

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE SOFTWARE

Tema:

DESARROLLO DE UN SISTEMA WEB Y MÓVIL QUE MEJORE LA OBTENCIÓN DE DATOS GEORREFERENCIADOS DE DELITOS CON EL PROPÓSITO DE CREAR UN MODELO ANALÍTICO DE BUSINESS INTELLIGENCE (BI) ORIENTADO AL SEGUIMIENTO DE LAS TENDENCIAS DE SINIESTROS EN LA CIUDAD DE AMBATO

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Software

ÁREA: Software

LÍNEA DE INVESTIGACION: Desarrollo de software

AUTOR: Jonathan José Pazmiño Fiallos

TUTOR: Ing. Hernán Fabricio Naranjo Avalos, Mg.

Ambato - Ecuador

agosto - 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: DESARROLLO DE UN SISTEMA WEB Y MÓVIL QUE MEJORE LA OBTENCIÓN DE DATOS GEORREFERENCIADOS DE DELITOS CON EL PROPÓSITO DE CREAR UN MODELO ANALÍTICO DE BUSINESS INTELLIGENCE (BI) ORIENTADO AL SEGUIMIENTO DE LAS TENDENCIAS DE SINIESTROS EN LA CIUDAD DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación, por el señor Jonathan José Pazmiño Fiallos, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2024.

Firmado electrónicamente por:
HERNAN FABRICIO NARANJO AVALOS

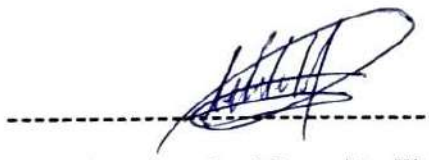
Ing. Hernán Fabricio Naranjo Avalos, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: DESARROLLO DE UN SISTEMA WEB Y MÓVIL QUE MEJORE LA OBTENCIÓN DE DATOS GEORREFERENCIADOS DE DELITOS CON EL PROPÓSITO DE CREAR UN MODELO ANALÍTICO DE BUSINESS INTELLIGENCE (BI) ORIENTADO AL SEGUIMIENTO DE LAS TENDENCIAS DE SINIESTROS EN LA CIUDAD DE AMBATO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2024.



Jonathan José Pazmiño Fiallos

C.C. 1805283452

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2024.



Jonathan José Pazmiño Fiallos

C.C. 1805283452

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Jonathan José Pazmiño Fiallos, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado DESARROLLO DE UN SISTEMA WEB Y MÓVIL QUE MEJORE LA OBTENCIÓN DE DATOS GEORREFERENCIADOS DE DELITOS CON EL PROPÓSITO DE CREAR UN MODELO ANALÍTICO DE BUSINESS INTELLIGENCE (BI) ORIENTADO AL SEGUIMIENTO DE LAS TENDENCIAS DE SINIESTROS EN LA CIUDAD DE AMBATO nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2024.

Firmado electrónicamente por:
ELSA PILAR URRUTIA URRUTIA

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
EDISON HOMERO ALVAREZ MAYORGA

Ing. , Mg. Edison Homero Alvarez Mayorga
PROFESOR CALIFICADOR

Firmado electrónicamente por:
JULIO ENRIQUE BALAREZO LÓPEZ

PhD. Julio Enrique Balarezo López
PROFESOR CALIFICADOR

DEDICATORIA

A mis queridos padres, Luzmila y José, por ser mi mayor inspiración y apoyo incondicional. Su amor y sacrificio han sido el pilar sobre el que he construido cada uno de mis logros. A mi hermana Evelyn, por estar siempre a mi lado, compartiendo alegrías y dificultades. Tu compañía y comprensión han sido esenciales en este camino. A mi sobrina Karla, cuya sonrisa y energía me recuerdan cada día la importancia de luchar por un futuro mejor. Y a mis amigos, por su apoyo incondicional en cada paso de mi etapa universitaria.

Jonathan José Pazmiño Fiallos

AGRADECIMIENTOS

Quiero expresar mi más profundo agradecimiento a todas las personas que han sido parte de este importante logro en mi vida. A mi familia, por su inquebrantable apoyo, amor y sacrificio que me han permitido llegar hasta este punto; a mis amigos, por la compañía, las risas y los momentos inolvidables que hemos compartido; a mis profesores y mentores, por su guía, conocimiento y paciencia; y a mi tutor, por su valiosa orientación y apoyo en la realización de este trabajo.

Jonathan José Pazmiño Fiallos

ÍNDICE GENERAL DE CONTENIDOS

PORTADA i

APROBACIÓN DEL TUTOR i

AUTORÍA ii

DERECHOS DE AUTOR iii

APROBACIÓN DEL TRIBUNAL DE GRADO iv

DEDICATORIA v

AGRADECIMIENTO vi

ÍNDICE GENERAL DE CONTENIDOS ix

ÍNDICE DE TABLAS xii

ÍNDICE DE FIGURAS xvi

ÍNDICE DE ANEXOS xviii

RESUMEN EJECUTIVO xix

ABSTRACT xx

CAPÍTULO I. MARCO TEÓRICO 1

1.1 Tema de investigación 1

1.1.1 Planteamiento del problema 1

1.2 Antecedentes investigativos 3

1.3 Fundamentación teórica 9

1.3.1 Metodología ágil 9

1.3.2 Framework Web 10

1.3.3 Framework Backend 10

1.3.4 Aplicación Móvil 11

1.3.5 Bases de datos 11
1.3.6 Mapa web 12
1.3.7 Geo Business Intelligence (GEOBI) 12
1.3.8 Patrón MVC 13
1.3.9 Diagrama UML 13
1.3.10 NextJS 13
1.3.11 NestJS 14
1.3.12 Flutter 14
1.4 Objetivos 15
1.4.1 Objetivo general 15
1.4.2 Objetivos específicos 15
CAPÍTULO II. METODOLOGÍA **16**
2.1 Materiales 16
2.1.1 Institucionales 16
2.1.2 Humanos 16
2.1.3 Materiales 16
2.1.4 Económicos 17
2.2 Métodos 18
2.2.1 Modalidad de investigación 18
2.2.2 Población y muestra 18
2.2.3 Recolección de información 19
2.2.4 Procesamiento y análisis de datos 28
CAPÍTULO III. RESULTADOS Y DISCUSIÓN **30**
3.1 Análisis de procesos 30
3.2 Análisis de herramientas de desarrollo 32
3.2.1 Análisis y selección de la metodología de desarrollo 32
3.2.2 Análisis y selección de herramientas de desarrollo 35
3.2.3 Análisis y selección del framework de desarrollo para el servidor web (Backend) 35
3.2.4 Análisis y selección del framework de desarrollo para el cliente web (Frontend) 37

3.2.5 Análisis y selección del framework de desarrollo para el cliente móvil (Frontend) 43
3.2.6 Análisis y selección de la base de datos 45
3.3 Análisis de la precisión de los datos georreferenciados 47
3.4 Desarrollo de la propuesta 53
3.4.1 Fase I: Planificación 53
3.4.2 Fase II: Diseño 81
3.4.3 Fase III: Codificación 124
3.4.4 Fase IV: Pruebas 163
3.4.5 Fase V: Lanzamiento 177
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 179
4.1 Conclusiones 179
4.2 Recomendaciones 180
REFERENCIAS BIBLIOGRÁFICAS 185
ANEXOS 186

ÍNDICE DE TABLAS

1	Artículos obtenidos	3
2	Criterios de exclusión e Inclusión	4
3	Presupuesto	17
4	Entrevista aplicada al Ing. Rubén Nogales experto en Business Intelligence	20
5	Entrevista aplicada al teniente coronel de estado mayor Christian Iván Quintana Guerra	25
6	Análisis y comparación entre metodologías de desarrollo ágil	33
7	Análisis y comparación entre los frameworks de NodeJS Express y Fastify	36
8	Análisis y comparación entre los frameworks Angular, ReactJS, VueJS	38
9	Análisis y comparación entre los frameworks React Native, Flutter y Xamarin	43
10	Análisis y comparación entre las bases de datos MySQL y PostgreSQL	45
11	Definición de requerimientos	54
12	Definición de requerimientos	56
13	Modelo de historia de usuario	59
14	Historia de usuario 1: Iniciar sesión sistema web	59
15	Historia de usuario 2: Cerrar sesión sistema web	60
16	Historia de usuario 3: Gestionar usuarios	60
17	Historia de usuario 4: Gestionar tipos de incidentes	61
18	Historia de usuario 5: Gestionar zonas de vigilancia	61
19	Historia de usuario 6: Asignar policías a las zonas de vigilancia	62
20	Historia de usuario 7: Gestionar alertas de incidentes	62
21	Historia de usuario 8: Visualizar mapa de calor	63
22	Historia de usuario 9: Iniciar sesión aplicación móvil	63
23	Historia de usuario 10: Cerrar sesión aplicación móvil	64
24	Historia de usuario 11: Registro de usuario	65
25	Historia de usuario 12: Asignar miembros al grupo familiar	65
26	Historia de usuario 13: Enviar alerta de incidente	66
27	Historia de usuario 14: Visualizar alertas de emergencia de familiares	66
28	Historia de usuario 15: Visualizar alertas de emergencia de ciudadanos	67
29	Historia de usuario 16: Cambiar contraseña	67

