicación móvil:

Los iconos que se utilizan para este apartado has sido proporcionados del propio framework Ionic 4 [10]. Y la parte implementada, tanto HTML [11] y Javascript se reflejan a continuación.

### **Visualizar menú de alertas**

El Menú de alertas es la primera página que aparece una vez el usuario se haya identificado en la plataforma. Las siguientes imágenes muestran la interfaz de navegación del usuario y la implementación de la misma.

Como se aprecia en la siguiente *figura 9*, el desarrollo de la actual herramienta es de hecho el primer icono que aparece como "Alert report" junto a otras aplicaciones de tipo demo.



*Figura 9: Detalle de la vista del Menú principal*

Y una parte de la implementación de código es la siguiente:

```

<ion-content padding>
  <ion-grid>
    <ion-row align-items-center>
      <ion-col text-center align-self-center padding="4" size="4" class="alert-image" routerLink="/menu/form2">
        <!--<ion-button shape="round">-->
          <ion-icon color="primary" offset="blabla" class="iconito" src="/assets/warning.svg"></ion-icon> <!--slot="start"-->
          <ion-label class="label-icon" position="stacked">Alert report</ion-label>
        <!--</ion-button-->
      </ion-col>

      <ion-col text-center align-self-center padding="4" size="4" class="alert-image" routerLink="/menu/edit-profile">
        <ion-icon class="iconito" src="/assets/frebsd-devil.svg"></ion-icon>
        <ion-label class="label-icon" position="stacked">Security breach</ion-label>
        <!--<ion-button shape="round" color="primary" routerLink="/form-alert">Create an alert</ion-button-->
      </ion-col>

      <ion-col text-center align-self-center padding="4" size="4" class="alert-image" routerLink="/menu/edit-profile">
        <ion-icon class="iconito" src="/assets/water.svg"></ion-icon>
        <ion-label class="label-icon" position="stacked">App3</ion-label>
        <!--<ion-button shape="round" color="primary" routerLink="/form-alert">Create an alert</ion-button-->
      </ion-col>
    </ion-row>

    <ion-row align-items-center>
      <ion-col text-center align-self-center padding="4" size="4" class="alert-image" routerLink="/menu/edit-profile">
        <ion-icon class="iconito" src="/assets/github.svg"></ion-icon>
        <ion-label class="label-icon" position="stacked">Armed conflict</ion-label>
        <!--<ion-button shape="round" color="primary" routerLink="/form-alert">Create an alert</ion-button-->
      </ion-col>

      <ion-col text-center align-self-center padding="4" size="4" class="alert-image" routerLink="/menu/edit-profile">
        <ion-icon class="iconito" src="/assets/model-s.svg"></ion-icon>
        <ion-label class="label-icon" position="stacked">App5</ion-label>
        <!--<ion-button shape="round" color="primary" routerLink="/form-alert">Create an alert</ion-button-->
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>

```

Figura 10: Fragmento de código implementado del Menú principal

### Formulario de la alerta

Parte de los datos principales que se registran en la base de datos están en esta funcionalidad. Al crear una nueva alerta (dar de alta alerta) se introducen ciertos campos como el tipo de alerta, nivel de complejidad, escala, descripción, localización, etc.

En las *figuras 11* y *12* se muestran las vistas de unos cuantos “slides” de lo que conforma el formulario de alertas (La ciudad donde reside el usuario, nombre de la alerta, nivel de complejidad y la localización donde se produce el incidente).

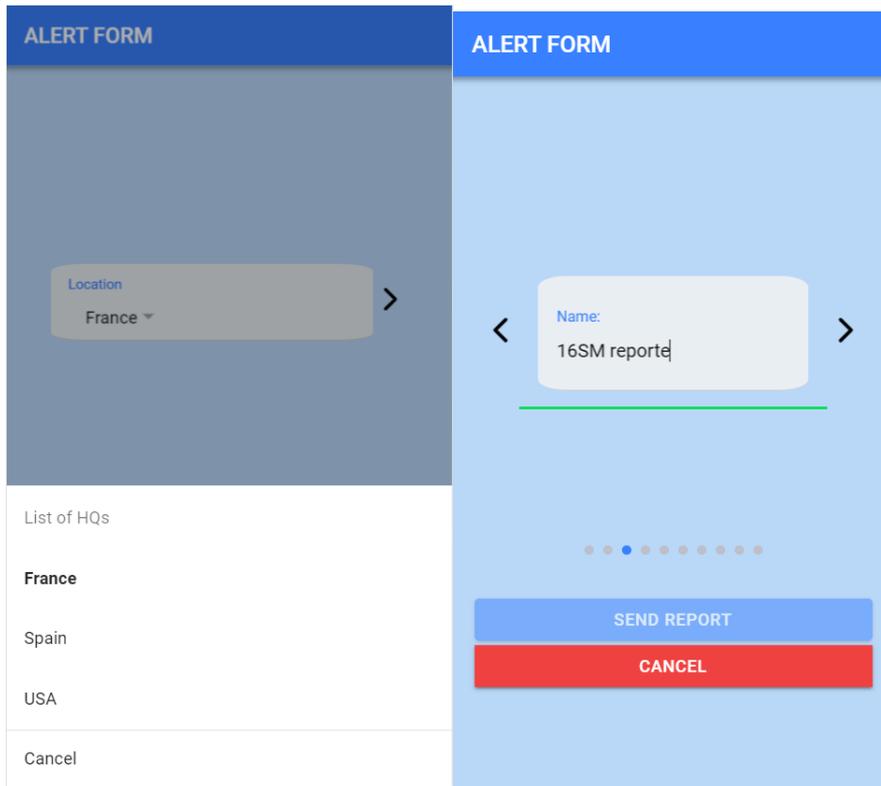


Figura 11: Detalles de algunas vistas del formulario de alerta – parte 1

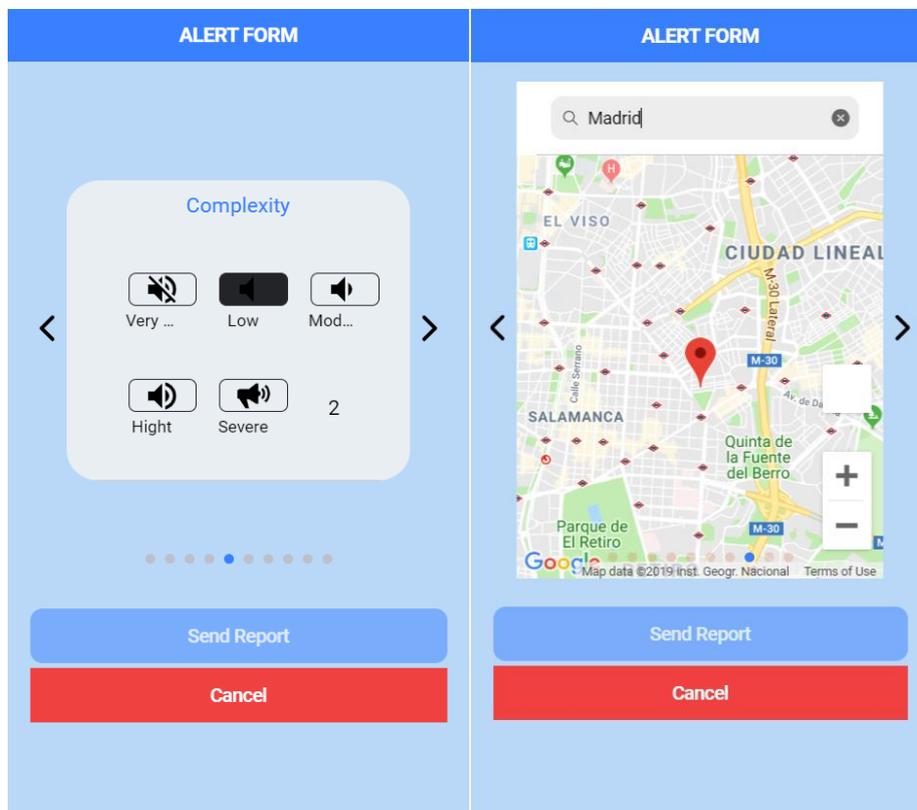


Figura 12: Detalles de algunas vistas del formulario de alerta – parte 2

Un fragmento de código de la funcionalidad de formulario de alertas se muestra en la *figura 13*.

```
ngOnInit() {  
  
    this._alertID = this.bdService.generateID();  
    console.log('Alert beta ID: ', this._alertID, "\n");  
    this._userID = this.authService.getUserEmail();  
  
    this._crisis = this.crisisService.getDataTypeOfCrisis();  
    this._locations = this.locationService.getDataLocation();  
  
    this._scale = this.levelOfRisk.getLevelRiskScale();  
    this._complexity = this.levelOfRisk.getLevelRiskComplexity();  
    this._urgency = this.levelOfRisk.getLevelRiskUrgency();  
  
    console.log("El usuario registrado es: ", this._userID);  
  
    this.rForm2 = this.fb.group({  
        'location': [null, Validators.required],  
        'typeOfCrisis': [null, Validators.required],  
        'name': [null, Validators.required],  
        'scale': [null, Validators.required],  
        'complexity': [null, Validators.required],  
        'urgency': [null, Validators.compose([  
            Validators.required,  
            Validators.min(1),  
            Validators.max(5)])],  
  
        'affectedPopulation': [null, Validators.required],  
        'affectedPersons': [null, Validators.required],  
        //'searchControl': [null, Validators.required],  
        //'file': [null, Validators.toString()],  
        'description': [null, Validators.required],  
    });  
    //Load map  
    //this.ionViewDidLoad();  
}
```

*Figura 13: Fragmento de código de la implementación del formulario de alertas*

## Iniciar sesión

Una de las pantallas que visualiza el usuario como primera interfaz es la funcionalidad del login. Ya que para poder interactuar con la herramienta el usuario debe haber iniciado sesión previamente. De este modo, los datos requeridos para iniciar sesión son el correo electrónico y la contraseña del usuario.

En la siguiente *figura 14*, se presenta la interfaz que visualiza un usuario nada más abrir la aplicación para poder interactuar en ella. En la derecha se ven algunos detalles de la implementación de esta funcionalidad y la *figura 15* muestra un fragmento de código de la parte HTML.

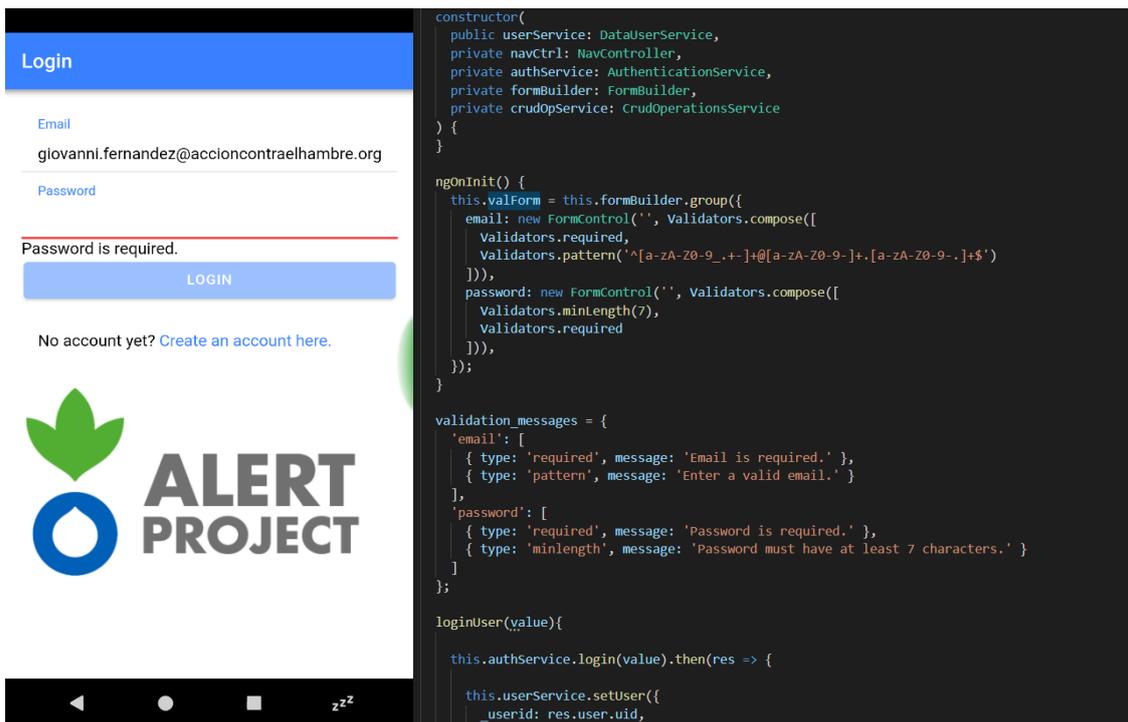


Figura 14: Detalle de la vista de Login y parte de la implementación de su modelo

```

<ion-header>
  <ion-toolbar color="primary">
    <ion-title>Login</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content padding>

  <form class="login_form" [formGroup]="valForm" (ngSubmit)="loginUser(valForm.value)">

    <ion-item>
      <ion-label position="floating" color="primary">Email</ion-label>
      <ion-input type="text" formControlName="email"></ion-input>
    </ion-item>
    <div class="validation-errors">
      <ng-container *ngFor="let validation of validation_messages.email">
        <div class="error-message" *ngIf="valForm.get('email').hasError(validation.type) && (valForm.get('email').dirty || valForm.get('email').t
          {{ validation.message }}
        </div>
      </ng-container>
    </div>
    <ion-item>
      <ion-label position="floating" color="primary">Password</ion-label>
      <ion-input type="password" formControlName="password" class="form-control" required></ion-input>
    </ion-item>
    <div class="validation-errors">
      <ng-container *ngFor="let validation of validation_messages.password">
        <div class="error-message" *ngIf="valForm.get('password').hasError(validation.type) && (valForm.get('password').dirty || valForm.get('pas
          {{ validation.message }}
        </div>
      </ng-container>
    </div>
  </form>

```

Figura 15: Fragmento de código de la implementación HTML de Login

### Consultar y Editar perfil

En ésta funcionalidad se puede consultar los datos personales del usuario y editarlos según sus necesidades. Ver figura 16 de la vista de la interfaz de perfil de usuario.

