Gestión de Incidencias



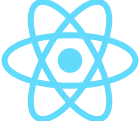
**GRADO DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**CURSO 23/24**

INCIDENCIAS EN ESPAÑA



Autor: Alberto Escobar Rodríguez S2DAM



1 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

***INDICE***

**Capítulo 1. Introducción y objetivos**, 3

**Capítulo 2. Especificación de Requisitos**, 5

**Capítulo 3. Planificación Temporal y Evaluación de Costes**, 9 **Capítulo 4. Tecnologías Utilizadas**, 12

**Capítulo 5. Desarrollo e Implementación**, 29

**Capítulo 6. Conclusiones y líneas futuras**, 36

**Capítulo 7. Bibliografía.**, 40

2 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

**Capítulo 1. Introducción y objetivos**

**1.1. Introducción**

En un mundo cada vez más conectado y digitalizado, la eficiencia en la gestión de incidencias y el mantenimiento de las infraestructuras se vuelve fundamental para garantizar el bienestar y la seguridad de la comunidad. En este contexto, surge la necesidad de desarrollar una aplicación móvil dedicada a la gestión de incidencias en las provincias de España.

Este proyecto surge de la observación de una problemática recurrente en diversas zonas, específicamente en mi pueblo, Olías del Rey. La falta de una plataforma eficiente para reportar y dar seguimiento a incidencias en áreas públicas ha llevado a un deterioro progresivo de la calidad de vida de sus habitantes. Calles deterioradas, mobiliario urbano en mal estado y otros problemas similares son comunes en muchas zonas descuidadas, lo que afecta negativamente a la experiencia de los residentes y aquellos visitantes de la localidad.

Conscientes de esta realidad y motivados por el deseo de contribuir a la mejora del entorno urbano, se propone el desarrollo de una aplicación móvil que facilite la comunicación y el registro de incidencias, agilizando su resolución y fomentando la participación ciudadana en la conservación y el mantenimiento del espacio público.

El objetivo principal de esta aplicación es proporcionar una herramienta intuitiva y accesible que permita a los usuarios reportar de manera rápida y eficiente cualquier incidencia que detecten en su entorno, ya sea un bache en la carretera, una farola averiada o cualquier otro problema que requiera atención por parte de las autoridades pertinentes.

A lo largo de este documento, se detalla el proceso de desarrollo de la aplicación, desde la identificación de requisitos y la planificación del proyecto hasta la implementación y la evaluación de los resultados obtenidos. Además, se discutirán los beneficios de esta solución, así como las posibles implicaciones para la mejora de la calidad de vida en las zonas afectadas.

3 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

**1.2. Objetivo del Proyecto**

El principal objetivo de este proyecto es desarrollar una aplicación móvil destinada a la gestión de incidencias en las provincias de España. A través de esta aplicación, se busca alcanzar las siguientes metas:

-Facilitar la comunicación y el reporte de incidencias: La aplicación permitirá a los usuarios notificar de manera sencilla y rápida cualquier problema que identifiquen en su entorno, utilizando el teléfono. Se implementarán formularios intuitivos y funciones de geolocalización para agilizar el proceso de reporte.

-Agilizar la resolución de incidencias: Una vez reportada la incidencia, la aplicación gestionará automáticamente su registro y asignación a las autoridades competentes para su resolución. Se establecerán mecanismos de seguimiento y notificación para informar a los usuarios sobre el estado de sus reportes.

-Fomentar la participación ciudadana: La aplicación promoverá la participación de la comunidad en la identificación y solución de incidencias, incentivando la colaboración entre ciudadanos y entidades responsables. Se incluirán funciones de retroalimentación y valoración para involucrar a los usuarios en el proceso de mejora del entorno urbano.

-Mejorar la calidad de vida en las zonas afectadas: Se espera que la aplicación tenga un impacto positivo en la seguridad, el bienestar y la satisfacción de los ciudadanos con su entorno.

-Proporcionar una herramienta estable y adaptable: La aplicación se diseñará con una arquitectura modular y flexible, que permita su fácil adaptación a diferentes necesidades. Se establecerán prácticas de desarrollo ágiles y colaborativas para garantizar la escalabilidad y una evolución continua de la plataforma.

4 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

**Capítulo 2. Especificación de Requisitos**

En esta sección, se discutirán los requisitos que necesitan crearse para construir la aplicación móvil diseñada para informar incidencias en las provincias de España. Los requisitos son un elemento clave en la lista a seguir con el fin de que el proyecto logre sus objetivos, para que sea capaz de satisfacer las necesidades de los destinatarios. Se dividen en funcionales (RF) y no funcionales (RNF), lo que garantiza una descripción detallada de las capacidades y el comportamiento esperado de la plataforma.

**2.1. Requisitos funcionales (RF)**

1. Usabilidad

RF1.1: La aplicación debe ser intuitiva y fácil de usar, con una interfaz de usuario amigable. 

2. Compatibilidad



RNF2.1: La aplicación debe ser compatible con las versiones más recientes de los sistemas operativos *iOS* y *Android*. Al ser una aplicación desarrollada con *React Native* que permite la multiplataforma es sencillo que en ambos sistemas operativos funcionen, pero hay algunas disfuncionalidades con los nuevos dispositivos *IOS*.

5 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

RNF2.2: Debe ser accesible tanto en *smartphones* como en *tablets*. El diseño está pensado para dispositivos móviles con no mucha resolución.



3. Rendimiento

RF3.1: La aplicación debe ser capaz de manejar múltiples reportes simultáneamente sin afectar su desempeño.

RF3.2: Los tiempos de respuesta deben ser inferiores a 2 segundos para la mayoría de las operaciones.

4. Seguridad

RF4.1: La aplicación debe proteger los datos personales de los usuarios mediante encriptación. 

5. Registro y autenticación de usuarios

RF5.1. La aplicación debe permitir a los usuarios registrarse utilizando un formulario de registro.

RF5.2. Los usuarios deben poder autenticarse usando sus credenciales (dirección de correo electrónico y contraseña).