30 Historia de usuario 17: Recuperar contraseña 68
31 Historia de usuario 18: Desarrollar modelo analítico de BI 68
32 Estimación de tiempo para las historias de usuario 69
33 Asignación de roles en el proyecto en base a la metodología XP 80
34 Historias de usuario Iteración N° 1 82
35 Tareas Iteración N° 1 82
36 Historias de usuario Iteración N° 2 88
37 Tareas Iteración N° 2 89
38 Historias de usuario Iteración N° 3 95
39 Tareas Iteración N° 3 95
40 Historias de usuario Iteración N° 4 99
41 Tareas Iteración N° 4 100
42 Historias de usuario Iteración N° 5 109
43 Tareas Iteración N° 5 110
44 Tarjeta CRC - Historia 1: Iniciar sesión sistema web 114
45 Tarjeta CRC - Historia 2: Cerrar sesión sistema web 115
46 Tarjeta CRC - Historia 3: Gestionar usuarios 115
47 Tarjeta CRC - Historia 4: Gestionar tipos de incidentes 116
48 Tarjeta CRC - Historia 5: Gestionar zonas de vigilancia 116
49 Tarjeta CRC - Historia 6: Asignar policías a las zonas de vigilancia 117
50 Tarjeta CRC - Historia 7: Gestionar alertas de incidentes 117
51 Tarjeta CRC - Historia 8: Visualizar mapa de calor 118
52 Tarjeta CRC - Historia 9: Iniciar sesión aplicación móvil 118
53 Tarjeta CRC - Historia 10: Cerrar sesión aplicación móvil 118
54 Tarjeta CRC - Historia 11: Registro de usuario 119
55 Tarjeta CRC - Historia 12: Asignar miembros al grupo familiar 119
56 Tarjeta CRC - Historia 13: Enviar alertas de emergencia 120
57 Tarjeta CRC - Historia 14: Visualizar alertas de emergencia de familiares 120
58 Tarjeta CRC - Historia 15: Visualizar alertas de emergencia de los ciudadanos 121
59 Tarjeta CRC - Historia 16: Cambiar contraseña 121
60 Tarjeta CRC - Historia 17: Recuperar contraseña 122
61 Tarjeta CRC - Historia 18: Desarrollar el modelo analítico de BI 122

62 Indicadores y perspectivas en base las pregustas de negocio 151
63 Dimensión de tiempo con sus atributos 152
64 Dimensión de usuarios con sus atributos 152
65 Dimensión de tipo de alarma con sus atributos 153
66 Dimensión de tipo de incidente con sus atributos 153
67 Dimensión de zonas de vigilancia con sus atributos 153
68 Dimensión de ubicación con sus atributos 153
69 Hechos de incidentes delictivos con sus atributos 154
70 Modelo de prueba de aceptación 163
71 Prueba de aceptación 1: Iniciar sesión sistema web 164
72 Prueba de aceptación 2: Cerrar sesión sistema web 165
73 Prueba de aceptación 3: Gestionar usuarios 165
74 Prueba de aceptación 4: Gestionar tipos de incidentes 166
75 Prueba de aceptación 5: Gestionar zonas de vigilancia 167
76 Prueba de aceptación 6: Asignar policías a las zonas de vigilancia 167
77 Prueba de aceptación 7: Gestionar alertas de incidentes 168
78 Prueba de aceptación 8: Visualizar mapa de calor 169
79 Prueba de aceptación 9: Iniciar sesión aplicación móvil 169
80 Prueba de aceptación 10: Cerrar sesión aplicación móvil 170
81 Prueba de aceptación 11: Registro de usuario 171
82 Prueba de aceptación 12: Asignar miembros al grupo familiar 171
83 Prueba de aceptación 13: Enviar alerta de incidente 172
84 Prueba de aceptación 14: Visualizar alertas de emergencia de familiares . 173
85 Prueba de aceptación 15: Visualizar alertas de emergencia de ciudadanos 174
86 Prueba de aceptación 16: Cambiar contraseña 174
87 Prueba de aceptación 17: Recuperar contraseña 175
88 Prueba de aceptación 18: Desarrollar modelo analítico de BI en Power BI 176

ÍNDICE DE FIGURAS

1 Formulario de recolección de datos 28
2 Proceso propuesto de reporte de incidentes al ECU 911. 31
3 Proceso de Business Intelligence (BI) para el análisis de incidentes reportados al ECU 911. 32
4 Prototipo de la aplicación móvil para la evaluación de la precisión de los datos georreferenciados. 47
5 Tabla de datos georreferenciados almacenados en la base de datos PostgreSQL. 49
6 Archivo de Excel con los datos georreferenciados para la comparación. . 50
7 Método del codo para determinar el número óptimo de clusters. 50
8 Cluster de puntos georreferenciados obtenidos mediante el algoritmo de K-Means. 51
9 Prototipo de la interfaz de inicio de sesión web 85
10 Prototipo de la interfaz para cerrar sesión en el sistema web 86
11 Prototipo de la interfaz para visualizar los usuarios 86
12 Prototipo de la interfaz para crear/editar usuario 87
13 Prototipo de la interfaz para visualizar los tipos de incidentes 87
14 Prototipo de la interfaz para crear/editar un tipo de incidente 88
15 Prototipo de la interfaz para visualizar las zonas de vigilancia 93
16 Prototipo de la interfaz para crear/editar una zona de vigilancia 93
17 Prototipo de la interfaz para asignar/desasignar policías a la zona de vigilancia 94
18 Prototipo de la interfaz para visualizar las alertas de incidentes 94
19 Prototipo de la interfaz para visualizar el mapa de calor 95
20 Prototipo de la interfaz para iniciar sesión en la aplicación móvil 98
21 Prototipo de la interfaz para cerrar sesión en la aplicación móvil 98
22 Prototipo de la interfaz para el registro de usuarios en la aplicación móvil 99
23 Prototipo de la interfaz para visualizar los miembros del grupo familiar . . 105
24 Prototipo de la interfaz para agregar miembros al grupo familiar 105
25 Prototipo de la interfaz para enviar una alerta de incidente 106

26 Prototipo de la interfaz para visualizar la notificación de alerta de incidente 106
27 Prototipo de la interfaz para visualizar las alertas de emergencia de familiares 107
28 Prototipo de la interfaz para visualizar la ubicación de la alerta de emergencia de un familiar 108
29 Prototipo de la interfaz para visualizar las alertas de emergencia de ciudadanos 108
30 Prototipo de la interfaz para visualizar la ubicación de la alerta de emergencia de un ciudadano 109
31 Prototipo de la interfaz para cambiar la contraseña en la aplicación móvil 113
32 Prototipo de la interfaz para recuperar la contraseña en la aplicación móvil 114
33 Esquema de la base de datos 123
34 Documentación de la API con Swagger. 127
35 Inicio de sesión en el sistema web. 129
36 Menú de usuario en el sistema web. 130
37 Tabla de usuarios en el sistema web. 130
38 Menú de opciones de la tabla de usuarios en el sistema web. 131
39 Formulario para crear/editar usuarios en el sistema web (Parte 1). 131
40 Formulario para crear/editar usuarios en el sistema web (Parte 2). 131
41 Filtros de búsqueda en la tabla de usuarios en el sistema web. 132
42 Tabla de tipos de incidentes en el sistema web. 132
43 Menú de opciones de la tabla de tipos de incidentes en el sistema web. . . 133
44 Formulario para crear/editar tipos de incidentes en el sistema web. 133
45 Filtros de búsqueda en la tabla de tipos de incidentes en el sistema web. . 134
46 Mapa de zonas de vigilancia en el sistema web. 134
47 Menú de opciones del mapa de zonas de vigilancia en el sistema web. . . 135
48 Formulario para crear/editar zonas de vigilancia en el sistema web. 135
49 Asignar policías a zonas de vigilancia en el sistema web. 136
50 Mapa de incidentes en el sistema web. 136
51 Detalles de incidente en el sistema web. 137
52 Tabla de incidentes en el sistema web. 137
53 Mapa de calor en el sistema web. 138
54 Inicio de sesión en la aplicación móvil. 139