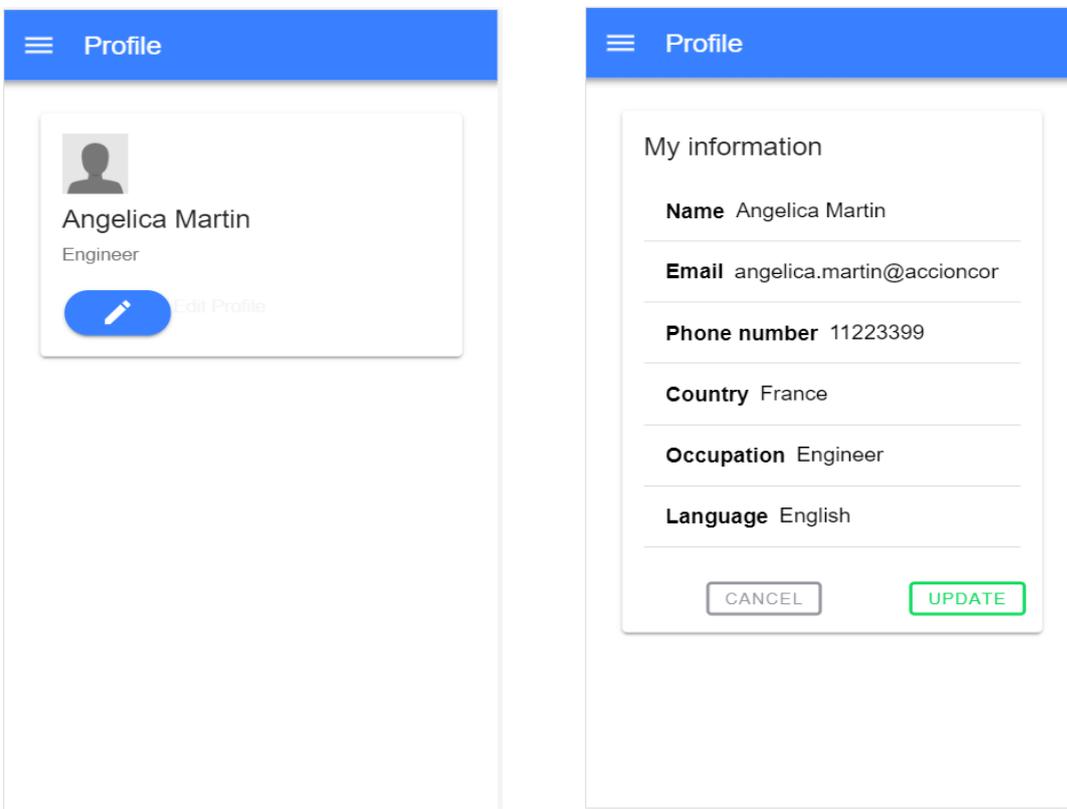


Figura 16: Vistas de la funcionalidad – Modificar perfil de usuario

Los datos del usuario se cargan en el momento en el que los componentes de la página de perfil empiezan a cargarse (ver *figura 17*). En la *figura 17* se muestra el método para editar el perfil de usuario con los campos que el usuario quiera modificar para luego pasar éstos a otro método de actualización de perfil.

Y en la *figura 18* se refleja cómo está estructurado las etiquetas de los componentes de Ionic para su implementación HTML.

```
ngOnInit() {  
  
  this.userInfo = this.userService.getUserInfo();  
  
  console.log('My email: ', this.authService.getUserEmail());  
  this.crudOpService.readUser(this.authService.getUserEmail()).subscribe(res => {  
  
    this.userInfo = {  
      _username: res.payload.data()['Name'],  
      _email: res.payload.data()['Email'],  
      _phonenumber: res.payload.data()['Phonenumber'],  
      _country: res.payload.data()['Country'],  
      _occupation: res.payload.data()['Occupation'],  
      _language: res.payload.data()['Language'],  
      _isEdit: false  
    };  
    console.log("Profile user: ", this.userInfo);  
  });  
}  
  
editProfile(data) {  
  if (this.authService.userLogged) {  
    console.log("Estoy LOGEADO");  
    data.isEdit = true;  
    data.EditName = data._username;  
    data.EditEmail = data._email;  
    data.EditNumber = data._phonenumber;  
    data.EditCountry = data._country;  
    data.EditOccupation = data._occupation;  
    data.EditLanguage = data._language;  
  }  
}  
  
async presentAlert(title: string, content: string) {  
  const alert = await this.alertCtrl.create({  
    header: title,  
    message: content,  
    buttons: ['OK']  
  });  
}
```

*Figura 17: Fragmento de código del modelo – Modificar perfil*

```

<ion-card-title>{{userInfo.username}}</ion-card-title>
<ion-card-subtitle>{{userInfo.occupation}}</ion-card-subtitle>
</ion-card-header>
<ion-card-content>
<ion-button shape="round" color="primary" size="medium" (click)="editProfile(userInfo)">
  <ion-icon size="medium" slot="icon-only" name="create"></ion-icon>
</ion-button>
<ion-label color="light">Edit Profile</ion-label>
</ion-card-content>
</span>
<ng-template #elseBlock>
<ion-card-header>
<ion-card-title>
  My information
</ion-card-title>
</ion-card-header>
<ion-card-content>
<ion-item>
  <ion-label><strong>Name</strong></ion-label>
  <ion-input type="text" [(ngModel)]="userInfo.EditName"></ion-input>
</ion-item>
<ion-item>
  <ion-label><strong>Email</strong></ion-label>
  <ion-input type="text" [(ngModel)]="userInfo.EditEmail"></ion-input>
</ion-item>
<ion-item>
  <ion-label><strong>Phone number</strong></ion-label>
  <ion-input type="text" [(ngModel)]="userInfo.EditNumber"></ion-input>
</ion-item>
<ion-item>
  <ion-label><strong>Country</strong></ion-label>
  <ion-input type="text" [(ngModel)]="userInfo.EditCountry"></ion-input>
</ion-item>
<ion-item>
  <ion-label><strong>Occupation</strong></ion-label>
  <ion-input type="text" [(ngModel)]="userInfo.EditOccupation"></ion-input>
</ion-item>
<ion-item>
  <ion-label><strong>Language</strong></ion-label>

```

*Figura 18: Fragmento de código HTML – Modificar perfil*

### **Ver mis alertas**

Al igual que el caso de consultar y editar perfil, en esta funcionalidad se puede ver una lista de todas las alertas creadas por el usuario con opciones de editar o eliminar una alerta seleccionada.

En la *figura 19* se aprecia la interfaz de la lista de alertas que creó el usuario (con la posibilidad de editarlas o eliminarlas). En la figura de la derecha se tiene la vista de la edición de una alerta del usuario (con la posibilidad de modificar los datos de dicha alerta).

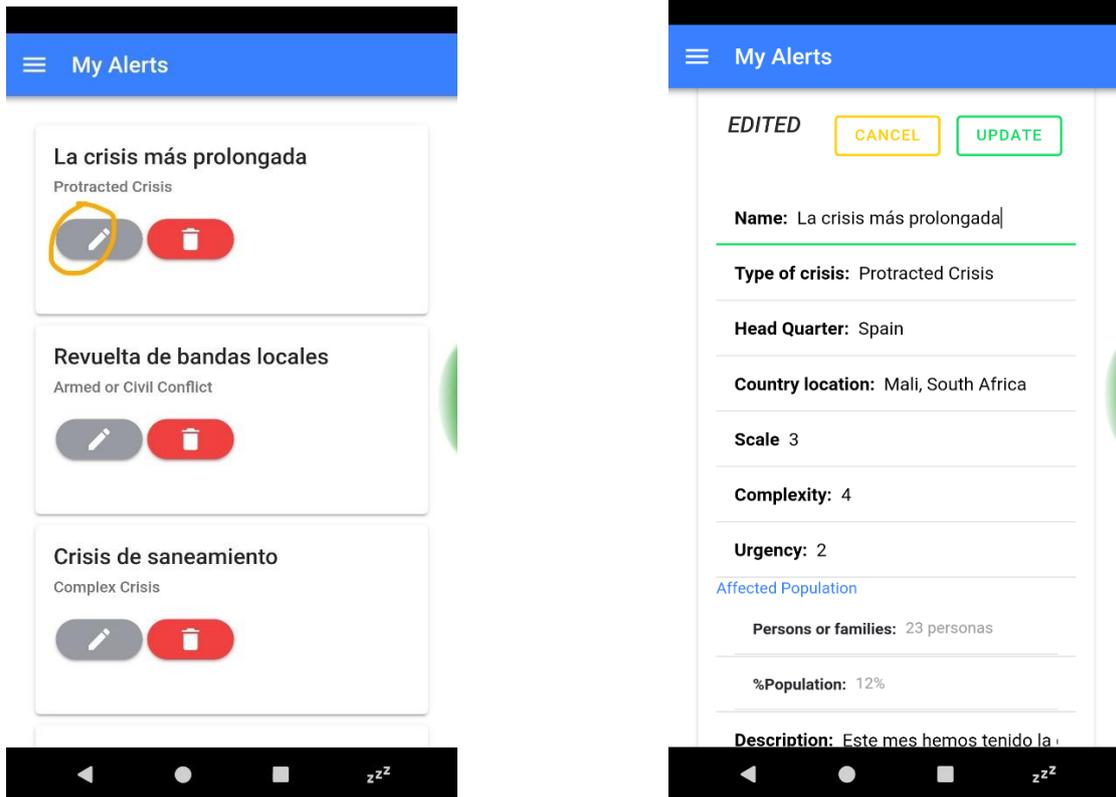


Figura 19: Vistas de la lista de alertas del usuario y la edición de una alerta

Los fragmentos de código de la implementación de ésta funcionalidad se muestran a continuación. Ver *figuras 20 y 21*. En la figura 21 se aprecia el modo en el que se cargan las alertas de un usuario utilizando la API de firebase [12], y también se pueden ver los métodos de “editar alerta” y “actualizar alerta” (editAlert, updateAlert).

```

<ion-content padding>
<ion-card *ngFor="let alert of alerts">
  <span *ngIf="!alert.isEdit; else elseBlock">
    <ion-card-header>
      <ion-card-title>{{alert.Name}}</ion-card-title>
      <ion-card-subtitle>{{alert.TypeOfCrisis}}</ion-card-subtitle>
    </ion-card-header>
    <ion-card-content>

      <ion-button shape="round" color="medium" size="small" (click)="editAlert(alert)">
        <ion-icon size="medium" slot="icon-only" name="create"></ion-icon>

      </ion-button>
      <ion-button shape="round" color="danger" size="small" (click)="removeAlert(alert.id)">
        <ion-icon size="medium" slot="icon-only" name="trash"></ion-icon>

      </ion-button>
    </ion-card-content>
  </span>

  <ng-template #elseBlock>
    <ion-card-header>
      <ion-card-title>
        <ion-grid>
          <ion-row>
            <ion-col>
              Edit
            </ion-col>
            <ion-col>
              <ion-button fill="outline" color="warning" size="small" (click)="alert.isEdit = false">
                Cancel
              </ion-button>
            </ion-col>
            <ion-col>
              <ion-button fill="outline" color="success" size="small" (click)="updateAlert(alert)">
                Update
              </ion-button>
            </ion-col>
          </ion-row>
        </ion-grid>
      </ion-card-title>
    </ion-card-header>
  </ng-template>
</ion-card>
</ion-content>

```

Figura 20: Fragmento de código HTML de las alertas del usuario

```

constructor(private authService: AuthenticationService, private crudService: CrudOperationsService) { }

ngOnInit() {
  this.crudService.readAlerts(this.authService.getUserEmail()).subscribe(data => {
    this.alerts = data.map(res => {
      return {
        id: res.payload.doc.id,
        Name: res.payload.doc.data()['Name'],
        TypeOfCrisis: res.payload.doc.data()['TypeOfCrisis'],
        CountryLocation: res.payload.doc.data()['CountryLocation'],
        Scale: res.payload.doc.data()['Scale'],
        Complexity: res.payload.doc.data()['Complexity'],
        Urgency: res.payload.doc.data()['Urgency'],
        AffectedPopulation: res.payload.doc.data()['AffectedPopulation'],
        Description: res.payload.doc.data()['Description'],
        isEdit: false
      };
    });
  });
  console.log("init MyAlerts: ", this.alerts);
}

editAlert(data) {
  data.isEdit = true;
  data.EditName = data.Name;
  data.EditTypeOfCrisis = data.TypeOfCrisis;
  data.EditCountryLocation = data.CountryLocation;
  data.EditScale = data.Scale;
  data.EditComplexity = data.Complexity;
  data.EditUrgency = data.Urgency;
  data.EditAffectedPopulation = data.AffectedPopulation;
  data.EditDescription = data.Description;
}

updateAlert(data) {
  let tempData = {};
  tempData['Name'] = data.EditName;
  tempData['TypeOfCrisis'] = data.EditTypeOfCrisis;
  tempData['CountryLocation'] = data.EditCountryLocation;
}

```

Figura 21: Fragmento de código del modelo de alertas del usuario, editar y actualizar la alerta

Funcionalidades de la aplicación web:

### Visualizar datos estadísticos de la base de datos

El administrador podrá acceder a la aplicación web para visualizar una serie de gráficos estadísticos de lo que está ocurriendo en la base de datos; por ejemplo, qué tipos de alertas se registran más frecuentemente en la base de datos, o cuáles son los usuarios que más veces utilizan la aplicación, o en qué localizaciones se registran más alertas, etc. En figura 22 se precia la gráfica de las N alertas más recientes agrupadas por el tipo de alerta al que pertenecen.