6 Alberto Escobar Rodríguez S2DAM

6. Reportar una incidencia

Gestión de Incidencias

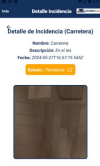
RF6.1. Los usuarios deben poder reportar una incidencia completando un formulario que consta de campos para descripción del problema, nombre de la incidencia y estado de la incidencia.

RF6.2. La aplicación debe permitir a los usuarios adjuntar fotos a sus informes de incidencias. RF6.3. La aplicación debe permitir a los usuarios especificar la ubicación de la incidencia. RF6.4. Fecha y hora de los informes. La fecha y la hora de un reporte de incidencia enviado se completan de manera automática.



7. Gestión de incidencias

RF7.1: Los usuarios deben poder ver el estado de sus reportes (pendiente, en proceso, resuelto, cancelado).

RF7.2: Las autoridades competentes deben poder actualizar el estado de las incidencias.

7 Alberto Escobar Rodríguez S2DAM

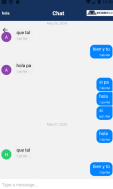
Gestión de Incidencias

8. Notificaciones y Seguimiento

RF8.2: La aplicación debe permitir a los usuarios seguir el progreso de las incidencias reportadas.

9. Valoración

RF9.1: La aplicación debe permitir a los usuarios comentar sobre la eficacia y la rapidez en la resolución de incidencias.



**2.2. Requisitos No Funcionales (RNF)**

1.Usabilidad

RNF1.1: Debe estar disponible en los idiomas oficiales de España (español, catalán, gallego, y vasco).

2.Mantenimiento y Escalabilidad

RNF2.1: La aplicación debe tener una arquitectura que facilite su mantenimiento y actualización.

RNF2.2: Debe ser escalable para soportar un creciente número de usuarios y reportes sin comprometer su rendimiento. Al estar conectada a una base de datos en *Firebase* sí que podría soportar un gran número de usuarios al momento, el problema es que en este proyecto se han intentado minimizar los costes, por lo que el plan básico de *Firebase* no permite tantas

8 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

solicitudes simultáneas. Como mejora futura se podría cambiar a una base de datos relacional ya sea *MySQL*.

4. Administración de la Aplicación

RF4.1: Los administradores deben poder gestionar usuarios, incidencias y categorías de incidencias. Cambios a futuro serían añadir diferentes roles de administración y de ciudadanía. RF4.2: La aplicación debe proporcionar estadísticas y reportes sobre las incidencias reportadas y resueltas.

**Capítulo 3. Planificación Temporal y Evaluación de Costes**

En este capítulo se presenta la planificación temporal y la evaluación de costes para el desarrollo de la aplicación móvil. Se describe en detalle el proceso de implementación y se estimaron los costos, considerando un escenario de un solo desarrollador durante 40 horas.

**3.1. Planificación Temporal**

La planificación temporal del proyecto se ha realizado utilizando un modelo de Gantt, dividiendo el proyecto en varias fases, cada una con tareas específicas:

| Fase | Tarea | Duración (horas) | Día 1 | Día 2 | Día 3 | Día 4 | Día 5< |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Análisis | Recolección de requisitos | 2 | X |  |  |  |  |
|  | Análisis de viabilidad | 1 | X |  |  |  |  |
| Diseño | Diseño de la arquitectura | 3 |  | X | X |  |  |
|  | Diseño de la interfaz de usuario | 3 |  |  | X | X |  |
| Desarrollo | Configuración del entorno de desarrollo | 2 |  |  |  | X |  |
|  | Desarrollo del backend | 8 |  |  |  | X | X |
|  | Desarrollo del frontend | 8 |  |  |  |  | X |
|  | Integración de servicios | 5 |  |  |  |  | X |

| Pruebas | Pruebas  unitarias y de integración | 4 |  |  |  |  | X |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pruebas de usabilidad | 2 |  |  |  |  | X |
| Documentación | Documentación del código y del proyecto | 2 |  |  |  |  | X |

**Análisis**

-Recolección de requisitos (2 horas): Identificación de necesidades y funcionalidades del proyecto. En este punto se analiza todo lo que se quiere hacer y los objetivos; aunque durante el camino vayan surgiendo nuevas metas.

-Análisis de viabilidad (1 hora): Evaluación de herramientas y tecnologías a utilizar. **Diseño**

-Diseño de la arquitectura (3 horas): Estructuración de la arquitectura del sistema y componentes.

-Diseño de la interfaz de usuario (3 horas): Creación de maquetas y diseño visual de la aplicación.

**Desarrollo**

-Configuración del entorno de desarrollo (2 horas): Instalación y configuración de herramientas y dependencias

-Desarrollo del *backend* (*Node.js*, *Firebase*) (8 horas): Implementación del servidor y la base de datos.

-Desarrollo del *frontend* (*React Native*, *Expo*) (8 horas): Implementación de la interfaz y la lógica de la aplicación.

-Integración de servicios (Autenticación, *API*) (5 horas): Conexión e integración con servicios externos.

**Pruebas**

-Pruebas unitarias y de integración (4 horas): Verificación de funcionalidad y corrección de errores.

-Pruebas de usabilidad (2 horas): Evaluación de la experiencia del usuario final.

10 Alberto Escobar Rodríguez S2DAM

**Documentación**

Gestión de Incidencias

-Documentación del código y del proyecto (2 horas): Creación de la documentación técnica y de usuario.