55 Menú de usuario en la aplicación móvil. 140
56 Registro de usuario en la aplicación móvil (Parte 1). 141
57 Registro de usuario en la aplicación móvil (Parte 2). 141
58 Registro de usuario en la aplicación móvil (Parte 3). 142
59 Gestión de grupo familiar en la aplicación móvil. 143
60 Agregar miembro al grupo familiar en la aplicación móvil. 143
61 Pantalla principal de la aplicación móvil. 144
62 Notificación de alerta recibida en la aplicación móvil. 145
63 Alerta recibida en la aplicación móvil. 145
64 Seguir ubicación en la aplicación móvil. 146
65 Notificación de alerta recibida en la aplicación móvil del policía. 147
66 Alerta recibida en la aplicación móvil del policía. 147
67 Seguir ubicación en la aplicación móvil del policía. 148
68 Cambiar contraseña en la aplicación móvil. 149
69 Recuperar contraseña en la aplicación móvil. 149
70 Correo electrónico de recuperación de contraseña. 150
71 Esquema del modelo dimensional para el sistema de reportería de incidentes delictivos 155
72 Diseño del proceso ETL para el modelo dimensional. 156
73 Limpieza de las tablas del modelo dimensional. 156
74 Extracción de datos para el modelo dimensional. 157
75 Mensaje de éxito al ejecutar el proceso ETL. 157
76 Conexión a la base de datos de SQL Server en el proyecto de Analysis Services. 158
77 Origen de datos en el proyecto de Analysis Services. 158
78 Diseño del cubo OLAP en el proyecto de Analysis Services. 159
79 Campos en el origen de datos del proyecto de Analysis Services. 159
80 Jerarquías de tiempo en la dimensión de tiempo del proyecto de Analysis Services. 160
81 Mensaje de éxito al procesar el cubo OLAP. 160
82 Conexión al cubo OLAP de Analysis Services en Power BI. 161
83 Informe en Power BI con los datos del cubo OLAP. 161

84 Informe en Power BI con los datos filtrados. 162
85 Informe en Power BI con los datos del cubo OLAP. 162
86 Informe en Power BI con los datos filtrados. 163
87 Despliegue de la aplicación web en Railway. 177
88 Despliegue del API en Railway. 178
89 Archivo APK de la aplicación móvil. 178

ÍNDICE DE ANEXOS

A Guía de entrevista aplicada a experto en Business Intelligence 186
B Guía de entrevista aplicada al coronel de la policía nacional 187
C Script en Node.js para la comparación de los datos georreferenciados 189
D Scrip en Node.js para el calculo de la distancia entre dos puntos georreferenciados 190
E Script en Python para la creación de clusters de puntos georreferenciados . 190
F Archivo package.json con las dependencias de la API REST 190
G Archivo de configuración .env de la API REST 191
H Guard de seguridad de la API REST 191
I Parte del modelo de datos de la API REST 192
J Validación de datos de la API REST 193
K Envío de correos electrónicos de la API REST 194
L Configuración de Cloudinary en la API REST 194
M Subida de imágenes a Cloudinary desde la API REST 195
N Subida de imágenes de usuario a Cloudinary desde la API REST 195
O Plantilla de correo electrónico de la API REST 195
P Configuración de OneSignal en la API REST 195
Q Envío de notificaciones a través de OneSignal desde la API REST 196
R Ejemplo de módulo de la API REST 196
S Ejemplo de controlador de la API REST 197
T Ejemplo de servicio de la API REST 197
U Ejemplo de DTO de la API REST 198
V Dependencias de la aplicación web 198
W Archivo de configuración .env.local de la aplicación web 198
X Configuración de la aplicación web 199
Y Rutas de la aplicación web 199
Z Validación de datos de la aplicación web 199
AA Guardar token de usuario en la aplicación web 200
AB Dependecias de la aplicación móvil 200

AC configuracion-env-movil 201
AD Configuración de la aplicación móvil 201
AE Guardar token de usuario en la aplicación móvi 202

RESUMEN EJECUTIVO

La ciudad de Ambato enfrenta desafíos significativos en la recolección y análisis de datos sobre delitos y siniestros, lo cual impacta negativamente la toma de decisiones en materia de seguridad pública. Los métodos tradicionales, basados en registros manuales y sistemas fragmentados, resultan ineficientes y propensos a errores, afectando la precisión y oportunidad en el seguimiento de tendencias delictivas.

El presente proyecto de investigación propone el desarrollo de un sistema que combina una aplicación móvil y una plataforma web, con el objetivo de optimizar la obtención y análisis de datos georreferenciados sobre delitos y crear un modelo analítico de Business Intelligence (BI). El sistema busca mejorar la eficiencia operativa, reducir los errores humanos y proporcionar una herramienta que facilite la administración y el acceso a la información tanto para las autoridades como para la ciudadanía.

El proyecto se desarrolló utilizando la metodología de desarrollo ágil XP (eXtreme Programming), lo que permitió una mayor flexibilidad y adaptabilidad a los cambios durante el proceso de desarrollo. Este enfoque facilitó iteraciones rápidas y la incorporación de retroalimentación continua, asegurando que el producto final cumpliera con los requisitos a medida que se avanzaba en el desarrollo del sistema.

El proyecto se divide en dos aplicaciones principales: web y móvil. Para el desarrollo de la aplicación móvil se utilizó Flutter, debido a su rendimiento multiplataforma. Para la aplicación web se empleó NextJS y NestJS, tecnologías que permiten la creación de aplicaciones web robustas y escalables. Adicionalmente, se empleó una base de datos geoespacial de PostgreSQL para gestionar eficientemente grandes volúmenes de datos georreferenciados y desarrollar un modelo analítico de BI que permita visualizar y analizar tendencias de criminalidad.

Palabras clave: Rappid Application Development, sistema web, aplicación móvil, Business Intelligence, georreferenciación, seguridad pública, Flutter, NextJS, NestJS

ABSTRACT

The city of Ambato faces significant challenges in the collection and analysis of crime and casualty data, which negatively impacts public safety decision-making. Traditional methods, based on manual records and fragmented systems, are inefficient and error-prone, affecting accuracy and timeliness in tracking crime trends.

This research project proposes the development of a system that combines a mobile application and a web platform, with the objective of optimizing the collection and analysis of geo-referenced crime data and creating a Business Intelligence (BI) analytical model. The system seeks to improve operational efficiency, reduce human error and provide a tool that facilitates administration and access to information for both authorities and citizens.

The project was developed using the XP (eXtreme Programming) agile development methodology, which allowed for greater flexibility and adaptability to changes during the development process. This approach facilitated rapid iterations and the incorporation of continuous feedback, ensuring that the final product met requirements as system development progressed.

The project is divided into two main applications: web and mobile. Flutter was used for the development of the mobile application, due to its cross-platform performance. For the web application, NextJS and NestJS were used, technologies that allow the creation of robust and scalable web applications. Additionally, a PostgreSQL geospatial database was used to efficiently manage large volumes of georeferenced data and develop a BI analytical model to visualize and analyze crime trends.

Keywords: Rappid Application Development, web system, mobile application, Business Intelligence, geo-referencing, public safety, Flutter, NextJS, NestJS

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

DESARROLLO DE UN SISTEMA WEB Y MÓVIL QUE MEJORE LA OBTENCIÓN DE DATOS GEORREFERENCIADOS DE DELITOS CON EL PROPÓSITO DE CREAR UN MODELO ANALÍTICO DE BUSINESS INTELLIGENCE (BI) ORIENTADO AL SEGUIMIENTO DE LAS TENDENCIAS DE SINIESTROS EN LA CIUDAD DE AMBATO

1.1.1 Planteamiento del problema

En [1], desde el 2020 se ha notado un significativo incremento en la cantidad de campos de estudio que llevan a cabo el proceso de toma de decisiones basadas en la información extraída de grandes conjuntos de datos. Según [2], para asegurar que el proceso de toma de decisiones conduzca a los resultados óptimos, es fundamental el uso de Business Intelligence (BI), el cual emplea un conjunto de herramientas que facilitan la adquisición, almacenamiento, análisis y entrega de la información necesaria.

En [3] se menciona que en el ámbito social, el incremento de la delincuencia, el miedo y la inseguridad ciudadana respecto a la zona en la que residen son temas que afectan directamente la calidad de vida de las personas. En este contexto, [1] señala que el análisis de datos sobre la delincuencia ha empezado a tomar una mayor relevancia, siendo estos datos usados de forma objetiva para fundamentar políticas, así como estrategias policiales y operaciones tácticas con el fin de reducir y prevenir la delincuencia mediante la toma de decisiones oportunas y precisas.