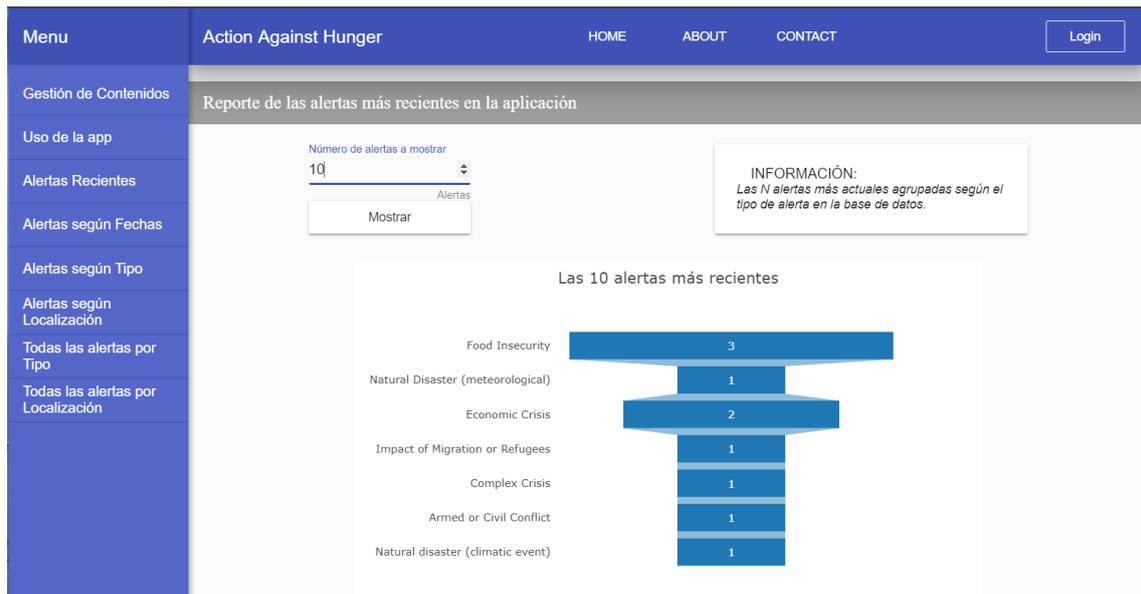


Figura 22: Gráficos de las Alertas más recientes, filtradas por la cantidad introducida

Como se puede ver en la figura 23 se visualiza los nombres de los usuarios que más veces han utilizado la herramienta.

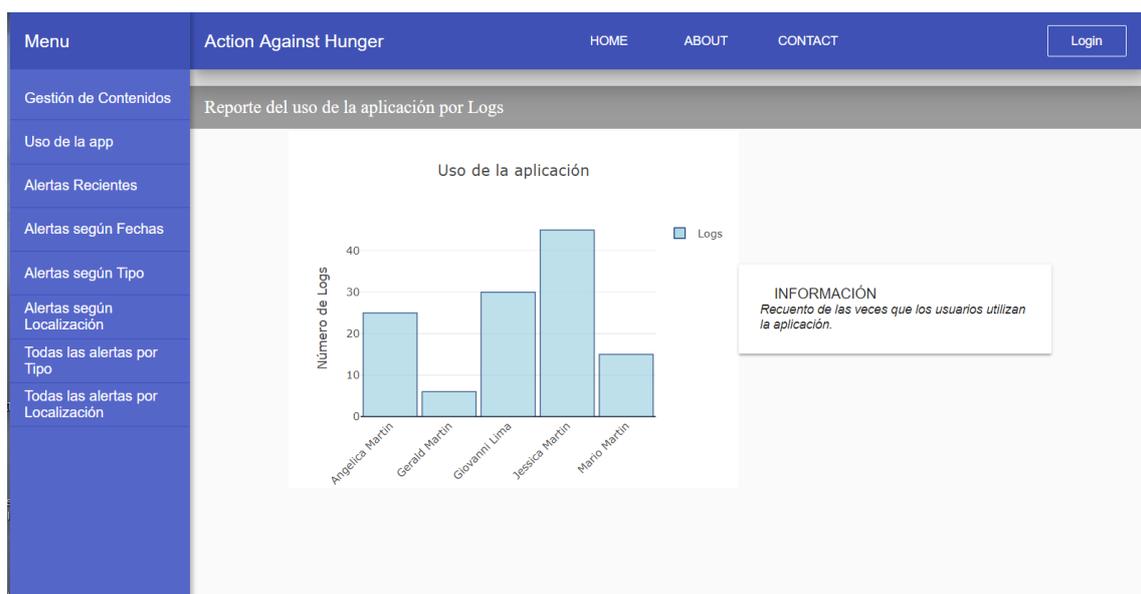


Figura 23: Gráficos de la frecuencia con la que un usuario utiliza la app

Otro gráfico representativo es la figura 24, que representa las alertas que se han registrado en la base de datos, filtradas y acotadas entre dos fechas que elige el usuario.

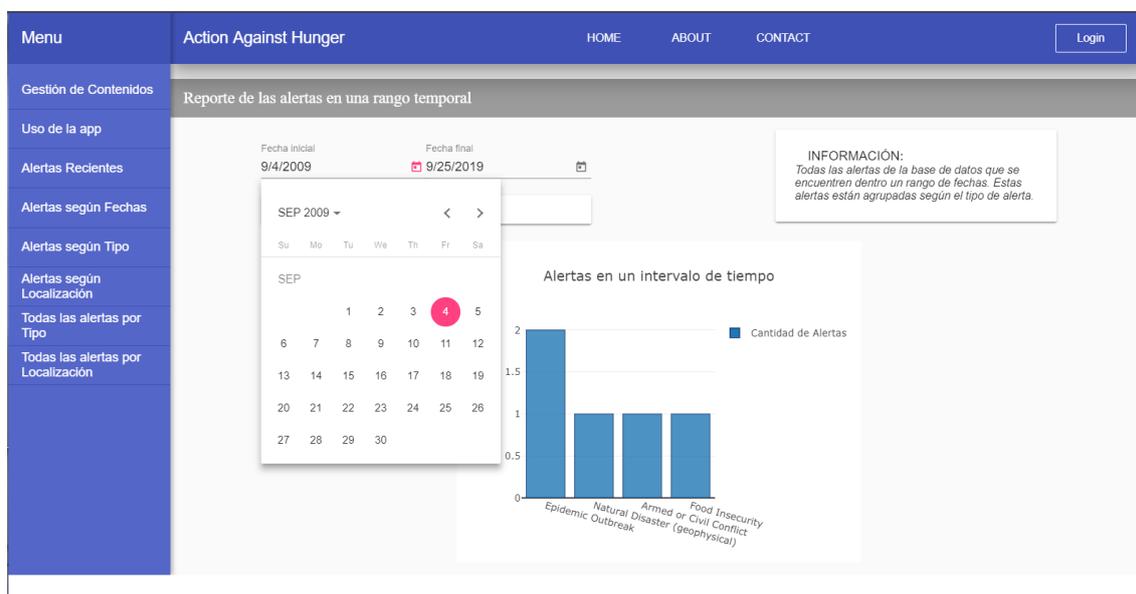


Figura 24: Gráficos de las Alertas que se han registrado entre las fechas elegidas

### Gestión de usuarios y alertas

Como es propio de un Usuario con permisos de administrador, este puede consultar la lista de usuarios que están dados de alta en la aplicación, modificar sus datos o eliminar a los usuarios (dar de baja a usuarios). De igual forma también puede acceder y consultar todas las alertas dadas de alta, modificar sus datos o eliminar sus datos. En las figuras 25 y 26 se pueden ver el contenido de unas tablas avanzadas con unos datos representativos de los usuarios (figura 25) y las alertas (figura 26) de la base de datos.

Menu	Action Against Hunger	PERFIL	ACERCA DE	CONTACTO	INICIAR SESIÓN		
Gestión de Contenidos	<b>GESTIÓN DE USUARIOS</b>						
- Uso de la app	Buscar usuario						
- Alertas Recientes							
- Alertas según Fecha	ÍNDICE	NOMBRE ↑	TELEFONO	HEADQUARTER	OCUPACION	REGISTROS	EMAIL
- Alertas según Tipo	1	Angelica Martin	11223389	France	Engineer	130	angelica.martin@accioncontraelhambre.org
- Alertas según Localización	2	Gerald Martin	336699887	Germany	Manager	6	gerald.martin@accioncontraelhambre.org
- Todas las alertas por Tipo	3	Giovanni Li	987654321	Spain	HHR	59	giovanni.lima@accioncontraelhambre.org
- Todas las alertas por Localización	4	Jessica Martin	123654789	Spain	SSTD	45	jessica.martin@accioncontraelhambre.org
	5	Laura Perez	665665665	Spain	PDO	1	laura.perez@accioncontraelhambre.org
	Items per page: 5 1 - 5 of 7 < >						

Figura 25: Gestión de todos los usuarios en la base de datos

Menu	Action Against Hunger	PERFIL	ACERCA DE	CONTACTO	INICIAR SESIÓN			
Gestión de Contenidos	<b>GESTIÓN DE ALERTAS</b>							
- Uso de la app	Buscar alerta							
- Alertas Recientes								
- Alertas según Fecha	ÍNDICE	NOMBRE	TIPO_DE_CRISIS	HEAD_QUARTER	NIVEL_COMPLEJIDAD	NIVEL_SCALE	NIVEL_URGENCIA	REPORTADOR
- Alertas según Tipo	1	La crisis más prolongada	Protracted Crisis	Spain	4	3	2	giovanni.lima@accioncontraelhambre.org
- Alertas según Localización	2	Terremoto	Natural Disaster (geophysical)	Spain	2	2	1	angelica.martin@accioncontraelhambre.org
- Todas las alertas por Tipo	3	Revolución de bandas locales	Armed or Civil Conflict	France	1	1	1	giovanni.lima@accioncontraelhambre.org
- Todas las alertas por Localización	4	Crisis de saneamiento	Complex Crisis	Spain	2	2	2	giovanni.lima@accioncontraelhambre.org
	5	Help my people	Natural Disaster (meteorological)		4	2	3	giovanni.lima@accioncontraelhambre.org
	Items per page: 5 1 - 5 of 14 < >							

Figura 26: Gestión de todas las alertas en la base de datos

En el apartado de “anexos” guía de uso, se podrán ver con más detalle y detenimiento las características y funcionalidades de la gestión de usuarios y alertas. Además de otros gráficos de análisis estadístico.

## 8 CONCLUSIONES Y TRABAJO FUTURO

En éste capítulo final se expresa las conclusiones del proyecto actual y se mencionarán algunas características o funcionalidades que se podrían incorporar a la herramienta desarrollada para mejorar sus capacidades.

### 8.1 Conclusiones

Para realizar este proyecto se ha puesto en práctica algunos conceptos de ingeniería y desarrollo web, base de datos, programación estructural y en general las labores de investigación sobre los procedimientos a seguir cuando se produce una alerta de nivel crítico. Los puntos esenciales que se han tenido en cuenta han sido desarrollar una herramienta de apoyo para que los usuarios puedan gestionar mejor las altas de ciertos tipos de alertas que puedan producirse, y monitorizar cada cierto tiempo cómo están evolucionando los reportes a través de ésta herramienta.

En definitiva, las funcionalidades de este proyecto sólo son una pequeña parte de lo que realmente los usuarios podrían realizar con la gestión de incidentes o alertas para focalizarse en las más importantes y tener un histórico de los reportes para analizarlos y actuar en medida y consecuencia.

## **8.2 Trabajo futuro**

A lo largo de la etapa de desarrollo del trabajo actual se han conseguido implementar la mayoría de las funcionalidades que se propusieron al inicio de esta etapa, sin embargo, existen muchos aspectos que se podrían plantear para complementar a este trabajo en un futuro:

### **Comunicación mediante un chat**

Una de las formas en que los usuarios puedan comunicarse, refiriéndose a alguna alerta en concreto, es mediante un sistema de chat. Así, con el intercambio de mensajes en texto los usuarios puedan ser mucho más eficaces a la hora de poder actuar en consecuencia sobre las alertas de nivel crítico.

### **Integración con la base de datos del cliente**

Actualmente el sistema dispone de una base de datos con prestación de servicios externos en la nube, en este caso Firebase de Google. Con lo que en un futuro podría ser más adecuado utilizar y conectar con los servicios que dispone directamente el cliente, en este caso Acción Contra El Hambre.

### **Obtener una localización más exacta**

Actualmente la herramienta cuenta con un servicio de localización utilizando librerías nativas de Ionic. Sin embargo, para obtener unas localizaciones más exactas hace falta mejorar el algoritmo de obtención e interpretación de esas coordenadas.

### **Traducción**

Una de las funcionalidades que no se pudo implementar por falta de tiempo fue la de traducción de la app en diferentes idiomas. Algo que en un futuro está contemplado en poder realizarse, al menos en tres idiomas: francés, inglés y español.

### **Sistema de notificaciones**

El envío y recepción de notificaciones es algo muy habitual en las herramientas de hoy en día, así que se podría integrar esta funcionalidad en un futuro.

## **Soporte y validación de contenidos**

Un de las tareas más importantes para el buen funcionamiento de una aplicación es el mantenimiento y soporte del mismo. Actualmente el administrador de la aplicación tiene permisos para manipular todos los contenidos de la base de datos, pero hace falta un gestor y soporte para validar los actuales contenidos de alertas. Ésta característica en un futuro se podría implementar, extendiéndolo de forma que se integre con el sistema de chat, mencionado antes.