**3.2. Evaluación de Costes**

La evaluación de costes incluye el salario del desarrollador, licencias de software, y otros gastos relevantes. A continuación, se detallan los costos estimados:

| Concepto | Detalle | Costo |
| --- | --- | --- |
| Salario de Desarrollador |  |  |
| Salario promedio | 25€ por hora |  |
| Total horas trabajadas | 40 horas |  |
| Costo total del salario | 25€ \* 40 horas | 1000€ |
| Licencias de Software |  | 0€ |
| Expo | Gratuito para la mayoría de las funcionalidades. Puede haber costos adicionales si se utilizan características avanzadas. |  |
| Firebase | Gratuito con la cuota básica. Costos adicionales dependen del uso de servicios avanzados. |  |
| Visual Studio Code | Gratuito |  |
| Node.js | Gratuito |  |
| Total Licencias |  | 0€ |
| Hardware |  | 0€ |
| Uso de ordenador | Asumiendo que ya se dispone de un ordenador personal, no habrá costos adicionales. |  |
| Uso de dispositivos móviles | Asumiendo que ya se cuenta con dispositivos móviles, no habrá costos adicionales. |  |
| Total Hardware |  | 0€ |
| Gastos Generales |  | 30€ |
| Conexión a Internet | 20€ (estimación mensual prorrateada) |  |
| Electricidad | 10€ (estimación mensual prorrateada) |  |
| Total Gastos Generales |  | 1030€ |

11 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

| Resumen de Costes |  | 1000€ |
| --- | --- | --- |
| Salario de Desarrollador |  | 0€ |
| Licencias de Software |  | 0€ |
| Hardware |  | 30€ |
| Gastos Generales |  | 1030€ |
| Total |  |  |

El coste analizado es de 1030€ para el desarrollo e implantación de la aplicación móvil de gestión de incidencias. Este presupuesto considera que un desarrollador trabajó durante 40 horas; utiliza principalmente herramientas gratuitas y reduce los costos generales y de hardware al asumir los supuestos existentes. No hemos contado los gastos de los cursos realizados tanto de *React Native* como de otras funcionalidades.

**Capítulo 4. Tecnologías Utilizadas**

En este capítulo se presentan las tecnologías seleccionadas para el desarrollo de la aplicación móvil de gestión de incidencias. La elección de las tecnologías adecuadas desempeña un papel muy importante para el éxito del proyecto, ya que determina la eficiencia del desarrollo, la escalabilidad de la aplicación y la experiencia del usuario final.

**Lenguaje de Programación**

Se optó por utilizar *JavaScript* como el lenguaje principal de programación, aprovechando la amplia adopción en el desarrollo móvil y su versatilidad para la construcción de aplicaciones multiplataforma.



12 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

***Framework* de Desarrollo**

Para el desarrollo de la aplicación móvil, se decidió utilizar *React Native*, un *framework* de código abierto desarrollado por *Facebook* que permite crear aplicaciones nativas para *IOS* y *Android*. Utilizando *JavaScript* y *React*, *React Native* permite a los desarrolladores escribir un solo código base que puede ser ejecutado en ambas plataformas, lo que ahorra tiempo y recursos en el proceso de desarrollo.



-Desarrollo Multiplataforma: Una de las principales ventajas de *React Native* es la capacidad para permitir el desarrollo multiplataforma. Esto significa que los desarrolladores pueden crear aplicaciones tanto para *IOS* como para *Android* utilizando un único código base. Esta característica es especialmente beneficiosa para equipos de desarrollo que buscan maximizar su eficiencia y reducir el tiempo de desarrollo, ya que elimina la necesidad de escribir y mantener dos códigos por separado para cada plataforma. Además, este enfoque contribuye a una mejor consistencia en la experiencia del usuario entre los diferentes dispositivos.

-Componentes Nativos: *React Native* se distingue por su uso de componentes nativos, lo que aporta una experiencia de alta calidad al usuario. A diferencia de otras que utilizan tecnologías web encapsuladas (utilizan el cifrado en los archivos creados, descargados o contenidos por la aplicación, para que solo la aplicación pueda leer dichos archivos), *React Native* compila el código *JavaScript* a componentes nativos. Esto permite que las aplicaciones que han sido hechas con *React Native* ofrezcan un rendimiento comparable al rendimiento de las aplicaciones nativas, con interfaces de usuario suaves y tiempos de respuesta rápidos. Los componentes nativos también aseguran que la aplicación se integre fácilmente con las características especificas del sistema operativo.

13 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

-Hot Reloading: Otra característica destacada de *React Native* es su soporte para *“hot reloading”* o recarga en caliente. Esta funcionalidad facilita la prueba y la depuración del código, que permite ver los cambios a los programadores en tiempo real sin tener que volver a compilar el código. Con *hot reloading*, los desarrolladores pueden realizar ajustes y correcciones de manera rápida y muy eficiente, lo que acelera el desarrollo y permite iterar sobre el diseño y la funcionalidad de la aplicación.

-Gran Comunidad y Ecosistema: Además de las ventajas técnicas, *React Native* cuenta con una gran comunidad de desarrolladores y un ecosistema en constante crecimiento. Al ser un proyecto de código abierto, hay una amplia gama de recursos disponibles, incluyendo bibliotecas, *plugins* y herramientas que puedan integrarse fácilmente en el proyecto. Esta comunidad también proporciona soporte y comparte conocimientos, lo que puede servir para resolver problemas y encontrar soluciones innovadoras.

La elección de *React Native* para el desarrollo de la aplicación móvil se basa en su capacidad para facilitar el desarrollo multiplataforma, garantizar una experiencia de usuario de alta calidad a través de componentes nativos, y permitir una prueba y depuración eficientes gracias a la recarga en caliente. Además, el respaldo de una comunidad activa y un ecosistema robusto hace que *React Native* sea una opción ideal para desarrollar aplicaciones móviles modernas y eficientes.

**Plataforma de Desarrollo**

*Expo* se utilizó como plataforma de desarrollo para simplificar la creación y el despliegue de la aplicación móvil, permitiendo a los desarrolladores trabajar tanto con un simulador como con un dispositivo móvil personal. *Expo* ofrece un conjunto de herramientas y servicios adicionales que optimizan el desarrollo de aplicaciones en *React Native*, proporcionando una experiencia de desarrollo más fluida y eficiente. Entre sus características se incluyen la visualización de las modificaciones en tiempo real en dispositivos móviles y la gestión automatizada de certificados de seguridad. A continuación, se describen algunas de las características más destacadas de *Expo*:

14 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias



-Facilidad de Configuración:

*Expo* simplifica la configuración del entorno de desarrollo. A diferencia de la configuración tradicional de *React Native*, que requiere de múltiples pasos y configuraciones específicas para cada plataforma, *Expo* ofrece un entorno ya configurado que lo hace más sencillo. Esto permite a los desarrolladores comenzar a trabajar en sus proyectos de inmediato, sin perder tiempo en la configuración del entorno de desarrollo.