La carencia de fuentes adecuadas de datos georreferenciados, con los cuales realizar análisis, dificulta la capacidad de la ciudadanía y las fuerzas del orden para tomar precauciones necesarias [4]. Según [5], los datos georreferenciados o espacio-temporales son un valioso recurso utilizado por analistas criminalistas. Estos datos son la base para la identificación de patrones delictivos en áreas de

alta concentración de delitos a través del mapeo de zonas críticas o puntos calientes (hotspots). El autor en [1] menciona que el uso de software permite la adquisición de datos delictivos de forma más precisa y eficiente, obteniendo información útil que permita identificar los puntos focales del crimen.

En el estudio de [6], según la Comisión Económica para América Latina y el Caribe (CEPAL), en América Latina y el Caribe, la georreferenciación se posiciona como la clave para realizar análisis más profundos y detallados sobre la delincuencia. Esta técnica además facilita la visualización de la distribución espacio-temporal de variables relacionadas con el crimen, permitiendo su análisis posterior. En ese sentido, el autor de [7] menciona que desde 1990 se han establecido paradigmas de entes policiales orientados a la comunidad, los cuales emplean estrategias contra la delincuencia haciendo uso de datos georreferenciados.

Según [8], el boletín semestral de homicidios intencionados en Ecuador, presentado por la Fundación Panamericana para el Desarrollo (PADF), menciona que en 2022, el país registró un total de 4603 homicidios intencionales, marcando así el registro histórico más alto hasta el momento. Se observó un promedio diario de 10 casos, junto a una tasa de 25.9 homicidios por cada 100 000 habitantes. Sin embargo, en el primer semestre de 2023, los casos de homicidios intencionados registrados descendieron a 3599, lo que equivale a un promedio diario de 19 casos, casi el doble de los registros obtenidos en 2022. Debido a esta tendencia, se prevé que la cifra total de homicidios supere los 7000 casos, alcanzando una tasa de 35 homicidios por cada 100 000 habitantes. Esto sitúa a Ecuador como uno de los tres países más violentos en América Latina, solo por detrás de Venezuela y Honduras. Por lo tanto, en [9] se plantea la necesidad de abordar activamente la seguridad en beneficio de la ciudadanía ecuatoriana, proponiendo el uso de tecnologías que permitan establecer estadísticas, interfaces gráficas y mapas con información georreferenciada.

En la ciudad de Ambato, se evidencia la carencia de sistemas que permitan establecer una fuente confiable de datos georreferenciados para analizar exhaustivamente los delitos y sus tendencias criminales. El autor de [10] determinó, a través de encuestas a la ciudadanía del cantón Ambato, que el 26.30% de la población ha sido víctima de

actos delictivos, situando a Ambato en el octavo puesto a nivel cantonal en Ecuador en cuanto al número de denuncias por robos. Según [11], la cantidad de incidentes registrados en Ambato afecta el nivel de confianza de la ciudadanía, debido al temor que sienten de ser víctimas de algún incidente y a la preocupación por la integridad de sus familiares.

1.2 Antecedentes investigativos

En esta sección se muestran todos aquellos trabajos que proporcionan un contexto precedente sobre el trabajo de investigación a realizarse.

Con el propósito de obtener información relevante sobre el tema propuesto para este proyecto se realizó una búsqueda exhaustiva en las revistas IEEE, SPRINGER LINK, así como en los repositorios de GOOGLE SCHOLAR y LA UNIVERSIDAD TÉCNICA DE AMBATO en el periodo de 2015 a 2023, en la tabla 1 se muestran los resultados obtenidos:

Tabla 1. Artículos obtenidos

Cadena	Revista / Repositorio	Número de artículos
(”georeferenced data”	IEEE	28
OR ”spatiotemporal data”	SPRINGER LINK	417
OR ”spatio-temporal	GOOGLE SCHOLAR	1.610
data” OR ”spatio	UNIVERSIDAD	0
temporal data”) AND	TÉCNICA DE AMBATO	
(”crime analysis”		
OR crime OR felony OR		
(”business intelligence”		
OR BI OR ”analytical		
”felony acts”) OR (date		
OR dates OR time OR hour		
OR hours OR ”crime types”)		
Continua en la siguiente página		

Tabla 1 – continuación de la página anterior

Cadena	Revista / Repositorio	Número de artículos
AND model" OR "analytical models") AND ("crime quantity" OR "crime count" OR "number of crimes" OR "total crimes")		

Como se muestra en la Tabla 1, mediante la revisión sistemática se obtuvo un total de 2055 artículos relacionados. Sin embargo, Con la finalidad de obtener información relevante para el desarrollo de este proyecto se descartaron aquellos artículos que no cumplan con los criterios de exclusión e inclusión que se muestran en la tabla 2:

Tabla 2. Criterios de exclusión e Inclusión

Inclusión	Exclusión
• Crimen	• Solo artículos científicos y tesis
• geointeligencia	• Solo textos en español e inglés
• geodata	• No contiene georreferenciación
• BI	
• Analítica	
• Georreferenciación	

Elaborado por: el investigador

Mediante la exclusión realizada se obtuvo un total de 20 artículos que contiene información sobre datos georreferenciados. Aunque los artículos obtenidos no cumplen exactamente con el objetivo propuesto para esta investigación, se destaca a continuación los trabajos más relevantes para el desarrollo de este proyecto.

Los autores de [12] mencionan las ventajas del uso de los Sistemas de Información Geográfica (GIS). Estas son herramientas masivas utilizadas para mostrar información de interés en mapas. Además, destacan las ventajas del empleo de herramientas de Business Intelligence para el procesamiento de grandes cantidades de datos, especialmente en el contexto de datos georreferenciados, denominado Geo Business Intelligence (GeoBI). Esta disciplina se encarga de abordar el análisis de datos

geoespaciales para mejorar la toma de decisiones en las organizaciones. En el trabajo se propone crear un sistema de Información Geográfica Voluntaria, del inglés (VGI) el cual conste de un portal web. Este sistema habilitará el manejo de usuarios y visualización de mapas. También se utilizará una aplicación cliente sea web o móvil, para ingresos de datos y una base de datos geográfica (GeoDB) con la cual almacenar la información geográfica voluntaria. En el sistema se hará uso del framework ”Monitor, Analyze and Drill to Detail” (MAD), el cual permite separar las necesidades de los usuarios dependiendo al perfil que tengan asignado, como administradores, analistas y entes de control de una organización. Se menciona que los sistemas GIS deberían proporcionar ciertos elementos básicos, como portales web, aplicaciones clientes, servidores de mapas y bases de datos con capacidades de almacenamiento de datos geográficos. Sin embargo, todo lo planteado en el trabajo es una propuesta por parte de los autores y no se muestra un prototipo funcional.

En [13] el trabajo propuesto se centra en el uso de machine learning para la predicción de delitos, específicamente en el uso de deep learning. En el trabajo se divide un mapa en cuadriculas, con el fin de verificar si en una cuadricula se a detectado un crimen y tomarlo como un punto caliente. Para este modelo se tomaron en cuenta siete patrones espacio temporales, las cuales son: mes, mes siguiente, año pasado, cuadrículas rodeadas en ocho direcciones en el mes actual, cuadrículas rodeadas en ocho direcciones el año pasado, tendencia y proporción, más no se menciona de donde fueron extraídas tales características ni mediante que medios. No obstante, aunque no se trate el tema del software en la adquisición de los datos georreferenciados parad delitos, las características que se muestran aportan información valiosa para ser utilizada en el contexto de modelos analíticos de BI.

En [14], el autor propone un modelo de predicción de delitos basado en el uso de datos espacio-temporales y el método Kernel Density Estimation (KDE), frecuentemente utilizado para estimar la densidad de crímenes. En el trabajo propuesto, se detalla el flujo de trabajo para el preprocesamiento de los datos, que implica la eliminación de características irrelevantes y valores geográficos fuera de la zona de estudio. El autor también menciona que el número de observaciones es de 1,762,311 y detalla las características del dataset, como fecha, categoría del crimen, día de la semana,

distrito policial, resolución del crimen, dirección, latitud y longitud. A pesar de esto, en el trabajo no se menciona el uso de software para la adquisición de datos georreferenciados, ya que el dataset proviene de Kaggle. No obstante, el artículo es importante para el desarrollo de este proyecto, ya que ofrece datos prácticos que pueden ser útiles en la creación de modelos analíticos de Business Intelligence (BI).