## 8 CONCLUSIONS AND FUTURE WORK

In this final chapter, the conclusions of the current project are expressed and some characteristics or functionalities that could be incorporated into the tool developed to improve its capabilities will be mentioned.

### 8.1 Conclusions

Some concepts of engineering and web development, database, structural programming and the research work on the procedures have been used when a critical level alert occurs. The project has a support tool so that users can better manage the discharge of certain types of alerts that may occur, and monitor from time to time how reports are evolving through this tool.

So, the functionalities of this project are only a small part of what users could really do with the management of incidents or alerts to focus on the most important ones and have a history of the reports to analyse them and act on measure and consequence.

## **8.2 Future work**

Throughout the current work development stage, most of the functionalities that were proposed at the beginning of this stage have been implemented, however, there are many aspects that could be proposed to complement this work in the future:

### **Chat communication**

One of the ways in which users can communicate, referring to a specific alert, is through a chat system. Thus, with the exchange of text messages, users can be much more effective when it comes to acting on critical level alerts accordingly.

### **Client database**

Currently the system has a database with provision of external services in the cloud, in this case Google Firebase. With what in the future it might be more appropriate to use and connect with the services available directly to the client, in this case Action Against Hunger.

### **Accurate location**

Currently the tool has a location service using native Ionic libraries. However, in order to be more accurate, the algorithm needs to be improved.

### **Translation**

One of the features that could not be implemented due to lack of time was the translation of the app in different languages. Something that in the future is contemplated in being able to be done, in at least three languages: French, English and Spanish.

### **Notification System**

Sending and receiving notifications is very common in today's tools, so this functionality could be integrated in the future.

### **Content support and validation**

One of the most important tasks for the application is its maintenance and support. Currently, the application administrator has permissions to manipulate all the contents of the database, but a manager and support are needed to validate the current alert contents. This feature could be implemented in the future, extending it in a way that integrates with the chat system, mentioned before.

## BIBLIOGRAFÍA

- «Alert Cops». Ministerio del Interior. [En línea]. disponible en:  
<https://alertcops.ses.mir.es/mialertcops/> [1]
- «Disaster Alert». Pacific Disaster Center (GLOBAL). [En línea]. disponible en:  
<https://www.pdc.org/apps/disaster-alert/> [2]
- «Spotlight – Incident Reporting». Safety Culture. [En línea]. disponible en:  
<https://safetyculture.com/spotlight/> [3]
- «Firebase». [En línea]. disponible en:  
<https://es.wikipedia.org/wiki/Firebase> [4]
- «Git» [En línea]. disponible en:  
<https://git-scm.com/about> [5]
- «Firestore». [En línea]. disponible en:  
<https://firebase.google.com/docs/firestore> [6]
- «CRUD» [En línea]. disponible en:  
<https://es.wikipedia.org/wiki/CRUD> [7]
- «Material Design» [En línea]. disponible en:  
<https://v6.angular.io/docs> [8]
- «Angular» [En línea]. disponible en:  
<https://angular.io> [9]
- «Ionic» [En línea]. disponible en:  
<https://ionicframework.com/docs/> [10]
- «HTML» [En línea]. disponible en:  
<https://developer.mozilla.org/es/docs/Web/HTML> [11]
- «API Firebase» [En línea]. disponible en:  
<https://firebase.google.com/docs/reference> [12]

# ANEXOS

## ANEXO I – GUÍA DE USO

La finalidad de este anexo es ayudar al usuario a utilizar la herramienta, facilitándole un manual de uso en el que se explican las principales funcionalidades junto con las vistas de cada una de las interfaces que estén en juego.

### APP MÓVIL

#### Iniciar sesión

Nada más abrir la aplicación el usuario se encontrará con la pantalla de “Iniciar sesión”, ya que es imprescindible para continuar interactuando con las demás funcionalidades. En la figura 27 se puede apreciar una interfaz en la que el usuario deberá introducir sus credenciales e inmediatamente después de iniciar sesión es redirigido a la pantalla de “menú principal de alertas”.

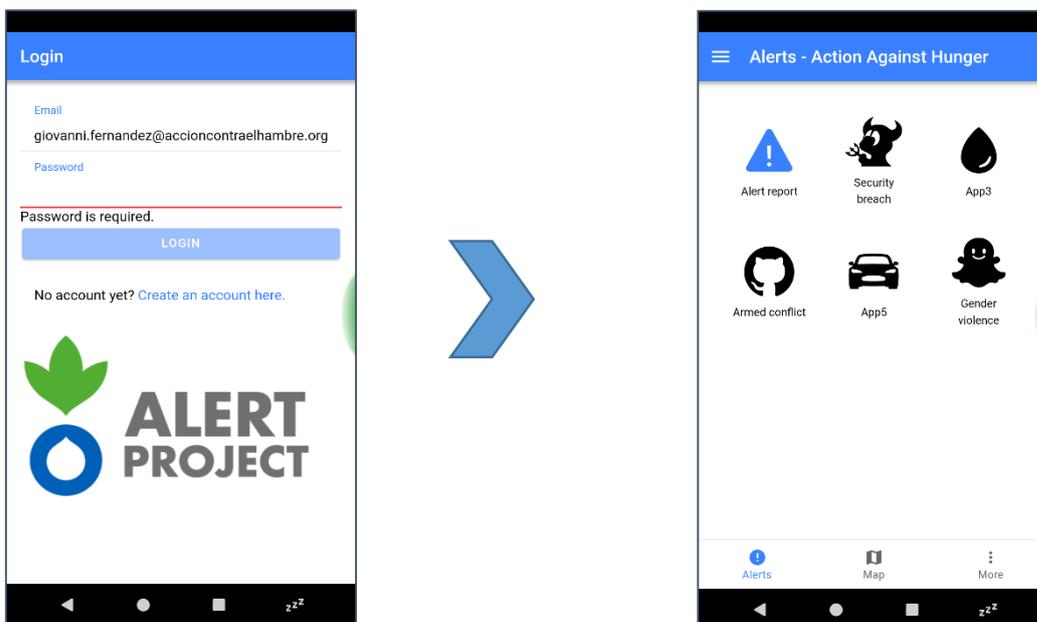


Figura 27: Pantalla – Iniciar sesión

## Crear Cuenta

En el caso de que el usuario quiera registrarse y crear una cuenta nueva, lo puede hacer en la pantalla “Crear cuenta”, tal como se muestra en la figura 28 se tendrá que rellenar todos los campos del formulario para finalizar el proceso. Una vez se haya creado la cuenta correctamente se redirige al usuario a la pantalla de “iniciar sesión” (figura 27) para que pueda proceder a iniciar sesión en la aplicación.

*Figura 28: Pantalla – Crear cuenta*

## Barra del menú inferior

En la barra inferior de la herramienta se encuentran los enlaces para acceder a las pantallas “Menú Principal de alertas” figura 29, “Mapa con localización GPS” figura 30 (en éste el que se puede consultar la localización actual del usuario como mero hecho informativo), y la pantalla “Más opciones” figura 31.

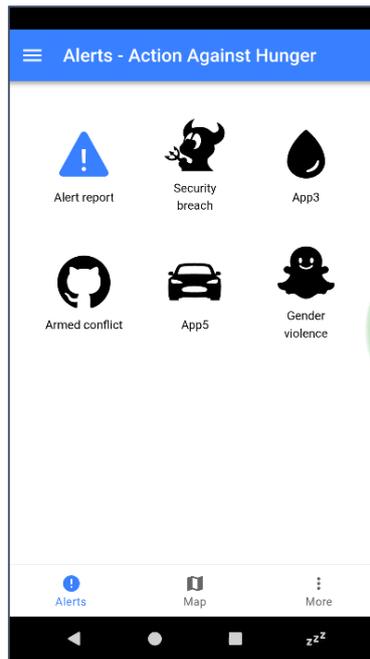


Figura 29: Pantalla – Menú principal

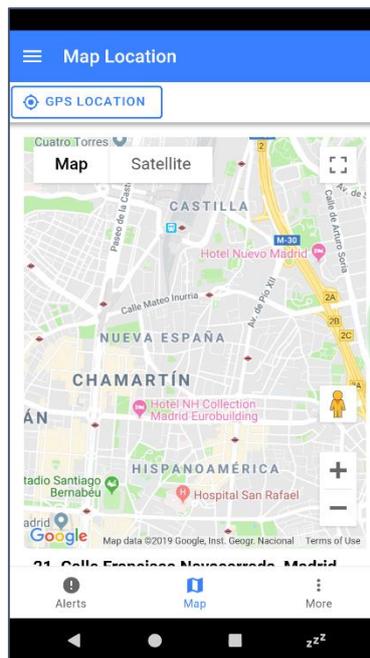


Figura 30: Pantalla – Mapa con localización GPS

Como se puede apreciar en la figura 31 se muestra los datos básicos de *acerca de la aplicación* y una pequeña guía de consejos y ayuda que tiene como finalidad el de facilitar al usuario a familiarizarse con la herramienta. Éste manual se puede con más detalles en la figura 32. Donde se podrá abrir un archivo pdf con una guía de usuario.

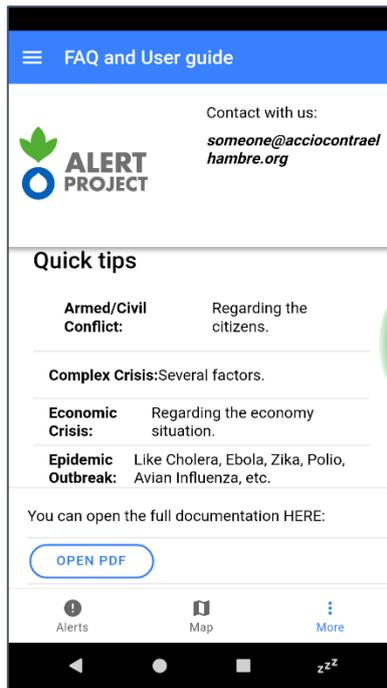


Figura 31: Pantalla – Más opciones

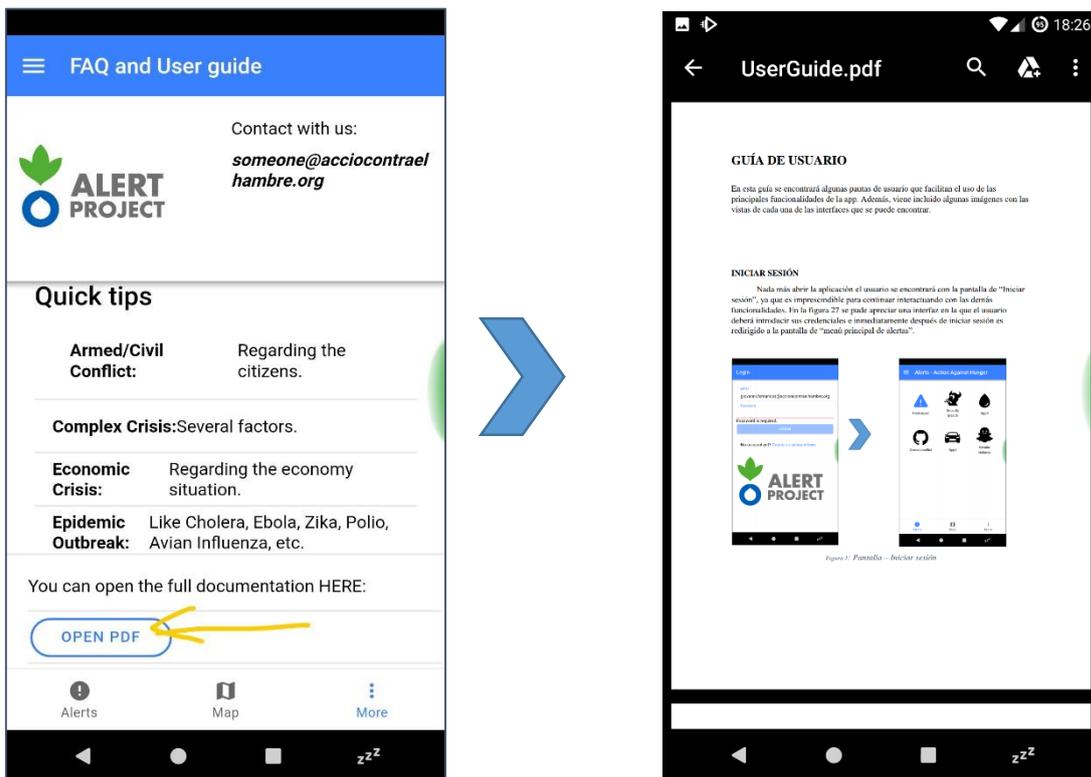


Figura 32: Pantalla – Guía de usuario

Ahora bien, para navegar a través otras funcionalidades dentro la herramienta, existe una pestaña de menú lateral en la parte superior izquierda (como se indica en la figura 33) que se despliega en forma de menú con enlaces, a través del cual se puede acceder al resto de funcionalidades.