-Bibliotecas Incluidas:

*Expo* proporciona una amplia gama de *API* nativas y bibliotecas de componentes que están listas para usar. Esto incluye funcionalidades esenciales como el acceso a la cámara, la ubicación, las notificaciones *push*, y muchos otros servicios que pueden ser integrados fácilmente en la aplicación. Con estas bibliotecas, *Expo* ahorra a los desarrolladores el esfuerzo

15 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

de buscar e implementar soluciones de terceros, facilitando el desarrollo de aplicaciones con características avanzadas.

-Herramientas de Desarrollo:

*Expo* incluye una serie de herramientas diseñadas para facilitar la depuración, prueba y despliegue de aplicaciones. Entre estas herramientas se encuentran *Expo CLI* y *Expo DevTools*, que proporcionan interfaces para la gestión de la aplicación. Estas herramientas permiten a los desarrolladores realizar pruebas y depuraciones de manera más eficiente, mejorando la calidad del código y acelerando el proceso de desarrollo.

-Visualización en Tiempo Real:

Una de las características más apreciadas de *Expo* es la capacidad de visualizar las modificaciones en tiempo real en dispositivos móviles. Esto se consigue mediante la recarga en vivo, que permite a los desarrolladores ver inmediatamente los cambios que realizan en el código reflejados en la aplicación en ejecución. Esta funcionalidad es muy importante para un desarrollo iterativo y rápido, ya que facilita la experimentación y el ajuste del diseño y la funcionalidad de la aplicación.

-Gestión de Certificados de Seguridad:

*Expo* también simplifica la gestión de certificados de seguridad necesarios para el despliegue de aplicaciones en las tiendas de aplicaciones *iOS* y *Android*. Este proceso, que puede ser complicado y propenso a errores cuando se realiza manualmente, es manejado por *Expo* de manera automática. Esto asegura que los certificados estén siempre en orden y que la aplicación cumpla con los requisitos de seguridad necesarios para su distribución, reduciendo el riesgo de errores y facilitando el proceso de publicación.

**Servidor *Backend***

Para el desarrollo del servidor *backend* de la aplicación, se eligió *Node.js*, una plataforma de ejecución de *JavaScript* conocida por su velocidad y eficiencia. *Node.js* permite a los desarrolladores construir aplicaciones escalables y de alto rendimiento gracias a su

16 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

arquitectura basada en eventos y su modelo de operación. A continuación, se detallan algunas de las características claves:



-Asíncrona:

Una de las características más destacadas de *Node.js* es su capacidad para manejar operaciones de manera asíncrona. A diferencia de otros modelos de ejecución que pueden bloquear el hilo principal durante la ejecución de tareas, *Node.js* utiliza un modelo de entrada/salida no bloqueante y un ciclo de eventos que permite manejar múltiples operaciones al mismo tiempo. Esto es muy beneficioso para aplicaciones que requieren manejar muchas conexiones concurrentes, como servidores web y *APIs*, ya que mejora significativamente el rendimiento y la capacidad de respuesta.

-NPM (*Node Package Manager*):



*Node.js* viene acompañado de NPM (*Node Package Manager*), una herramienta que se encarga de la gestión de dependencias. NPM facilita la instalación, actualización y gestión de paquetes de terceros que pueden ser integrados en el proyecto. Con una gran colección de paquetes disponibles, NPM permite a los desarrolladores acelerar el proceso de desarrollo al reutilizar soluciones ya existentes para tareas comunes, como la autenticación, la gestión de bases de datos, el manejo de archivos y mucho más. Esto no solo reduce el tiempo de

17 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

desarrollo, sino que también permite a los desarrolladores enfocarse en las funcionalidades específicas de la aplicación.



-Creación de *APIs*:

*Node.js* es ideal para la creación de servicios web y *APIs*, que son esenciales para la comunicación entre la aplicación móvil y la base de datos. Utilizando *frameworks* populares como *Express.js*, los desarrolladores pueden definir rutas y controladores que gestionan las solicitudes *HTTP*, permitiendo que la aplicación móvil realice operaciones como la recuperación, actualización y eliminación de datos. La flexibilidad y simplicidad de *Node***.js** y sus *frameworks* asociados permiten construir *APIs* robustas y bien estructuradas, facilitando la integración con otros servicios y aplicaciones.

-Ecosistema y Comunidad Activa:

Además de sus características técnicas, *Node.js* cuenta con una comunidad muy amplia que continuamente aporta nuevas herramientas, librerías y mejoras. La comunidad de *Node.js* es conocida por su disposición para compartir conocimientos y soluciones, lo que facilita la resolución de problemas y la implementación de nuevas funcionalidades. Esta comunidad asegura que *Node.js* se mantenga actualizado con las últimas tendencias y avances en el desarrollo de software.

-Escalabilidad y Rendimiento: La arquitectura de *Node.js* también facilita la escalabilidad horizontal, permitiendo que las aplicaciones se expandan fácilmente para manejar un aumento en la carga de trabajo. Gracias a su naturaleza no bloqueante y la capacidad de manejar múltiples conexiones al mismo tiempo, *Node.js* es capaz de mantener un rendimiento alto incluso en condiciones de alta demanda. Esto lo convierte en una excelente opción para aplicaciones que necesitan escalar rápidamente en respuesta a un crecimiento en la base de usuarios.

18 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

En resumen, la elección de *Node.js* para el desarrollo del servidor *backend* de la aplicación proporciona una plataforma robusta y eficiente que soporta operaciones asíncronas, una fácil gestión de dependencias a través de NPM, y la creación de *APIs* sólidas para la comunicación entre la aplicación móvil y la base de datos. Con el respaldo de una comunidad activa y un ecosistema en constante evolución, *Node.js* ofrece las herramientas necesarias para construir servidores *backend* de alto rendimiento y escalabilidad.