Los autores de [15] proponen el algoritmo Criminal Activity Clustering (CAC) el cual está basado en la clusterización difusa de puntos delictivos con la finalidad de agrupar grandes volúmenes de datos georreferenciados de delitos, así como identificar patrones de comportamiento de los siniestros. En el trabajo se muestra una descripción de las características utilizadas en el desarrollo del algoritmo, las cuales son: fecha, lugar del incidente, información del incidente, información del ataque, información de la víctima, información del perpetrador, información del arma, causalidades y consecuencias, estas características son obtenidas desde la base de Datos de Terrorismo Global, del inglés (GTD). Tales características propuestas son útiles para su uso en modelos analíticos de BI. Sin embargo, el trabajo como tal no aborda el uso de software para la adquisición de datos georreferenciados o su uso en BI.

En el trabajo propuesto en [16] no se aborda el uso de software para la adquisición de datos georreferenciados de crímenes ni su utilización en modelos de BI. En su lugar, se tratan marcos para el análisis de datos geoespaciales, los cuales incluyen la recopilación, normalización, geolocalización y almacenamiento de los datos. Estos marcos permiten crear un algoritmo de clustering del lado del servidor para mapas, capaz de manejar una gran cantidad de datos georreferenciados de forma masiva. En el trabajo se presentan los diagramas de entidad-relación con los tipos de datos utilizados para desarrollar el algoritmo.

En el trabajo presentado en [17], se propone un modelo de predicción de delitos basado en el uso del algoritmo de K-Nearest Neighboring (KNN). Aunque en el trabajo propuesto no se considera el uso de software para la adquisición de datos georreferenciados de delitos ni el desarrollo de un modelo analítico de BI, se muestra cómo los datos fueron preprocesados eliminando valores nulos y datos innecesarios, obteniendo las siguientes características importantes: Hora, latitud, longitud, año, mes

y semana del año. Aun así, el autor menciona que los atributos obtenidos pueden no ser suficientes, por lo cual se deberían encontrar una mayor cantidad de atributos importantes relacionados con los crímenes.

En [18] el autor se centra principalmente en el uso de algoritmos de aprendizaje supervisado para la predicción de delitos. La finalidad del uso de estos algoritmos es proporcionar información temprana sobre la actividad delictiva a entidades de control y agentes de la ley. El trabajo presenta los atributos propuestos para el dataset tales como: fecha, categoría, día, distrito policial, resolución del crimen, dirección, latitud y longitud. Estos datos son extraídos del sistema de reportes de incidentes del Departamento de Policía de San Francisco (SFPD). Sin embargo, el trabajo no aborda el uso de software para la adquisición de datos georreferenciados de crímenes ni su uso para modelos analíticos de Business Intelligence (BI). A pesar de ello, este trabajo resulta relevante para el desarrollo del proyecto actual, ya que proporciona datos útiles que pueden ser empleados en la creación de un modelo analítico de BI.

En [19] se muestra un prototipo de una aplicación web desarrollada utilizando el patrón de arquitectura cliente-servidor en Visual Studio 2017 Community, utilizando el lenguaje de programación C# junto al framework ASP.NET, también la aplicación utiliza el API de google map. Esta aplicación permite recolectar y mostrar datos georreferenciados de delitos en la ciudad de Ambato. El trabajo propuesto exhibe el modelo de la base de datos, desarrollado en SQL Server y las interfaces propuestas para el prototipo. La aplicación desarrollada permite a los usuarios reportar delitos mediante un formulario y muestra los puntos de siniestros mediante un mapa de calor de la ciudad. Además, permite visualizar reportes de los delitos desde un punto de vista cualitativo y cuantitativo mediante el uso de la herramienta Power BI integrada en la aplicación web. No obstante, el trabajo presentado no profundiza en el uso de los datos obtenidos para generar modelos analíticos de BI.

En [20] se destaca el uso de la georreferenciación en sistemas de seguridad en el cual se propone una aplicación móvil desarrollada en IONIC usando el lenguaje de programación Javascript. La aplicación permite el envío de geolocalización mediante notificaciones push utilizando el servicio de one signal. El backend de esta

aplicación esta desarrollado en NodeJS utilizando express y como base de datos utiliza MongoDB. El objetivo de esta aplicación es ayudar acometer la problemática social de la inseguridad ciudadana en la parroquia Santa Rosa de la ciudad de Ambato.

En [21] se describe el desarrollo de una aplicación móvil que se encarga de enviar la ubicación en tiempo real para el transporte privado de la empresa Plasticaucho S.A. de la ciudad de Ambato. Esta aplicación móvil utiliza Xamarin Forms con el lenguaje de programación C#. En el backend, se empleó el framework ASP.NET, también utilizando el lenguaje de programación C#, e implementando el patrón de arquitectura Model View Controller (MVC), junto a una base de datos en PostgreSQL. El proyecto se llevó a cabo utilizando la metodología de desarrollo ágil Extreme programming (XP).

En [22], se desarrolló una aplicación web que incorpora georreferenciación. Esta aplicación utiliza el framework Angular junto con el lenguaje de programación JavaScript para el frontend. El backend fue implementado mediante el framework ASP.NET utilizando el lenguaje de programación C#, y se empleó una base de datos MySQL. Para el desarrollo de este proyecto, se aplicó la metodología de desarrollo Extreme Programming (XP), orientada a grupos reducidos de desarrollo. La aplicación creada permite gestionar servicios, notificaciones de pago y geolocalización de viviendas para la empresa Optynet.

Los autores en [23] presentan el diseño y desarrollo de una aplicación móvil que aborda la problemática de la predicción de crímenes. Esta aplicación se fundamenta en la integración de tecnologías modernas, la recopilación de datos geoespaciales, el análisis de datos e inteligencia artificial. En el trabajo, se detallan las técnicas utilizadas para la recopilación de información. Se menciona el empleo de la herramienta Taled Open Studio, la cual se utilizó para llevar a cabo el proceso de Extracción, Transformación y Carga de datos (ETL), almacenándolos en una base de datos MySql. Además, el proyecto exhibe las interfaces de la aplicación móvil, que constan de un módulo de autenticación, otro de reporte de delitos y un tercer módulo para la visualización de estos. Dentro del módulo de visualización de delitos, se presentan mapas de calor que muestran los incidentes ocurridos, agrupándolos por áreas mediante esferas de colores que representan el total de siniestros en cada zona. Por otro lado, el módulo de reporte

de delitos permite a los usuarios informar los incidentes de los que han sido víctimas mediante un formulario y un mapa para señalar la ubicación del suceso. Además, la aplicación ofrece estadísticas sobre los incidentes reportados en distintos periodos de tiempo para el conocimiento de la ciudadanía. Este trabajo proporciona un medio para la adquisición de datos georreferenciados mediante el uso de software, así como una plataforma para mostrar información relevante a la ciudadanía. No obstante, no incluye detalles sobre las características utilizadas ni los diagramas de la base de datos. Tampoco aborda el uso de modelos analíticos de Business Intelligence (BI).

1.3 Fundamentación teórica

Esta sección establecerá las bases, fundamentos y teorías pertinentes a través de una exhaustiva revisión de antecedentes. Esto permitirá respaldar y contextualizar el desarrollo de la presente investigación, ofreciendo un marco sólido que sustente cada paso del estudio.

1.3.1 Metodología ágil

Una metodología ágil es un conjunto de técnicas aplicadas utilizadas con el propósito de facilitar los tiempos de desarrollo y obtener un producto final de calidad. Los autores de [21] y [22] señalan el uso de la metodología ágil Extreme Programming (XP). Los autores mencionan la flexibilidad y facilidad de modelado que ofrece la metodología ágil XP, además, de mejorar los tiempos de desarrollo, calidad del producto y ayuda a minimizar los errores durante el desarrollo. También se indica que la metodología XP está orienta a grupos de trabajo pequeño, además, de adaptarse mejor a las necesidades de los proyectos ya que es versátil al incorporar cambios durante el desarrollo. Así mismo los autores señalan que una de las principales ventajas de la metodología XP es la capacidad de dividir el trabajo en pequeñas historias de usuario las cuales son completamente funcionales, lo cual garantiza la entrega del producto final en etapas manejables.

En [20] se utilizó la metodología de desarrollo Mobile-D, ya que combina beneficios de

metodologías como XP, Crystal y Rational Unified Process (RUP). El autor menciona que la metodología Mobile-D se enfoca en el desarrollo de aplicaciones móviles, especialmente útil en aplicaciones pequeñas con pocos módulos y vistas. Además, ofrece un estilo de desarrollo por iteraciones, el cual es rápido y útil para grupos pequeños.

1.3.2 Framework Web

Un framework web constituye un conjunto de herramientas y librerías esenciales para el desarrollo de aplicaciones web. En [22], el autor destaca Angular como un framework que posibilita el diseño de sistemas web estables y escalables. Esto se debe a las capacidades que ofrece para crear aplicaciones Single Page Application (SPA) eficientes. Asimismo, menciona que Angular proporciona una arquitectura basada en componentes y módulos organizados, los cuales se integran fácilmente con otras tecnologías, como el caso de ASP.NET Core, utilizado como plantilla de interfaz de usuario.