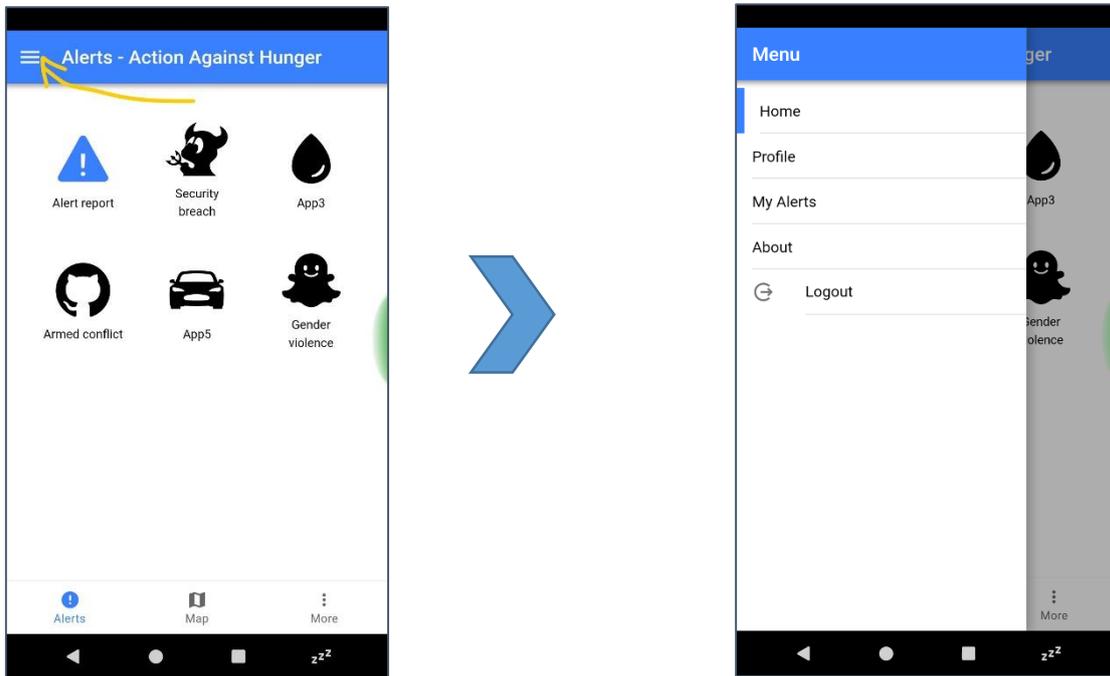


Figura 33: Pantalla – Barra lateral de opciones del menú

### Alta de alerta

Para el alta de una nueva alerta hay que acceder al formulario de alertas, que consiste en una serie de *slides* para rellenar los campos y datos principales tiene una alerta. En las figuras 34 a 40 se aprecian cada uno de los slides del formulario. En cada figura se detalla sus funciones.

Selección del país (Head Quarter al que pertenecerá):

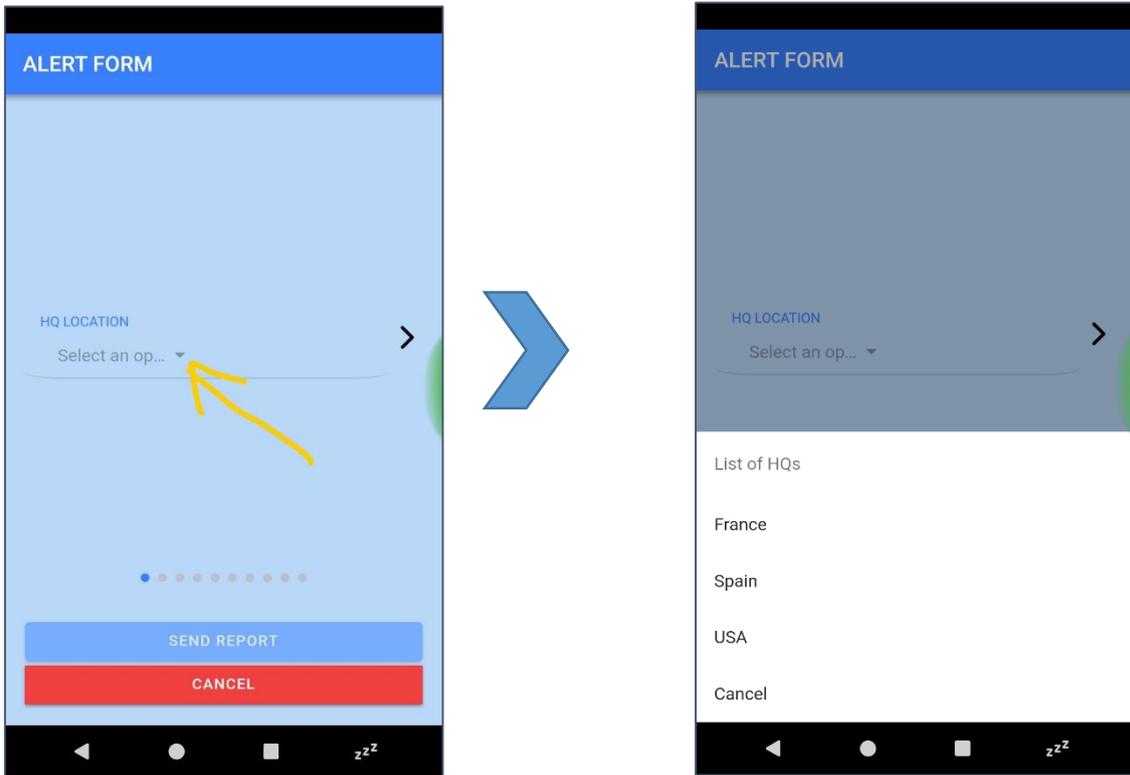


Figura 34: Pantalla – Formulario de alerta (slide\_1)

Selección del tipo de crisis:

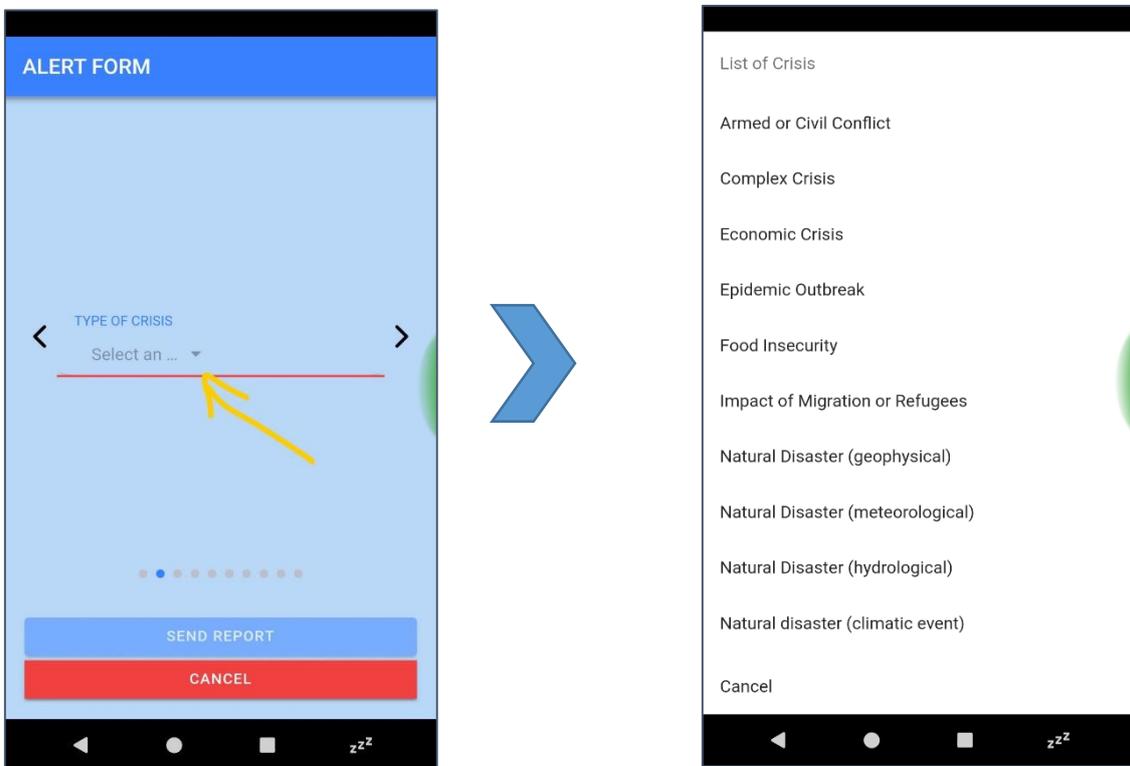


Figura 35: Pantalla – Formulario de alerta (slide\_2)

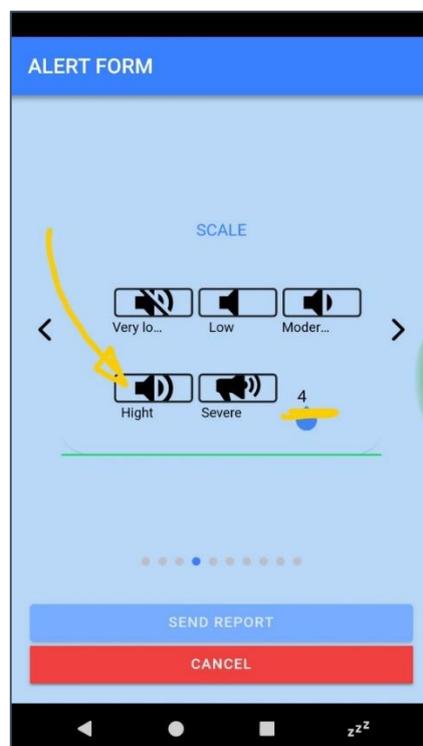
Nombre de alerta:



The screenshot shows a mobile application interface titled "ALERT FORM". At the top, there is a blue header with the text "ALERT FORM". Below the header, there is a light blue background. In the center, there is a text input field with the placeholder text "Name of the alert". Above the input field, the word "NAME" is displayed in blue. To the left and right of the input field are navigation arrows. Below the input field, there is a horizontal line. At the bottom of the screen, there are two buttons: a blue button labeled "SEND REPORT" and a red button labeled "CANCEL". The Android navigation bar is visible at the very bottom.

Figura 36: Pantalla – Formulario de alerta (slide\_3)

Nivel de escala (pulsar cualquier botón [reflejan valores de 1 al 5], del menos severo al más severo):



The screenshot shows the same "ALERT FORM" interface, but now the "SCALE" section is active. The word "SCALE" is centered above five selection buttons. Each button contains an icon of a speaker with a different level of sound waves. The buttons are labeled "Very lo...", "Low", "Moder...", "Hight", and "Severe". A yellow arrow points to the "Hight" button. To the right of the "Severe" button, the number "4" is displayed. Below the selection buttons, there is a horizontal line. At the bottom of the screen, the "SEND REPORT" and "CANCEL" buttons are visible. The Android navigation bar is at the bottom.

Figura 37: Pantalla – Formulario de alerta (slide\_4)

Nivel de complejidad (valores del 1 al 5, del menos crítico al más crítico)

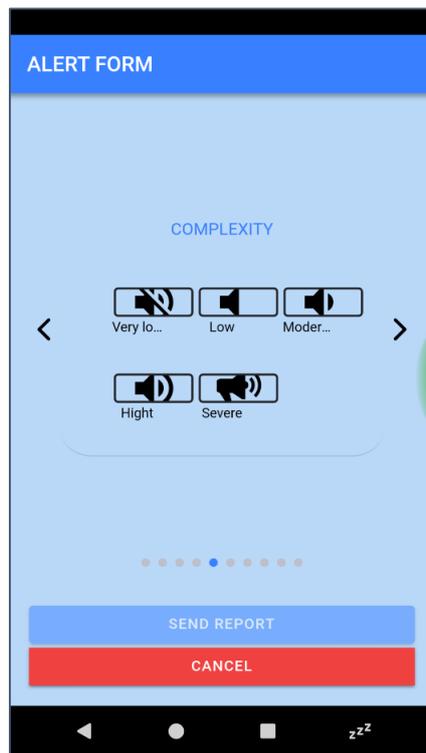


Figura 38: Pantalla – Formulario de alerta (slide\_5)

Nivel de urgencia (valores del 1 al 5, del menos crítico al más crítico)

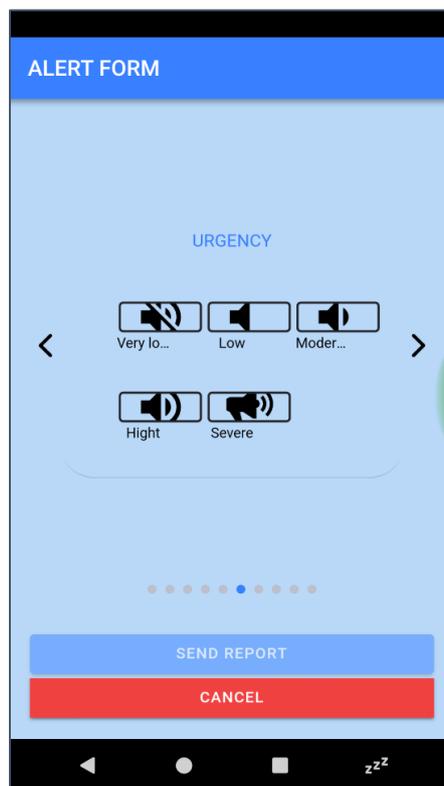


Figura 39 Pantalla – Formulario de alerta (slide\_6)

Población afectada (considerando el número de personas o familias afectadas, y el porcentaje de la población afectada):

ALERT FORM

AFFECTED POPULATION

Persons or families: 13 familias

%Population: %Percentage

SEND REPORT

CANCEL

*Figura 40: Pantalla – Formulario de alerta (slide\_7)*

Localización de zona afectada: En el que se despliega un mapa con un buscador que tiene función de autocompletar según se escribe en el campo de búsqueda. Una vez se selecciona la localización deseada se marca automáticamente con un ping de señalización sobre localización exacta, después se muestra un mensaje con el nombre de lugar que se había buscado. Como se aprecia en la figura 41 las acciones que puede hacer el usuario están marcadas.