**Base de Datos**

*****Firebase Firestore* se utilizó como base de datos en tiempo real para gestionar y almacenar de manera eficiente los datos de las incidencias reportadas por los usuarios, así como la información correspondiente de cada comunidad autónoma y cada provincia. Este sistema de base de datos en la nube ofrece una serie de características avanzadas que lo hacen ideal para aplicaciones que requieren actualizaciones rápidas y confiables. Entre las características principales de *Firestore* se destacan:

- Sincronización en Tiempo Real: *Firestore* permite la sincronización de datos en tiempo real entre la aplicación y la base de datos, lo que garantiza que todos los usuarios vean la información más actualizada sin necesidad de refrescar manualmente. Esto es muy útil en aplicaciones con muchas interacciones donde los datos cambian constantemente.

- Actualización Instantánea: Una de las ventajas más destacables de *Firestore* es su capacidad para proporcionar actualizaciones instantáneas de los datos en la aplicación. Cada vez que se realiza un cambio en los datos, esos cambios se reflejan de inmediato en todas las instancias

19 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

de la aplicación que están conectadas, asegurando así que los usuarios siempre trabajen con la información más reciente.

- Integración Sencilla: *Firestore* se integra de manera muy sencilla en las aplicaciones móviles, facilitando el proceso de desarrollo y reduciendo el tiempo necesario para poner en funcionamiento una base de datos grande y escalable. Su compatibilidad con diversas plataformas y su fácil configuración hacen que los desarrolladores se puedan enfocar más en la funcionalidad de la aplicación en lugar de en la gestión de la infraestructura de la base de datos.

Además, *Firestore* ofrece otras ventajas como la escalabilidad automática, seguridad mediante reglas de acceso detalladas, y la posibilidad de realizar consultas complejas con gran eficiencia. Esto lo convierte en una solución muy robusta y versátil para una amplia gama de aplicaciones, desde pequeñas *startups* hasta grandes corporaciones que requieren un manejo de datos en tiempo real y con alta disponibilidad.

20 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias



**Autenticación de los Usuarios**

Para garantizar la autenticación segura de los usuarios en la aplicación, se utilizó *Firebase Authentication*. Este servicio proporciona una serie de funciones que aseguran que solo usuarios autorizados puedan acceder a la aplicación, aportando una experiencia de usuario segura y confiable. Las principales funciones incluidas son:



- Inicio de Sesión Seguro: *Firebase Authentication* permite la autorización de inicio de sesión mediante direcciones de correo electrónico y contraseñas, asegurando que solo los usuarios registrados puedan acceder a la aplicación y pueden acceder a diferentes funciones dentro de

21 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

ésta. Este método proporciona una capa adicional de seguridad al verificar la identidad del usuario antes de otorgar acceso.



- Autenticación Social: Además del inicio de sesión tradicional, *Firebase Authentication* soporta la autenticación a través de proveedores externos como Google y Facebook. Esto facilita a los usuarios registrarse e iniciar sesión utilizando sus cuentas de redes sociales, mejorando la experiencia del usuario y aumentando la tasa de registro al reducir la fricción en el proceso de autenticación.



- Restablecimiento de Contraseñas: En caso de que un usuario olvide su contraseña, *Firebase* proporciona una solución sencilla para restablecerla. Los usuarios pueden solicitar el restablecimiento de su contraseña directamente desde la consola de autenticación de *Firebase*. Al hacerlo, recibirán un correo electrónico con un enlace para crear una nueva contraseña, lo que facilita la recuperación del acceso a la cuenta sin comprometer la seguridad.

22 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias



**Almacenamiento de Archivos**

Para almacenar las imágenes asociadas a los informes de incidentes, así como las banderas de las comunidades y las imágenes de fondo de cada provincia, se utilizó *Firebase Storage*. Esta plataforma proporciona un almacenamiento en la nube que es tanto escalable como seguro, ideal para archivos multimedia. La capacidad de que la base de datos sea escalable permite manejar grandes volúmenes de datos sin comprometer el rendimiento, mientras que las medidas de seguridad integradas garantizan que los archivos estén bien protegidos contra accesos no autorizados. De este modo, *Firebase Storage* se presenta como una solución eficiente para la gestión y el resguardo de estos tipos de archivos.

23 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias



24 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias



**Herramientas de Navegación**

*React Navigation* es una biblioteca que ofrece una amplia gama de componentes y funciones diseñadas para facilitar la navegación dentro de aplicaciones *React Native*. Sus usos principales incluyen:

- Navegación entre Pantallas: Gestiona de manera eficiente la transición entre las diferentes pantallas de la aplicación, proporcionando una experiencia fluida y coherente para el usuario.

- Navegación Anidada: Permite la creación de flujos de navegación complejos, lo que facilita el manejo de estructuras de navegación dentro de la aplicación que pueden incluir múltiples niveles de pantallas y *subpantallas*.

- Gestos y Animaciones: Ofrece soporte para gestos y animaciones, mejorando la experiencia del usuario al permitir interacciones más naturales y dinámicas con la aplicación.

25 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

En el caso de la aplicación desarrollada la navegación se realiza mediante la pulsación, ya sean botones o cualquier otro tipo de objeto; y la información se va pasando de una pantalla a otra, esto hace que sea mucho más entendible por el usuario.

En este ejemplo, al pulsar en el *card* de cualquier comunidad, abriría la pantalla ‘Provincias’ y le pasaría el nombre de la comunidad pulsada a la segunda pantalla.

Gracias a estas capacidades, *React Navigation* se considera como una herramienta esencial para desarrolladores que buscan implementar una navegación avanzada en sus aplicaciones *React Native.*

**

**Animaciones**

Una biblioteca que facilita la creación de animaciones en componentes de *React Native* es *React Native Animatable*. Esta herramienta es muy útil para los desarrolladores que desean añadir animaciones fluidas y atractivas a sus aplicaciones móviles sin tener que lidiar con un código muy complejo que generalmente implica la creación de animaciones personalizadas desde cero. Las ventajas que tiene *React Native Animatable* son:

26 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

- Animaciones Predefinidas: *React Native Animatable* proporciona una serie de animaciones listas para usar. Estas animaciones predefinidas permiten implementar efectos visuales comunes de manera rápida, sin tener que escribir manualmente los detalles de la animación. Esto incluye transiciones, efectos de entrada y salida, y otros movimientos dinámicos que mejoran la experiencia del usuario.