1.3.3 Framework Backend

Un framework backend consiste en un conjunto de herramientas y librerías predefinidas que permiten desarrollar y mantener la lógica de una aplicación en el lado del servidor. En [22], el autor menciona que ASP.NET es un framework backend gratuito, seguro, de alto rendimiento y de código abierto. El autor destaca que ASP.NET permite ejecutar aplicaciones web en sistemas operativos como Windows, macOS y Linux utilizando C# en el lado del servidor.

El autor de [20] destaca a Node.js como tecnología backend la cual permitiendo el manejo eficiente de múltiples procesos simultáneamente. Además, se menciona que la curva de aprendizaje de Node.js es baja, al igual que su costo de desarrollo. También se resalta que Node.js ofrece una manera sencilla de crear APIs Representational State Transfer (REST) utilizando Express, las API REST permiten comunicar diferentes sistemas mediante los protocolos Hypertext Transfer Protocol (HTTP), lo que facilita

la creación rápida y flexible de un backend completo.

1.3.4 Aplicación Móvil

Una aplicación móvil se refiere a un tipo de aplicación diseñada para ejecutarse en teléfonos inteligentes o tabletas. En [20], el autor destaca que IONIC, utilizado en el desarrollo de aplicaciones móviles, presenta una curva de aprendizaje reducida, ya que puede combinarse con frameworks como Angular y librerías de componentes como React y Svelte. Además, menciona que IONIC permite la creación de componentes reutilizables, lo que conlleva a un código más legible y fácil de mantener. Asimismo, resalta su extensa documentación respaldada por una gran comunidad, lo que facilita la solución a errores de manera más sencilla. Por otra parte, en [21], el autor enfatiza el uso de Xamarin Forms para el diseño de aplicaciones móviles debido a su capacidad para crear aplicaciones multiplataforma. Expone que Xamarin permite el desarrollo de aplicaciones utilizando el lenguaje C#, el cual se compila en código nativo para Android e iOS. Además, al ser una tecnología de Microsoft, posibilita la integración sencilla de paquetes externos mediante NuGet.

1.3.5 Bases de datos

Una base de datos es un sistema electrónico que permite el almacenamiento de datos de forma organizada. En [22], el autor destaca que MySQL, como Sistema Gestor de Bases de Datos Relacionales (SGBDR), ofrece un alto rendimiento y fiabilidad, lo que permite almacenar, buscar, ordenar y obtener información de forma eficiente. También se indica que MySQL es uno de los SGBDR más populares, siendo de código abierto y respaldado por Oracle. En el mismo sentido, en [21], el autor resalta a PostgreSQL como SGBDR por ser uno de los gestores más rápidos y seguros del mundo. Expone su gran escalabilidad, facilidad de uso y capacidad para manejar no solo variables primitivas, sino también variables tipo objeto. Además, señala que la base de datos de PostgreSQL es ilimitada, permitiendo un número ilimitado de campos por tabla.

En otro enfoque, el autor de [20] menciona a MongoDB como una base de datos NoSQL

capaz de manejar una gran cantidad de información. Se destaca su carácter de código abierto y su flexibilidad para trabajar con diversas tecnologías de backend. Además, se resalta su escalabilidad al estar vinculada a servidores de Amazon, Google y Microsoft, ofreciendo también planes gratuitos convenientes para aplicaciones pequeñas.

1.3.6 Mapa web

Un mapa web es una visualización interactiva de información geográfica que se puede abrir en navegadores y dispositivos móviles. En [19], el autor destaca que el Application Programming Interface (API) proporcionado por Google Maps para renderizar mapas ofrece varias características y prestaciones, siendo uno de los productos más importantes de Google. También menciona que una de las principales ventajas de esta API es su capacidad para integrarse con distintos lenguajes de programación y, por ende, con diferentes frameworks. Esto hace que sea altamente flexible para su uso en diferentes sistemas. Por otro lado, en [23], el autor señala que Leaflet permite crear mapas interactivos y aplicaciones de mapas en línea, además de ser de código abierto. También destaca la amplia gama de herramientas disponibles en Leaflet que facilitan la visualización de datos georreferenciados. Incluso, ofrece a los desarrolladores la capacidad de crear experiencias de usuario altamente interactivas y personalizables, lo cual resulta ideal para aplicaciones tanto móviles como web.

1.3.7 Geo Business Intelligence (GEOBI)

En [12] se empleó GEOBI dado que esta mejora la toma de decisiones en organizaciones. El GEOBI proporciona información más precisa y oportuna incorporando la dimensión espacial en la información. También permite que sus componentes se vuelvan geográficos, es decir, en lugar de El procesamiento analítico en línea, del inglés (OLAP) se utiliza una combinación de un Sistema de información geográfica (SIG) y un cubo OLAP denominado SOLAP, además, sus bases de datos se convierten en GeoDB, bases de datos que contienen información geográfica.

1.3.8 Patrón MVC

En [21], el autor menciona la importancia del patrón Model View Controller (MVC), el cual permite separar una aplicación en tres capas de componentes: Modelo, Vista y Controlador. El autor señala que la separación de funciones de este patrón de arquitectura mejora la mantenibilidad del código y hace más sencillo depurar errores. El autor indica que el uso del patrón MVC permite disminuir los errores, dado que las dependencias no están distribuidas en diferentes secciones del código y, por lo tanto, evita mezclar la lógica de negocio con las interfaces de usuario.

1.3.9 Diagrama UML

En los artículos revisados no se señala el uso de diagramas Unified Modeling Language (UML) en el desarrollo de sistemas de software. Sin embargo, se considera necesario mencionarlo para el desarrollo de este proyecto. UML es una representación visual y semántica de la arquitectura, diseño e implementación de sistemas de software. Los diagramas UML permiten describir los límites, la estructura, el comportamiento y los objetos que contiene un sistema. Aunque este no es un lenguaje de programación, mediante herramientas externas puede ser utilizado para la generación de código para diferentes lenguajes. Según el Object Management Group (OMG), el propósito de UML es proveer a los ingenieros, desarrolladores y arquitectos de software un medio por el cual puedan realizar análisis, diseño e implementación de sistemas de software. Aunque UML no está enfocado en bases de datos, este permite crear modelos de datos conceptuales de alto nivel.

1.3.10 NextJS

En los artículos revisados no se menciona el uso de NextJS como framework frontend para el desarrollo de aplicaciones web. Sin embargo, se considera necesario mencionarlo para el desarrollo de este proyecto. NextJS es un framework web el cual utiliza la librería de componentes React la cual es desarrollada por Meta. NextJS

permite crear aplicaciones web completas utilizando React para crear interfaces de usuario dinámicas y rápidas basadas en componentes y añadiendo características adicionales y optimizaciones tales como enrutador basado en archivos, compilación y empaquetado de la aplicación, optimización de imágenes, Renderizado del Lado del Servidor, del inglés (SSR), Renderizado del Lado del Cliente, del inglés (CSR), Generación de Sitios Estáticos del inglés (SSG). NextJS permite desarrollar un backend basado en NodeJS con el cual crear un API REST en la misma aplicación.

1.3.11 NestJS

En los artículos revisados no se menciona el uso de NestJS como framework backend para el desarrollo de API REST. Sin embargo, se considera necesario mencionarlo para el desarrollo de este proyecto. NestJS es un framework backend basado en NodeJS el cual permite diseñar aplicaciones en el lado del servidor como. NestJS combina elementos de la Programación Orientada a Objetos (POO), Programación Funcional (PF) y Programación Reactiva Funcional (PRF) junto con con TypeScript para diseñar aplicaciones eficientes y escalables. NestJS brinda un nivel de abstracción a framework de servidor HTTP sólidos como Express y Fastify, además de exponer su API directamente con lo cual permite a los desarrolladores implementar modules externos disponibles para la plataforma.

1.3.12 Flutter

En los artículos revisados, no se menciona el uso de Flutter como framework para el desarrollo de aplicaciones móviles; sin embargo, se considera necesario mencionarlo para el desarrollo de este proyecto. Flutter es un framework desarrollado por Google para el desarrollo de aplicaciones móviles, web y de escritorio de forma nativa con solo código base. Permite crear interfaces de usuario visualmente atractivas al manejar los elementos de la interfaz en componentes más pequeños llamados widgets. Flutter tiene un alto rendimiento, dado que utiliza Dart como lenguaje de programación, el cual compila código a máquina, permitiendo que las aplicaciones desarrolladas sean

rápidas y eficaces. Además, Flutter ofrece una buena experiencia de desarrollo al incluir herramientas que permiten realizar actualizaciones en caliente y revisar el estado de los componentes de la aplicación.