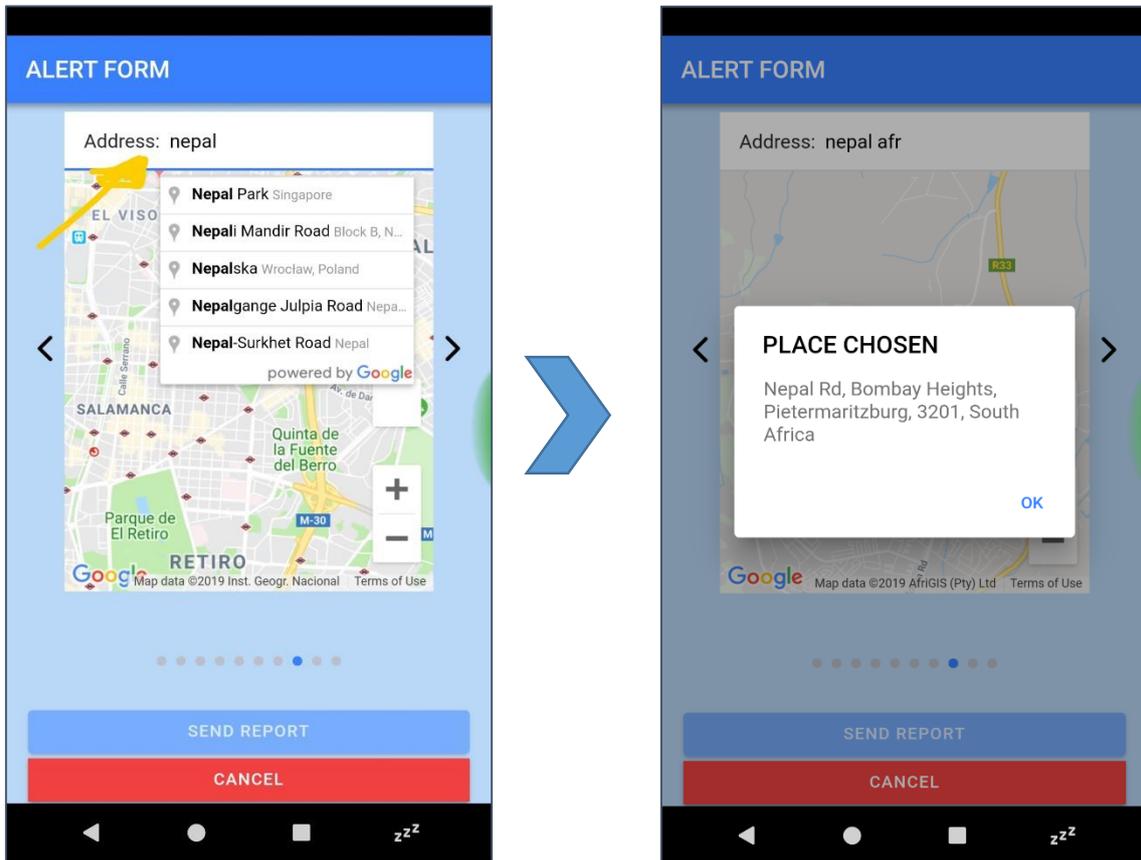


Figura 41: Pantalla – Formulario de alerta (slide\_8)

Subir una imagen o fichero de texto (se puede subir una imagen desde el dispositivo o utilizar la cámara desde la misma aplicación)

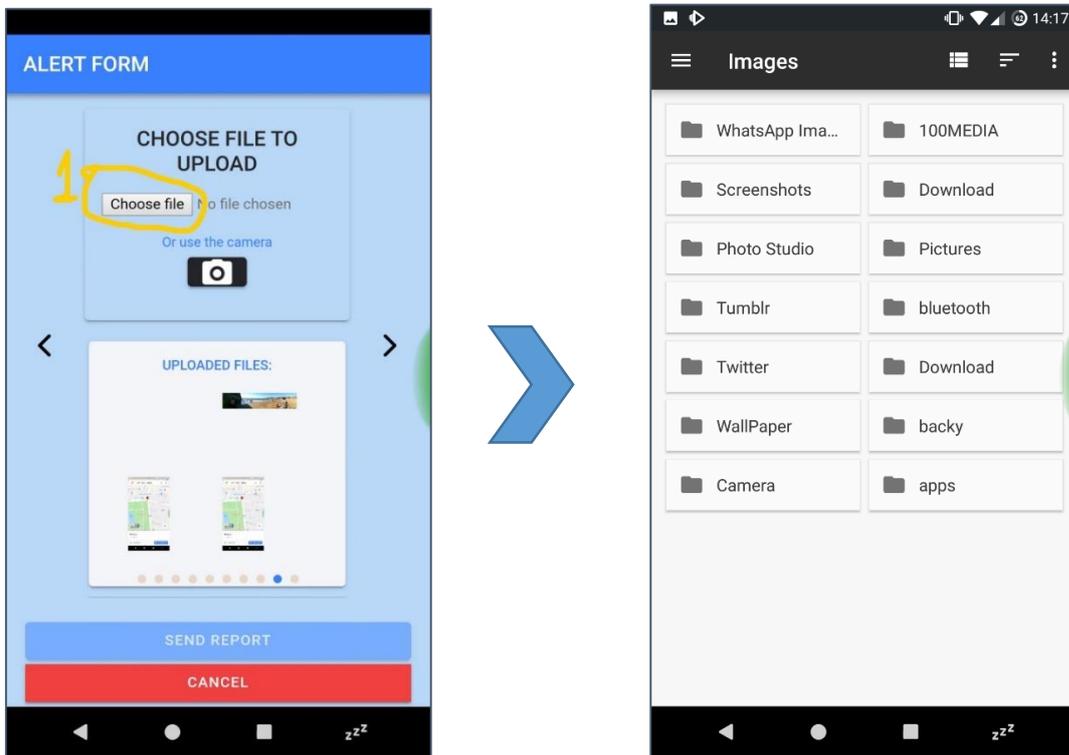


Figura 42: Pantalla – Formulario de alerta (slide\_9)

Además de podrá visualizar el progreso de la subida de la imagen y el tamaño total del mismo. Se detallan en las siguientes figuras:

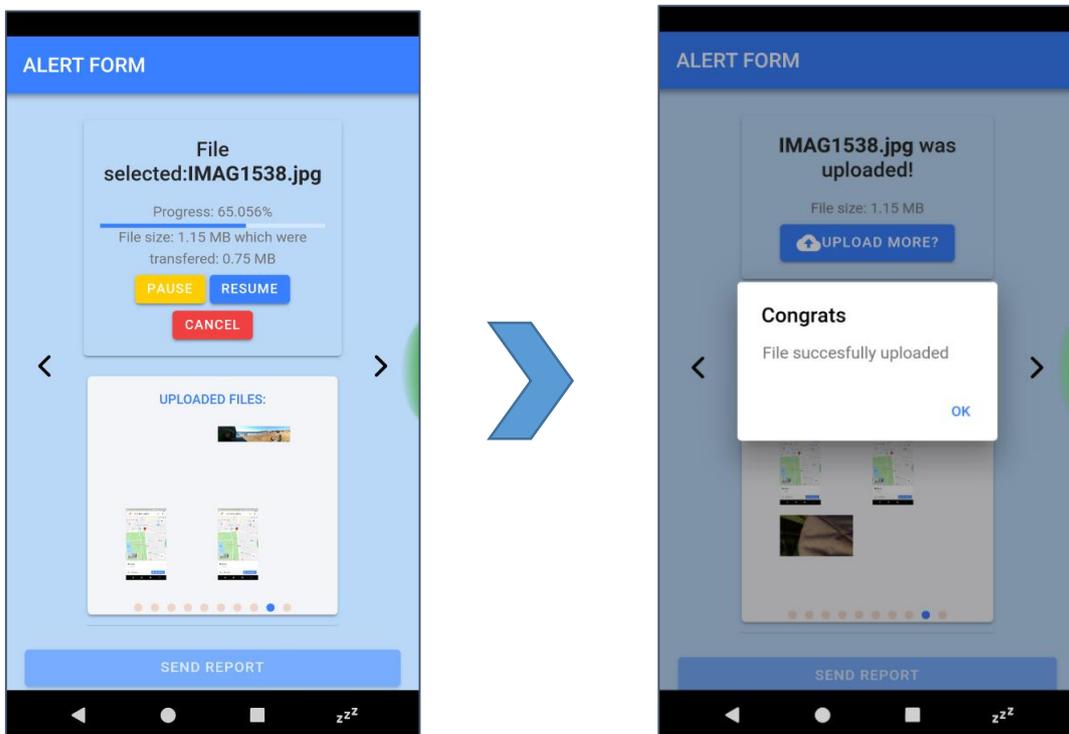


Figura 43: Pantalla – Formulario de alertas (slide\_9). Progreso de carga de imagen

Descripción de la alerta (campo de texto)

ALERT FORM

DESCRIPTION

< Enter description

SEND REPORT

CANCEL

*Figura 44: Pantalla – Formulario de alerta (slide\_10)*

## Modificar mi perfil

Un usuario puede modificar sus datos personales, tales como el nombre, correo electrónico, género, país etc. Para ello se debe pulsar sobre la pestaña del menú lateral (figura 33) y seleccionar "Perfil". Para editar el perfil de usuario se debe pulsar sobre el icono de edición como se muestra en la figura 45, inmediatamente se mostrarán los datos del perfil de usuario, que son modificables, y una vez hecho los cambios oportunos proceder a actualizar los datos, o en caso contrario cancelar la acción.

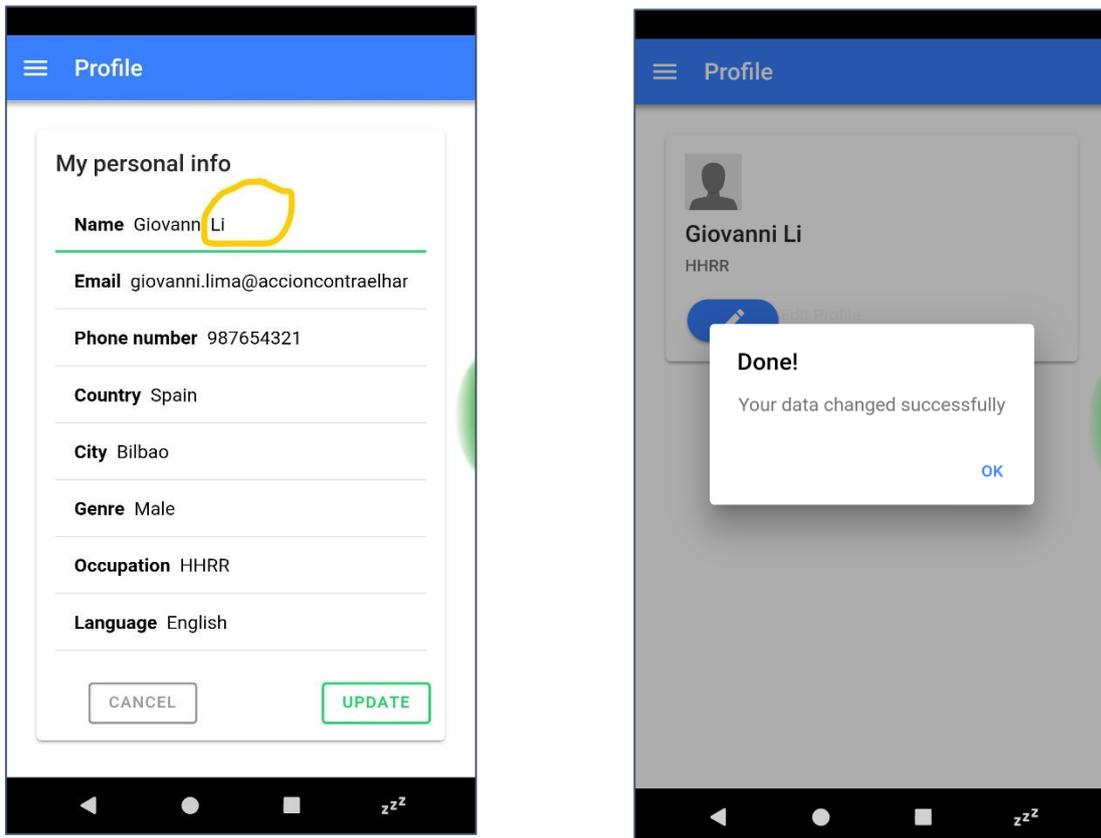


Figura 45: Pantalla – Modificar perfil

## Ver mis alertas

Las alertas que ha creado el usuario están almacenadas en la base de datos y asociadas únicamente al usuario. Para acceder a esta pantalla previamente se debe seleccionar el menú lateral (figura 33).

Como se aprecia en la figura 46 las alertas del usuario están listadas por el nombre de cada alerta. Y se puede desplegar cada una mostrando así sus características e información.