- Facilidad de Uso: Una de las características más atractivas es su facilidad de uso. La biblioteca simplifica la implementación de animaciones complejas mediante una sintaxis sencilla e intuitiva. Incluso aun con poca experiencia en animaciones se pueden aprovechar sus funcionalidades para crear efectos visuales impresionantes. La simplicidad de su *API* permite definir animaciones en muy poco tiempo, reduciendo significativamente el tiempo de desarrollo y la curva de aprendizaje.

Además de estas ventajas, *React Native Animatable* se integra perfectamente con otros componentes y librerías de *React Native*, proporcionando una experiencia de desarrollo coherente y sin fricciones. La capacidad de aplicar animaciones a cualquier componente con facilidad, junto con la amplia documentación y comunidad de soporte, hace de *React Native Animatable* una herramienta imprescindible para cualquiera que busque mejorar la interactividad y el atractivo visual de sus aplicaciones móviles.

En este desarrollo, se han configurado botones, *pop-ups* e incluso tarjetas como la de las provincias para realizar distintas animaciones durante su funcionamiento. A veces, estas animaciones son imperceptibles, pero su presencia contribuye a un efecto que mejora enormemente el acabado final de la app.

27 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

Este código crea una vista animada que puede escalarse y rotarse, y dentro de esta vista, se muestra un icono basado en el estado de *currentUser*. La animación y el cambio de iconos proporcionan una interfaz de usuario más dinámica y visualmente atractiva.

**Herramientas de Desarrollo**

Para mejorar la eficiencia y la calidad del código durante el desarrollo, se utilizaron diversas herramientas especializadas. Estas herramientas no solo facilitaron el proceso de desarrollo, sino que también aseguraron que el código sea de alta calidad y fácil de mantener. Entre las principales herramientas utilizadas se encuentran:

- *Visual Studio Code*: Este fue el entorno de desarrollo integrado (*IDE*) principal utilizado en el proyecto. *Visual Studio Code* es muy popular debido a su flexibilidad, facilidad de uso y la gran cantidad de extensiones disponibles. Permite una edición de código eficiente con características como el resaltado de sintaxis, la integración de terminales y la depuración en tiempo real. Además, su capacidad para integrarse con otros servicios y herramientas hace que sea ideal para proyectos de desarrollo complejos.



- *Git* y *GitHub*: Para el control de versiones y la colaboración en el código, se utilizaron *Git* y *GitHub*. *Git* es un sistema de control de versiones distribuido que permite a los desarrolladores rastrear los cambios en el código a lo largo del tiempo. Esto es esencial para poder revertir a versiones anteriores en caso de errores o problemas. *GitHub*, por su parte, es una plataforma que facilita la colaboración entre desarrolladores; aunque esta aplicación se haya desarrollado por una sola persona, permite ver los cambios que has realizado en el código y comparar los cambios con el código actual. A través de *GitHub*, los miembros del equipo pueden trabajar en diferentes ramas del proyecto, revisar y fusionar cambios, y gestionar fallos y *pulls*

28 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

solicitados. Esta herramienta es fundamental para la colaboración en equipo y el mantenimiento de un historial claro y organizado del desarrollo del proyecto.



**Capítulo 5. Desarrollo e Implementación**

Este capítulo detalla el proceso de desarrollo e implementación de la aplicación. Se destacan la metodología, las estrategias de desarrollo y las aplicaciones clave utilizadas en la implementación.

**5.1. Metodología del desarrollo**

Para desarrollar esta aplicación, se eligió un método que fuese ágil y que se pudiese adaptar a un desarrollador individual. La flexibilidad de esta metodología permitió una mejor adaptación a los cambios. El desarrollo se organizó en *sprints* de una semana, con una planificación y revisión al inicio y al final de cada sprint.

**5.2. Etapas de desarrollo**

El desarrollo de la aplicación se dividió en varias etapas clave, cada una de esas etapas estaban enfocadas en un conjunto de diferentes funcionalidades y tareas que se tenían que cumplir.

**Planificación y Diseño**

-Definición de Requisitos: Se identificaron las funcionalidades esenciales de la aplicación a través de un análisis y una definición de los objetivos personales.

-Diseño de la Interfaz de Usuario: Se crearon unos prototipos de las pantallas utilizando diferentes herramientas, la más destacada fue el uso de *Figma*.

29 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias



-Arquitectura del Sistema: Diseño de la arquitectura de la aplicación, incluyendo la estructura de la base de datos en *Firebase* y la organización del código.

**Configuración del Entorno de Desarrollo**

-Configuración de *Expo* y *React Native*: Se instaló y configuró el entorno utilizando *Expo* para tener y facilitar la visualización en tiempo real de los cambios que se iban realizando. -Configuración *Backend* con *Node.js*: Se instaló *Node.js* y se configuró el servidor *backend* para la gestión de *APIs* y la comunicación con *Firebase* y el código.

**Desarrollo de Funcionalidades**

Como la aplicación se llevó a cabo en diferentes *Sprints*, cada uno está enfocado a diferentes módulos y características:

30 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias



-**Sprint 1: Gestión de Incidencias**

>Pantallas de Listas: Se desarrollaron las listas de comunidades autónomas y provincias. Inicialmente, se imprimieron todas en una sola pantalla y luego se creó una segunda pantalla donde solo aparecían las provincias. Al final del sprint, se logró filtrar las provincias por comunidad en la segunda pantalla, implementando la navegación entre ambas.

>Tercera y Cuarta Pantalla: Se implementó una pantalla para mostrar la lista de incidencias. También se desarrolló una pantalla para la creación de incidencias directamente desde la aplicación, permitiendo a los usuarios añadir nuevas incidencias a la base de datos.