1.4 Objetivos

1.4.1 Objetivo general

Implementar un sistema web y móvil que mejore la recolección de datos georreferenciados precisos de delitos en la ciudad de Ambato.

1.4.2 Objetivos específicos

- Analizar frameworks para el desarrollo web y móvil, enfocados en sistemas para la obtención de datos georreferenciados.
- Evaluar la precisión de los datos georreferenciados mediante el empleo de técnicas estadísticas para determinar la fiabilidad de los datos obtenidos.
- Desarrollar un sistema web y móvil para la obtención de datos georreferenciados de delitos con los cuales desarrollar un modelo analítico de Business Intelligence (BI) orientado al seguimiento de las tendencias de siniestros en la ciudad de Ambato.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

En esta sección se especificarán los elementos requeridos para el desarrollo del presente proyecto. Se muestran aspectos importantes como el equipo necesario, materiales, tecnologías y cualquier otro elemento vital para llevar a cabo las actividades programadas.

2.1.1 Institucionales

Para el desarrollo de esta investigación, no se cuentan con recursos institucionales, ya que el proyecto carece de relación con cualquier institución o empresa. Esto implica la ausencia de respaldo financiero, infraestructura o apoyo logístico proveniente de entidades institucionales o empresariales.

2.1.2 Humanos

- Usuarios de la aplicación.

2.1.3 Materiales

- Laptop.
- Resma de papel bond.
- Impresora.
- Tinta.
- Material de escritorio.
- Smartphone.

- Internet.
- Servicio de hosting web.
- Servicio de hosting backend.
- Servicio de hosting BD.
- Herramientas de desarrollo (IDE's).

2.1.4 Económicos

Tabla 3. Presupuesto

N.	Detalle	Unidad	Cantidad	Valor Unitario	Valor Total
1	Resma de papel bond	c/u	1	$05.00	$05.00
2	Impresora	c/u	1	$200.00	$200.00
2	Material de escritorio	c/u	1	$05.00	$05.00
2	Herramientas de desarrollo (IDE's)	c/u	1	$00.00	$00.00
2	Tinta	c/u	1	$10.00	$10.00
3	Laptop	c/u	1	$950.00	$950.00
4	Smartphone	c/u	1	$350.00	$350.00
5	Servicio de internet	c/u	4	$25.00	$100.00
6	Servicio de hosting web	c/u	1	$20.00	$20.00
7	Servicio de hosting api	c/u	1	$20.00	$20.00
8	Servicio de hosting BD	c/u	1	$20.00	$20.00
8	Personas (Usuarios)	c/u	1	$00.00	$00.00
Subtotal					$1680
Imprevistos (%15)					$252
Total					$1932

Elaborado por: el investigador

2.2 Métodos

En esta sección se describe la metodología que se empleará en el desarrollo de la presente investigación, tras haber analizado tanto los antecedentes investigativos como el marco teórico.

2.2.1 Modalidad de investigación

En esta sección se muestran los enfoques a los que se encuentra orientada la investigación propuesta para este proyecto. A continuación, se detalla cada una de ellas.

Investigación cuantitativa

En el presente proyecto se emplea una metodología de investigación cuantitativa. Esta se centra en la recopilación de datos numéricos y estadísticos que permiten establecer las tendencias de siniestros a través del análisis del número de crímenes. Esta información se utiliza como base fundamental para el diseño de un modelo analítico de BI.

Investigación aplicada

Se emplea una investigación aplicada, ya que el desarrollo del proyecto propuesto utilizará la teoría, métodos, técnicas, tecnologías y conocimientos adquiridos a lo largo de ciclos académicos previos. Este enfoque permitirá aplicar de manera práctica y específica el conocimiento acumulado durante la trayectoria educativa.

2.2.2 Población y muestra

La población de estudio está conformada por los incidentes delictivos ocurridos en la ciudad de Ambato. Al no contar con una estimación adecuada sobre la cantidad de incidentes requerida para estipular la población, se optó por definir una muestra infinita con población desconocida.

- n = tamaño de la muestra
- Z = nivel de confianza, un nivel de confianza habitual es de 90% con una puntuación estándar de 1.645
- p = probabilidad de éxito o proporción esperada
- q = probabilidad de fracaso
- pq = varianza de la población, cada una equivale a 0.50, (p=q=0.50)
- e = error de estimación máximo aceptable, el error habitual es de 5% con una puntuación estándar de 0.05

$$n = \frac{Z^2 \cdot p \cdot q}{e^2}$$

$$n = \frac{(1.645)^2 \cdot 0.50 \cdot 0.50}{(0.05)^2}$$

$$n = 270.60 \approx 270$$

Con un nivel de confianza del 90% y un error de estimación máximo aceptable del 5%, se obtiene una muestra aproximada de 270 incidentes.

2.2.3 Recolección de información

La información fundamental para este proyecto se obtuvo a través de entrevistas con un profesional experto en el campo de BI y un coronel de la policía nacional. Además, se elaboró un protocolo técnico para la recopilación de datos delictivos.

Es relevante señalar que las respuestas proporcionadas a continuación no son transcripciones literales de las entrevistas; se han reformulado para ajustarse al contexto de la investigación en curso.

Tabla 4. Entrevista aplicada al Ing. Rubén Nogales experto en Business Intelligence

N. Pregunta	Respuesta	Observación
1	Debe ser medible, debe tener la facilidad de ser medido, la variable del KPI debe cumplir con la función de agregación, es decir, en su momento dado pueda generar procesos aritméticos no solo en los hechos sino en las dimensiones.	La relevancia de un KPI radica en su capacidad de ser medible y agregable, lo que sugiere la importancia de que proporcione una visión general y permita comparaciones.
2	Semanal es una buena frecuencia a menos que los datos que se generen tengan una gran retroalimentación.	La frecuencia semanal de actualización de informes de BI se sugiere como un equilibrio entre actualización oportuna y necesidad de datos actualizados para la toma de decisiones.
Continua en la siguiente página		

Tabla 4 – continuación de la página anterior

N. Pregunta	Respuesta	Observación
3	Depende del caso, si los datos necesitan ser comparados, lo mejor son gráficos de barras, si son frecuencias, se deben visualizar histogramas, si en algún momento la información es en función del tiempo, deben ser por medio de gráficos de líneas, y si se desea mostrar porcentajes puede ser mediante un gráfico de pie.	La adaptabilidad en el formato de presentación de datos destaca la necesidad de ajustarse al tipo de información para facilitar su comprensión y análisis.
4	Que los datos no provean la suficiente información, es decir, el dashboard generado no permita ver claramente lo que se desea para poder tomar decisiones, siendo el principal desafío la organización de los datos.	La organización de los datos se identifica como el principal desafío al interpretarlos, lo que destaca la importancia de una estructura clara y comprensible.
5	Que sea visual, dinámico, fácilmente entendible, con el cual se pueda realizar filtros y se pueda obtener fácilmente lo que se necesita.	La efectividad de un sistema de BI para la toma de decisiones se relaciona con su accesibilidad, visualización dinámica y comprensibilidad para los usuarios finales.

Continua en la siguiente página

Tabla 4 – continuación de la página anterior

N. Pregunta	Respuesta	Observación
6	Herramientas para limpieza de datos, completación de datos como cluster, así como herramientas para leer y guardar datos estructurados y no estructurados. También herramientas para visualizar la información como Power BI, Tableau y JS para generar gráficos dinámicos.	Se enfatiza la importancia de herramientas de limpieza y visualización de datos, así como la capacidad para manejar datos estructurados y no estructurados.
7	Teniendo un buen modelo entidad-relación dimensional, también se debe contar con un modelado de casos de uso dimensional bien establecido.	Un modelo dimensional sólido y un modelado de casos de uso bien establecido se destacan como clave para abordar problemas de escalabilidad en BI.
Continua en la siguiente página		

Tabla 4 – continuación de la página anterior

N. Pregunta	Respuesta	Observación
8	Se debe realizar un EDA, un Análisis Exploratorio de Datos completo tomando en cuenta que los datos que se obtengan sean de carácter independiente. También definir la distribución de los datos y de cada una de las variables, revisar que los datos sean completos, que no cuenten con ambigüedad, que no haya campos en blanco ni repetidos y también la anonimidadde los datos es importante.	Un análisis exploratorio completo y la garantía de calidad de datos son esenciales para asegurar la confiabilidad de los informes de BI.