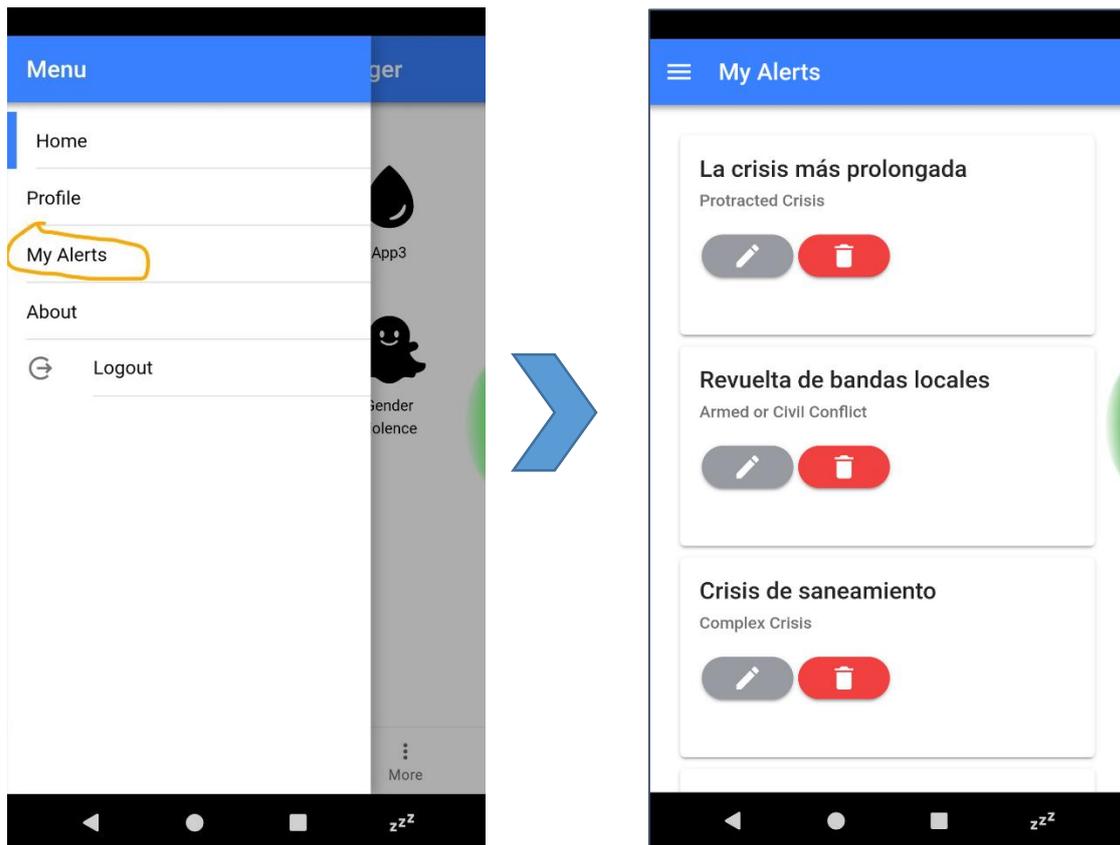


Figura 46: Pantalla – Visualizar Alertas

## Modificar alerta

Para modificar los datos de una alerta específica se debe pulsar sobre el icono de edición, como se muestra en la figura 47 y proseguir con los nuevos cambios que se quiera reflejar y finalmente confirmar pulsando el botón "actualizar".

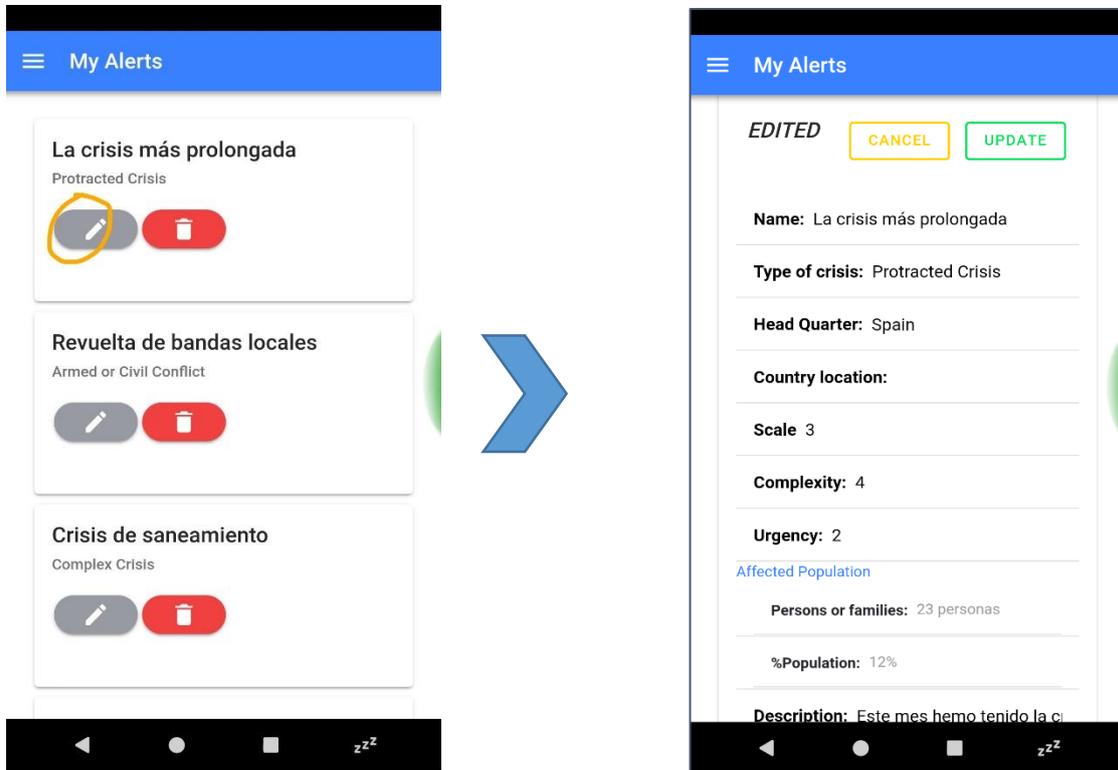


Figura 47: Pantalla – Modificar alerta

## APLICACIÓN WEB

### Iniciar sesión

Para poder gestionar los datos que están almacenados en la base de datos, sólo el usuario con permisos de administrador podrá realizar estas acciones habiendo accedido con sus credenciales previamente a la plataforma. En la anterior figura 48 se puede ver la interfaz de inicio de sesión.

Menu	Action Against Hunger	PERFIL	ACERCA DE	CONTACTO	INICIAR SESIÓN
Gestión de Contenidos	<h3>Accede a tu cuenta</h3> <p>Email *</p> <p><input type="text" value="giovanni.lima@accioncontraelhambre.org"/></p> <p>Password *</p> <p><input type="password" value="....."/></p> <p><input type="button" value="INICIAR SESIÓN"/></p>				
- Uso de la app					
- Alertas Recientes					
- Alertas según Fecha					
- Alertas según Tipo					
- Alertas según Localización					
- Todas las alertas por Tipo					
- Todas las alertas por Localización					

Figura 48: Pantalla – Iniciar sesión

### Ver y editar perfil

Los datos del usuario administrador pueden ser editados y modificados. Como se puede ver en la figura 49 los datos personales del usuario administrador están en un formulario con edición disponible.

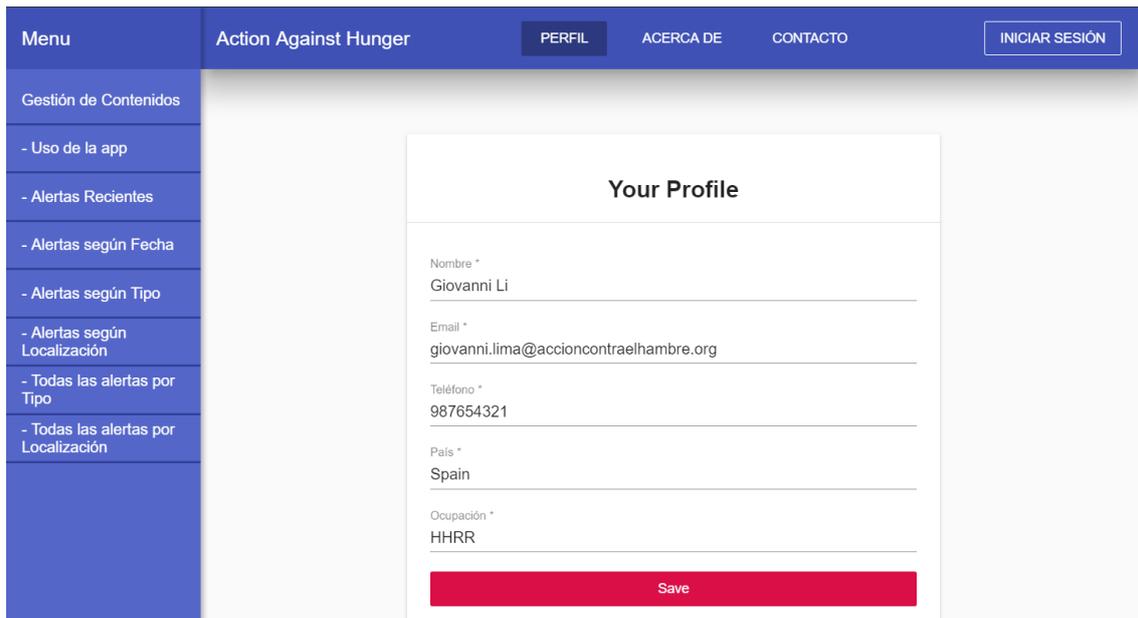


Figura 49: Pantalla – Ver y editar perfil

## Reporte con gráficos estadísticos

Para poder acceder a la visualización de gráficos estadísticos se debe pulsar sobre del menú lateral izquierdo. Como se puede apreciar en la figura 50, todos los enlaces a los distintos gráficos estadísticos están agrupados en este menú.

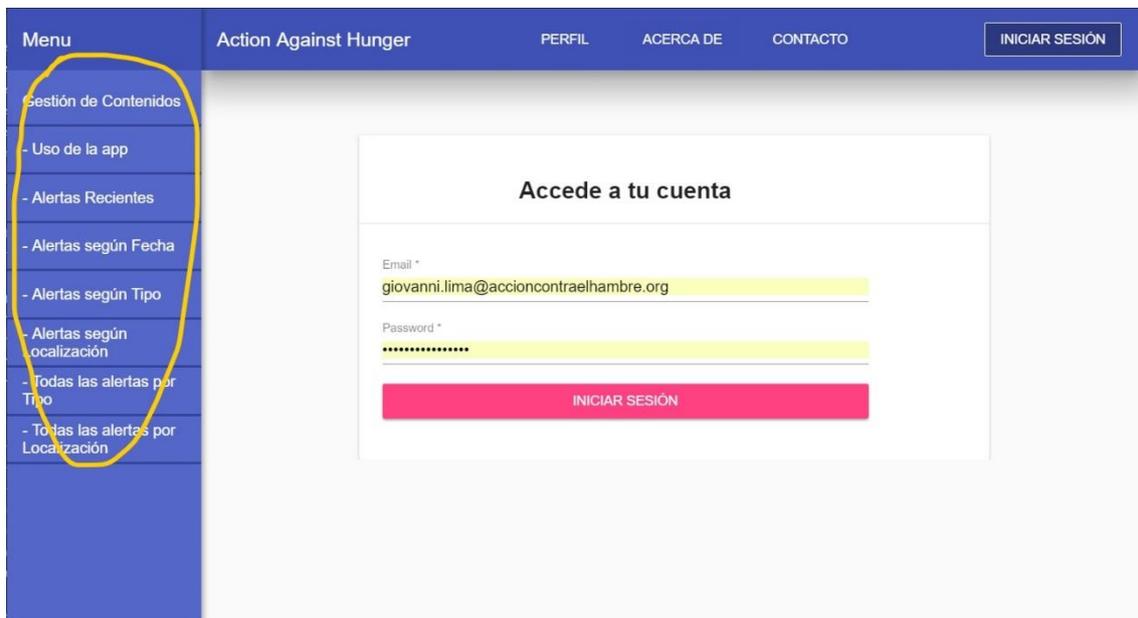


Figura 50: Pantalla – Menú lateral

Cada enlace lleva a una página de reporte estadístico y las gráficas tienen la opción de interactuar con ellas, así como poder descargarlas localmente en formato

de imagen. En las figuras 51, 52 y 53 se muestra algunos ejemplos de interacción sobre las gráficas mencionadas.