**-SPRINT 2(LOGIN, CHAT & CAMARA)**

>Gestión de Usuarios: Se creó una nueva pantalla para la gestión de usuarios, donde se podía iniciar sesión y registrarse utilizando *Firebase Auth*. Se añadieron funcionalidades como el cierre de sesión y la pre visualización de la contraseña.

>Chat de Usuarios: Se implementó un chat donde los usuarios registrados podían enviar mensajes, los cuales se almacenan en una colección de la base de datos *Firestore*.

31 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

>Funcionalidad de Cámara: Se habilitó la funcionalidad de la cámara para tomar fotos y se integró la capacidad de acceder a la cámara del dispositivo desde la aplicación. 

**-SPRINT 3(FOTO, GALLERY & CAMARA)**

>Subida y Visualización de Imágenes: Se añadió la funcionalidad para permitir a los usuarios subir imágenes de incidencias utilizando *Expo Camera* y *Firebase*.



32 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

>Imágenes de Comunidades Autónomas y Provincias: Se añadieron imágenes de banderas y paisajes en la base de datos para hacer la aplicación más atractiva visualmente. >Subida de Imágenes desde la Cámara: Se implementó la funcionalidad de subir imágenes tomadas con la cámara directamente a la base de datos. Se añadieron botones para acceder a la galería y a la cámara del dispositivo, permitiendo la subida de fotos al almacenamiento de *Firebase*.



**-SPRINT 4(PANTALLA CARGA, UBICACIÓN, MANEJO ERRORES)**

>Pantalla de Carga: Se implementó una pantalla de carga con animaciones para mejorar la experiencia del usuario.



33 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

>Funcionalidades de Ubicación: Se integraron servicios de ubicación para añadir características basadas en la localización del usuario.



>Manejo de Errores: Se desarrollaron mecanismos para manejar errores de manera efectiva, asegurando una experiencia de usuario más robusta.



**Pruebas y Depuración**

-Pruebas de Integración: Se llevaron a cabo pruebas de integración para observar el correcto funcionamiento entre los distintos componentes de la aplicación.

34 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

-Depuración: Se utilizaron diversas herramientas de depuración y registro para detectar y solucionar los errores en el código.

**5.3. Funcionalidades Implementadas**

La aplicación final incluye una serie de funcionalidades que permiten a los usuarios gestionar las incidencias de una manera eficiente:

-Autenticación de Usuarios: Registro e inicio y cierre de sesión mediante correo y contraseña. -Gestión de Incidencias (CRUD): Creación, visualización, actualización y eliminación de incidencias.

-Integración de Multimedia: Subida y visualización de imágenes asociadas a las incidencias. -Geolocalización: A través de un mapa, es posible identificar el lugar preciso donde se creó la incidencia utilizando la latitud y longitud.

-Mensajería: Disposición de un Chat general en que cualquier usuario registrado puede participar y mandar mensajes.

-Navegación Intuitiva: Navegación fluida entre las diferentes pantallas de la aplicación -Interfaz de Usuario Atractiva: Uso de animaciones y estilos para mejorar la experiencia del usuario.

**5.4. Desafíos y Soluciones**

Durante el desarrollo de la aplicación hubo desafíos a los cuales se abordaron con soluciones específicas como:

-Sincronización en Tiempo Real: La integración de *Firebase Firestore* como base de datos permitió resolver el problema de mantener la sincronización en tiempo real de los datos entre los usuarios.

-Autenticación Segura: La implementación de *Firebase Authentication* garantizó que la autenticación fuese segura y confiable para los usuarios.

-Gestión de Imágenes: *Firebase Storage* proporcionó una solución eficiente y escalable al problema del almacenamiento de las imágenes.

El desarrollo y la implementación de esta aplicación demuestra la efectividad de usar tecnologías modernas como son el caso de *React Native, Expo, Firebase*, que permiten la

35 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

creación de una aplicación robusta y de un rendimiento alto en un plazo de tiempo que es relativamente corto.

**Capítulo 6. Conclusiones y líneas futuras**

Crear esta aplicación móvil para gestionar incidencias ha sido una experiencia enriquecedora y desafiante. A lo largo del proyecto, se lograron una serie de hitos importantes y se adquieren conocimientos valiosos sobre el desarrollo de aplicaciones móviles utilizando tecnologías modernas como *React Native, Expo, Node.js y Firebase*. A continuación, se muestra el final más apropiado para el proyecto:

**6.1. Conclusiones**

**Eficiencia en el Desarrollo Multiplataforma**

Utilizando *React Native*, fue posible desarrollar una aplicación para ambas plataformas principales, que son *IOS* y *Android*, con un único código base. Esto no solo ahorró mucho tiempo, sino que también garantizo una experiencia consistente para los usuarios en ambas plataformas.

**Facilidad de Integración y uso de *Expo***

Expo proporcionó un entorno de desarrollo flexible que hizo que muchas de las tareas comunes fueses más sencillas, como la inicialización de proyectos, la gestión de las dependencias, y la implementación de funciones nativas. La capacidad de monitorear los cambios en tiempo real en dispositivos simplificó enormemente el proceso de desarrollo.

**Eficiencia y Escalabilidad con *Firebase***

*Firebase* demostró ser una solución muy sólida para la gestión de los datos en tiempo real, la autenticación de usuarios y el almacenamiento de archivos. La integración de *Firebase* permitió crear una aplicación funcional y segura, capaz de manejar una gran cantidad de usuarios y datos de una manera muy eficiente.

36 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

**Desarrollo Ágil y Adaptable**

La metodología ágil adaptada al proyecto individual permitió una planificación flexible y cambios rápidos en los plazos de entrega. El cronograma en sprints facilitó la entrega gradual de las tareas, ahorrando tiempo y recursos.

**Desafíos Superados**

Durante el desarrollo, surgieron varios desafíos técnicos, como el procesamiento de datos en tiempo real y la implementación de una autenticación segura. Estos desafíos se abordaron con éxito mediante el uso de *Firebase* y otras herramientas de desarrollo, demostrando la capacidad de resolver problemas complejos y mejorar continuamente la aplicación.