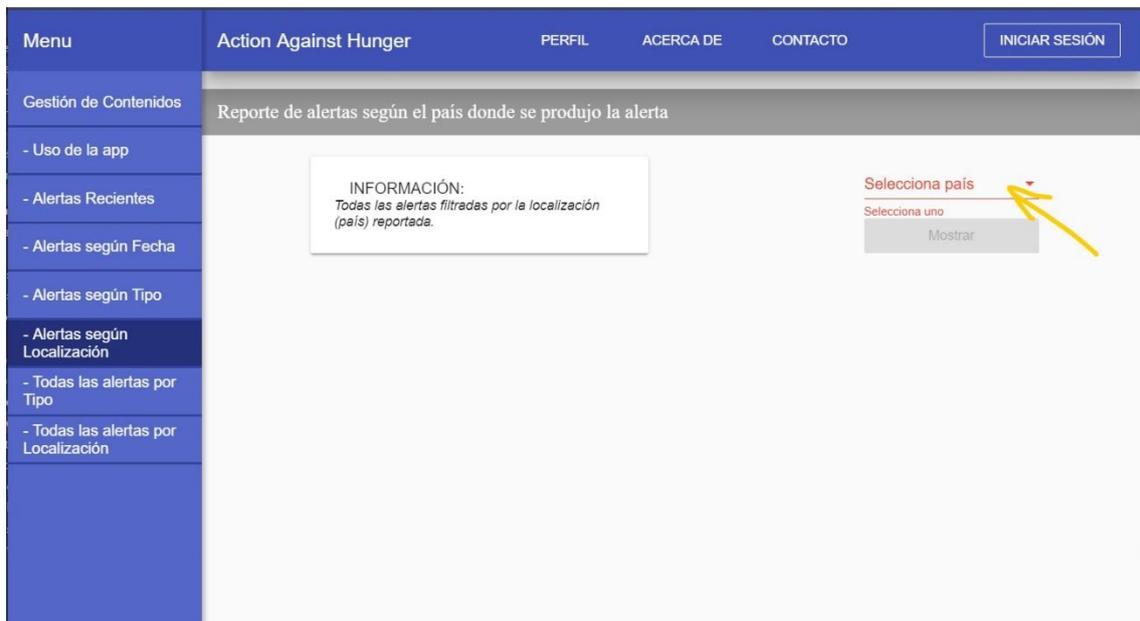


Figura 51: Pantalla – Selección de una las distintas opciones

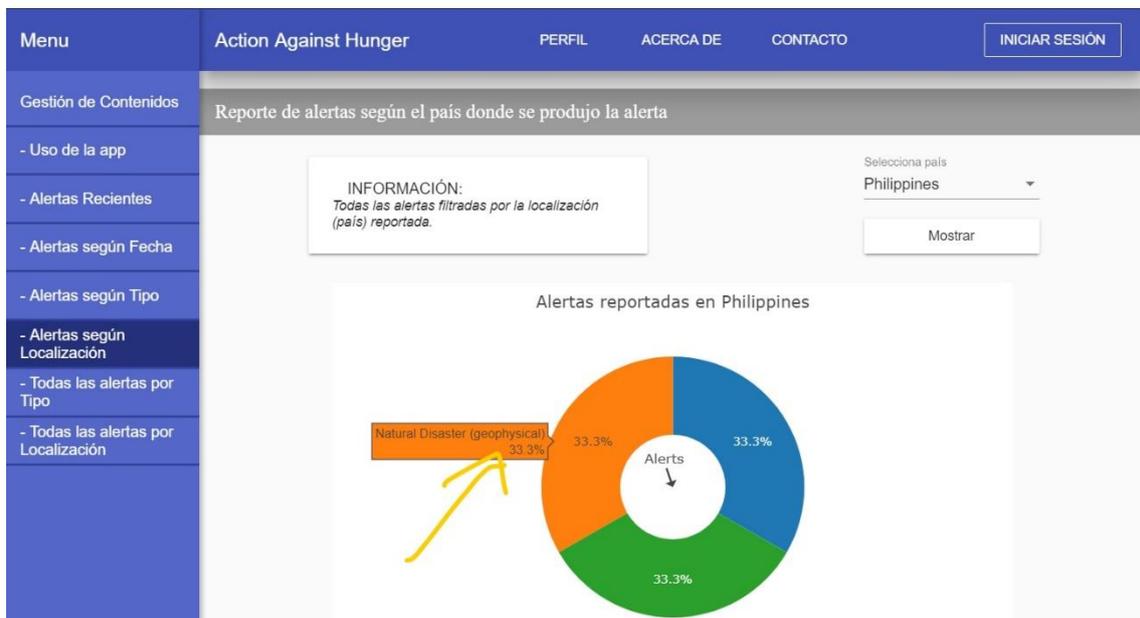


Figura 52: Pantalla – Interacción sobre las gráficas

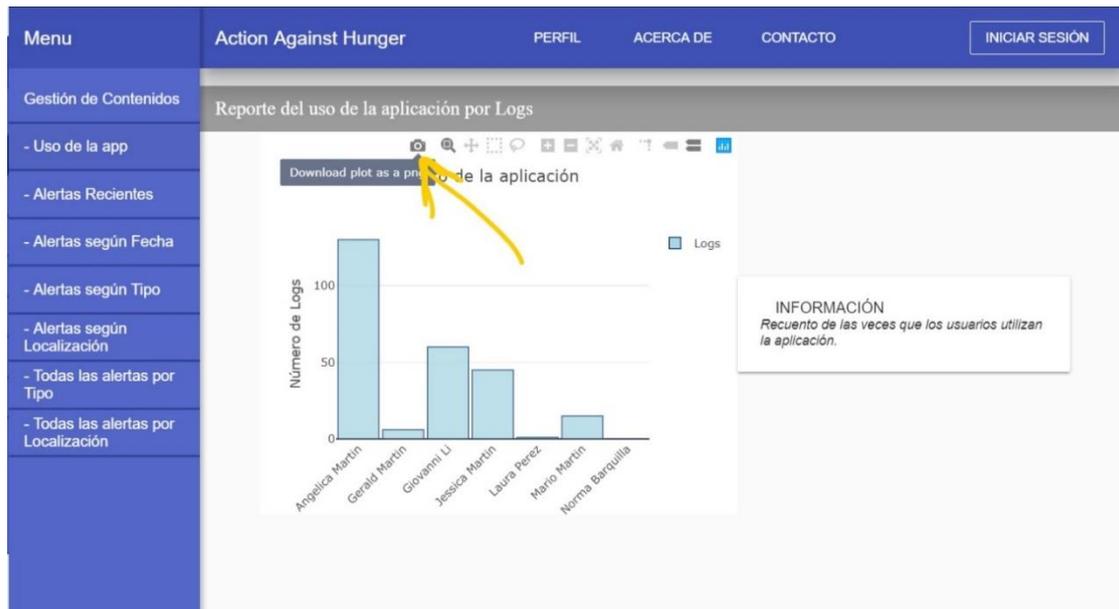


Figura 53: Pantalla – Exportar la gráfica actual localmente y en formato imagen

## Gestionar usuarios

Una de las principales funciones del administrador es gestionar a los usuarios que están registrados en la base de datos de la aplicación. En la figura 54 se puede apreciar la pantalla de administración con una tabla avanzada de todos los usuarios registrados. En cada celda se puede desplegar un pequeño contenedor con más información sobre un usuario en concreto, para ello se debe pulsar sobre dicha celda como se muestra en la figura 55 y además se tiene la opción de eliminar al usuario seleccionado.

Menu	Action Against Hunger	PERFIL	ACERCA DE	CONTACTO	INICIAR SESIÓN		
Gestión de Contenidos	Buscar usuario						
- Uso de la app	ÍNDICE	NOMBRE	TELEFONO	HEADQUARTER	OCUPACION	REGISTROS	EMAIL
- Alertas Recientes	1	Angelica Martin	11223389	France	Engineer	130	angelica.martin@accioncontraelhambre.org
- Alertas según Fecha	2	Gerald Martin	336699887	Germany	Manager	6	gerald.martin@accioncontraelhambre.org
- Alertas según Tipo	3	Giovanni Li	987654321	Spain	HHRR	59	giovanni.lima@accioncontraelhambre.org
- Alertas según Localización	4	Jessica Martin	123654789	Spain	SSTD	45	jessica.martin@accioncontraelhambre.org
- Todas las alertas por Tipo	5	Laura Perez	665665665	Spain	PDO	1	laura.perez@accioncontraelhambre.org
- Todas las alertas por Localización	Items per page: 5 1 - 5 of 7 < >						

Figura 54: Pantalla – Tabla de gestión de usuarios (vista general)

Si se selecciona sobre un usuario se desplegará un panel incrustado con toda la información del respectivo usuario. En la figura 55 y 56 se muestra un ejemplo de la edición del perfil de un usuario.

Menu	Action Against Hunger	PERFIL	ACERCA DE	CONTACTO	INICIAR SESIÓN		
Gestión de Contenidos	Buscar usuario						
- Uso de la app	ÍNDICE	NOMBRE	TELEFONO	HEADQUARTER	OCUPACION	REGISTROS	EMAIL
- Alertas Recientes	1	Angelica Martin	11223389	France	Engineer	130	angelica.martin@accioncontraelhambre.org
- Alertas según Fecha	2	Gerald Martin	336699887	Germany	Manager	6	gerald.martin@accioncontraelhambre.org
- Alertas según Tipo	3	Giovanni Li	987654321	Spain	HHR	59	giovanni.lima@accioncontraelhambre.org
- Alertas según Localización	4	Jessica Martin	123654789	Spain	SSTD	45	jessica.martin@accioncontraelhambre.org
- Todas las alertas por Tipo	5	Laura Perez	665665665	Spain	PDO	1	laura.perez@accioncontraelhambre.org
- Todas las alertas por Localización	Items per page: 5 1 - 5 of 7 << < > >>						

Figura 55: Pantalla – Tabla de gestión de usuarios(vista seleccionar usuario)

Menu	Action Against Hunger	PERFIL	ACERCA DE	CONTACTO	INICIAR SESIÓN		
Gestión de Contenidos	Buscar usuario						
- Uso de la app	ÍNDICE	NOMBRE	TELEFONO	HEADQUARTER	OCUPACION	REGISTROS	EMAIL
- Alertas Recientes	1	Angelica Martin	11223389	France	Engineer	130	angelica.martin@accioncontraelhambre.org
- Alertas según Fecha	<div style="border: 1px solid gray; padding: 5px;"> <p>Nombre * Angelica Martin Género * Female</p> <p>Email * angelica.martin@accioncontraelhambre.org Teléfono * 11223389</p> <p>Head Quarter * France Ciudad * Paris</p> <p>Ocupación * Engineer</p> <p style="text-align: right;"> <input type="button" value="Save"/> <input type="button" value="Eliminar perfil"/> </p> </div>						
- Alertas según Tipo	2	Gerald Martin	336699887	Germany	Manager	6	gerald.martin@accioncontraelhambre.org
- Alertas según Localización	3	Giovanni Li	987654321	Spain	HHR	59	giovanni.lima@accioncontraelhambre.org
- Todas las alertas por Tipo							
- Todas las alertas por Localización							

Figura 56: Pantalla – Tabla de gestión de usuarios (más opciones)

Además, dispone de una de buscador para aplicar un filtro de texto y buscar a un usuario específico mediante su nombre, correo electrónico o algún dato personal.

ÍNDICE	NOMBRE	TELEFONO	HEADQUARTER	OCUPACION	REGISTROS	EMAIL
5	Laura Perez	665665665	Spain	PDO	1	laura.perez@accioncontraelhambre.org

Figura 57: Pantalla – Tabla de gestión de usuarios (con filtro de)

Como se muestra en la figura 58 también es posible orden la tabla según la columna deseada para mostrar los elementos ordenados alfabéticamente de forma ascendente o descendente.

ÍNDICE	NOMBRE	TELEFONO	HEADQUARTER	OCUPACION ↑	REGISTROS	EMAIL
7	Norma Barquilla	661662663	France	Administration0		norma.barquilla@actionagainsthunger.org
6	Mario Martin	666666663	Spain	DevOps	15	mario.martin@accioncontraelhambre.org
1	Angelica Martin	11223389	France	Engineer	130	angelica.martin@accioncontraelhambre.org
3	Giovanni Li	987654321	Spain	HHR	59	giovanni.ima@accioncontraelhambre.org
2	Gerald Martin	336699887	Germany	Manager	6	gerald.martin@accioncontraelhambre.org

Figura 58: Pantalla – Tabla de gestión de usuarios (orden por columnas)

## Gestionar alertas

De igual forma todas las alertas son gestionadas por el usuario administrador y estas alertas se agrupan en otra tabla avanzada en la que se mostrará en tiempo real toda la información actual de las alertas que están en la base de datos. Como se aprecia en la figura 59 en el momento de hacer algún cambio de las alertas en la base de datos inmediatamente se reflejarán los cambios en la vista de la tabla de administración. Las acciones y operaciones en la tabla son las mismas que la tabla anterior de gestión de usuarios.

Menu	Action Against Hunger	PERFIL	ACERCA DE	CONTACTO	INICIAR SESIÓN			
Gestión de Contenidos	<b>GESTIÓN DE ALERTAS</b>							
- Uso de la app	Buscar alerta							
- Alertas Recientes								
- Alertas según Fecha	ÍNDICE	NOMBRE	TIPO_DE_CRISIS	HEAD_QUARTER	NIVEL_COMPLEJIDAD	NIVEL_SCALE	NIVEL_URGENCIA	REPORTADOR
- Alertas según Tipo	1	La crisis más prolongada	Protracted Crisis	Spain	4	3	2	giovanni.lima@accioncontraelhar
- Alertas según Localización	2	Terremoto	Natural Disaster (geophysical)	Spain	2	2	1	angelica.martin@accioncontraelh
- Todas las alertas por Tipo	3	Revolta de bandas locales	Armed or Civil Conflict	France	1	1	1	giovanni.lima@accioncontraelhar
- Todas las alertas por Localización	4	Crisis de saneamiento	Complex Crisis	Spain	2	2	2	giovanni.lima@accioncontraelhar
	5	Help my people	Natural Disaster (meteorological)		4	2	3	giovanni.lima@accioncontraelhar
	Items per page: 5 1 - 5 of 14 < < > >							

Figura 59: Pantalla – Tabla de gestión de alertas

## ANEXO II – GUÍA DE INSTALACIÓN

### Instalación de la app móvil

Para poder usar la app, lo primero es tener un dispositivo con sistema operativo Android, se recomienda tener la versión 5.x (Android Lollipop) en adelante.

Ahora debemos acceder a la configuración de ajustes del dispositivo y seleccionar el apartado de *Seguridad* y habilitar la opción de *orígenes desconocidos*.

*Ajustes -> Seguridad -> Orígenes desconocidos*

La primera vez que se inicia la app solicitará algunos permisos del dispositivo para su funcionamiento como son: *la activación del GPS, almacenamiento interno y acceso a la cámara*.

### Instalación de la aplicación web

Los requisitos básicos para poder desplegar la aplicación están ligados a unos prerrequisitos de compilación del código fuente. Pues bien, se debe tener instalado un entorno de ejecución para Javascript, el cual es Node.js con la versión 8.x.

Se debe instalar el gestor de paquetes npm, en caso de que no venga incluido por defecto con Node.js. Desde una terminal puedes comprobar si se han instalado correctamente las versiones de Node.j y NPM con los comandos:

```
$ node -v
```

```
$ npm -v
```

Lo siguiente es acceder desde una terminal (o consola de comandos) a la carpeta donde está el proyecto web de angular en instalar todas las dependencias necesarias para su funcionamiento con el comando:

```
$ npm install
```

Por último, para arrancar el servidor se debe introducir el siguiente comando:

```
$ ng serve -o
```