**6.2. Líneas futuras**

Aunque la aplicación consigue los objetivos iniciales, existen muchas áreas donde se pueden mejorar y ampliar en el futuro para aumentar su funcionalidad y usabilidad. A continuación, se comentan algunos posibles desarrollos futuros de la aplicación:

**Notificaciones de *Push***

Implementar notificaciones *push* para alertar a los usuarios sobre las nuevas incidencias, actualizaciones y otros eventos importantes. Esto mejoraría la interacción y el compromiso entre los usuarios y la aplicación.

**Mejoras en la Interfaz de Usuario**

Continuar mejorando la interfaz de usuario y la experiencia de usuario (*UI/UX*) mediante la incorporación de nuevas animaciones, diseños que sean más intuitivos y una navegación más fluida. Hacer pruebas piloto para obtener un *feedback* y realizar cambios en base a sus sugerencias.

**Nuevas Funcionalidades de Geolocalización**

Integrar capacidades de geolocalización para permitir a los usuarios ver las incidencias en un mapa. Esto proporcionará una visión más clara y contextual de las incidencias.

37 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

Integrar un mapa para poder ver las diferentes incidencias en una región concreta. Esto proporcionara una visión más clara para la administración.

**Soporte Multilenguaje**

Añadir soporte para múltiples idiomas para que la aplicación sea accesible a una audiencia más amplia. Esto requerirá la traducción de la interfaz de usuario y cualquier contenido a varios idiomas.

**Estadísticas y Reportes**

Implementar módulos y estadísticas y generación de informes para que los usuarios y administradores puedan ver eventos, tendencias y patrones detallados. Esto ayudará a mejorar constantemente la toma de decisiones y los servicios.

**Integración con Redes Sociales**

Permitir a los usuarios compartir noticias y actualizaciones directamente en sus redes sociales. Esto no solo aumentará la visibilidad de la aplicación, sino que también puede ayudar a obtener más información y apoyo de la comunidad.

**Automatización y *Bots***

Incorporar *bots* y automatización para responder preguntas, guiar a los usuarios a través de canales y realizar tareas comunes. Esto mejoraría la eficiencia y la satisfacción del usuario.

**Roles**

Dependiendo del usuario que utilice la aplicación, este puede ser un ciudadano o un administrador, con privilegios y recursos diferenciados según su rol.

**Seguridad y Privacidad Mejoradas**

Continuar aportando las medidas de seguridad y privacidad para proteger los datos de los usuarios y cumplir con las leyes y mejoras prácticas de protección de datos. A su vez se podría implementar una función para la recuperación de la contraseña de un usuario en el caso de la que haya olvidado, pudiendo recibir un correo electrónico para reestablecerla, y no tener que

38 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

acceder a la base de datos para poder hacerlo, sino que desde la propia aplicación se pueda hacer.

En conclusión, este proyecto ha creado una base sólida para una aplicación móvil funcional y de alto rendimiento. Las tecnologías implementadas han demostrado ser efectivas y el proceso de desarrollo ha resultado en mejoras consistentes y sistemáticas. Con futuras mejoras y ampliaciones, esta aplicación tiene el potencial de convertirse en una herramienta indispensable para la gestión de incidentes, proporcionando un valor significativo a los usuarios.

39 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

**Capítulo 7. Bibliografía.**

Para la realización de este proyecto se requirieron diversas fuentes de información y referencias que fueron fundamentales para el desarrollo de la aplicación. A continuación, se presenta la bibliografía utilizada, organizada por categorías:

**Documentación Técnica**

React Native. (n.d.). *React Native Documentation*. Recuperado de

https://reactnative.dev/docs/getting-started

Expo. (n.d.). *Expo Documentation*. Recuperado de https://docs.expo.dev/ Node.js. (n.d.). *Node.js Documentation*. Recuperado de https://nodejs.org/en/docs/ Firebase. (n.d.). *Firebase Documentation*. Recuperado de https://firebase.google.com/docs React Navigation. (n.d.). *React Navigation Documentation*. Recuperado de https://reactnavigation.org/docs/getting-started

React Native Animatable. (n.d.). *React Native Animatable Documentation*. Recuperado de https://github.com/oblador/react-native-animatable

**Sitios Web y Blogs**

Varios autores. (n.d.). *Medium Articles on React Native*. Recuperado de https://medium.com/tag/react-native

Stack Overflow. (n.d.). *React Native and Expo Solutions*. Recuperado de https://stackoverflow.com/questions/tagged/react-native

Varios autores. (n.d.). *CSS-Tricks on Animations and Styling in React Native*. Recuperado de https://css-tricks.com/

40 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

**Cursos y Tutoriales**

Udemy – *The Complete React Native and Redux Course*. Udemy. Recuperado de https://www.udemy.com/course/the-complete-react-native-and-redux-course/ Academind. (n.d*.). React Native Tutorials*. Youtube. Recuperado de

https://www.youtube.com/c/Academind

**Herramientas y Documentación de Desarrollo**

Microsoft. (n.d.). *Visual Studio Code Documentation*. Recuperado de

https://code.visualstudio.com/docs

GitHub. (n.d.). *Github Documentation*. Recuperado de https://docs.github.com/en Npm. (n.d.). *npm Documentation*. Recuperado de https://docs.npmjs.com/

Artículos y Publicaciones Académicas

IEEE Xplore. (n.d.). *Articles on Real-Time Databases and User Authentication*. Recuperado de https://ieeexplore.ieee.org/

Esta bibliografía representa una recopilación de fuentes y de datos que han sido revisados y utilizados en el desarrollo del proyecto. Cada uno de estos artículos proporciona información esencial y herramientas útiles que han facilitado la creación de aplicaciones móviles eficaces y responsivas para la gestión de eventos.

41 Alberto Escobar Rodríguez S2DAM

Gestión de Incidencias

42 Alberto Escobar Rodríguez S2DAM