9	Definir los KPI obtenidos de los modelados de caso de uso dimensional.	La identificación de requisitos de usuarios se sugiere a través de la definición de KPI basados en modelados de casos de uso dimensional, lo que alinea los informes con las necesidades específicas de los usuarios.
Conclusión: La entrevista resalta la importancia crítica de una sólida estructura de datos y un enfoque meticuloso en la interpretación y presentación de la información en el ámbito de Business Intelligence (BI). Desde la definición de KPI relevantes hasta la selección adecuada de herramientas de análisis y visualización, cada aspecto destaca la necesidad de una estrategia integral que permita convertir los datos en acciones concretas. También se señala que una efectiva estrategia de BI requiere no solo tecnología avanzada, sino también una comprensión profunda de las necesidades del usuario y una atención constante a la calidad y confiabilidad de los datos.		

Tabla 5. Entrevista aplicada al teniente coronel de estado mayor Christian Iván Quintana Guerra

N. Pregunta	Respuesta	Observación
1	Todos los eventos delictivos que sucedan en el territorio provienen de una demanda ciudadana. Con base en la información recibida del demandante, se recopilan datos georreferenciados de los eventos delictivos, así como horas, fechas, los tipos de personas involucradas en el evento y el modus operandi, lo cual permite realizar los análisis respectivos y tomar las acciones adecuadas.	La precisión en la recopilación de datos es vital para la gestión de seguridad pública, especialmente a través de la georreferenciación y la categorización detallada de eventos.
2	La calidad aproximada se estima en alrededor del 70%, tomando en cuenta que los ciudadanos en varios casos confunden los diferentes delitos, siendo el caso más común la confusión entre robo y hurto, lo que resulta en reportes inconsistentes.	La calidad de los datos sobre delitos se ve afectada por la inconsistencia en los reportes ciudadanos
3	Uno de los desafíos que se encuentra comúnmente es que los demandantes proporcionen los datos necesarios para realizar un trabajo más efectivo.	La obtención de datos completos y precisos de los demandantes representa un desafío significativo para la policía en la recopilación y análisis de datos.

Continua en la siguiente página

Tabla 5 – continuación de la página anterior

N. Pregunta	Respuesta	Observación
4	Los datos son recopilados de forma continua en una matriz de Excel.	Los datos sobre delitos se recopilan en una matriz de Excel de forma continua, sugiriendo una falta de sistemas más avanzados de gestión de datos.
5	El sistema debería proporcionar una información completa sobre un evento delictivo específico con la cual se pueda identificar datos de los agresores, tales como modus operandi, nacionalidad, etnia, altura y contextura. Esto permitiría identificar posibles sospechosos de estar involucrados en algún delito.	Existe una necesidad de sistemas de análisis de delitos que proporcionen información detallada sobre los agresores para facilitar la identificación de sospechosos.
6	Contar con las protecciones necesarias en las bases de datos.	Se reconoce la importancia de medidas sólidas de seguridad y privacidad de datos, aunque no se especifican detalladamente.
7	Esto facilitaría el poder compartir información entre las diferentes unidades con la finalidad de realizar un trabajo más eficiente.	Los sistemas de análisis de delitos pueden mejorar la colaboración entre unidades policiales al facilitar el intercambio eficiente de información.

Continua en la siguiente página

Tabla 5 – continuación de la página anterior

N. Pregunta	Respuesta	Observación
8	El nivel de capacitación debe ser alto y constante, dado que uno de los mayores problemas que se enfrenta actualmente es que las promociones más recientes de los miembros de la policía cuentan con conocimientos limitados en cuanto a las nuevas tecnologías.	Se destaca la necesidad de una capacitación constante para el personal policial en el uso efectivo de sistemas de análisis de delitos, especialmente para abordar la brecha tecnológica entre las promociones más recientes.
Conclusión: La entrevista con el teniente coronel Christian Iván Quintana Guerra resalta varios desafíos clave en la gestión de seguridad pública, particularmente en la recolección y análisis de datos delictivos. A pesar de la alta demanda ciudadana y la implementación de una matriz de Excel para la recopilación continua de datos, existen problemas significativos con la calidad y precisión de los reportes debido a la confusión de los ciudadanos entre diferentes tipos de delitos. Además, la obtención de datos completos y precisos de los demandantes sigue siendo un reto importante. Es evidente la necesidad de sistemas de gestión de datos más avanzados y seguros, así como de una capacitación constante y especializada para el personal policial en nuevas tecnologías..5		

La recopilación de información para este proyecto se realizó mediante un protocolo técnico que facilitó la obtención de datos necesarios para el desarrollo de la propuesta. A continuación, se explica el procedimiento utilizado para recopilar la información.

Se diseñó un formulario integrado en una aplicación web para recopilar detalles sobre las víctimas de los incidentes en la ciudad de Ambato. Este formulario permitirá el registro de información relevante, como el tipo de delito, la fecha y hora del suceso, y también incluirá un mapa para obtener la ubicación exacta del incidente. Además, se obtendrán datos a través de alertas de peligro que podrán ser enviadas mediante la

aplicación móvil de botón de pánico que se desarrollara en el proyecto. En la Figura 1 se muestra el formulario diseñado para la recopilación de información.

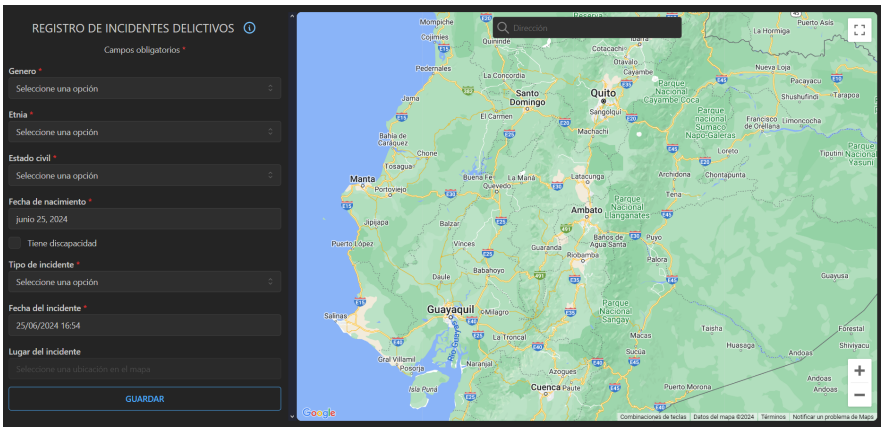


Figura 1. Formulario de recolección de datos

Mediante este formulario, se recopilarán la cantidad de incidentes establecidos en la muestra, que se estima en 270 incidentes. La información recopilada se utilizará para el análisis de datos y la implementación de la propuesta con el modelo analítico de BI.

2.2.4 Procesamiento y análisis de datos

Este apartado se centrará en la conclusión de la entrevista y en el procesamiento de los datos obtenidos del protocolo técnico. Estos datos serán fundamentales para desarrollar un modelo analítico de Business Intelligence (BI) dirigido al seguimiento de las tendencias de siniestros en la ciudad de Ambato.

En base a la información recopilada de las entrevistas, se puede concluir que:

- La relevancia de los KPIs es crucial para la toma de decisiones informadas en BI. Es fundamental que los KPIs sean medibles y fácilmente agregables, permitiendo análisis aritméticos para proporcionar una visión clara del rendimiento.
- La actualización regular de informes es esencial. Se sugiere una frecuencia semanal, aunque puede variar dependiendo de la retroalimentación necesaria de los datos.
- La elección del formato de presentación de datos juega un papel importante en la comprensión y análisis de la información. Los gráficos de barras son preferibles

para comparaciones, mientras que los gráficos de líneas son ideales para datos temporales.

- Se requiere un nivel alto y constante de capacitación, especialmente para los miembros más recientes, debido a la necesidad de adaptarse a nuevas tecnologías.
- La interpretación efectiva de datos en BI enfrenta desafíos, especialmente en la organización de los datos para una visualización clara y la identificación de patrones significativos.
- Un sistema de BI efectivo debe ser visual, dinámico y fácilmente entendible, permitiendo filtros para obtener la información necesaria de manera rápida y precisa.
- Es esencial contar con medidas de seguridad adecuadas para proteger la información sensible relacionada con incidentes delictivos.
- La recopilación continua de datos se realiza en una matriz de Excel, destacando la necesidad de modernizar los métodos de recopilación y análisis de datos.
- Los sistemas de análisis de incidentes delictivos deben proporcionar información completa sobre eventos delictivos y detalles de agresores para facilitar la identificación de sospechosos potenciales.
- Se requiere un nivel alto y constante de capacitación para utilizar eficazmente sistemas de análisis de incidentes delictivos, especialmente debido a la evolución de las tecnologías.

En relación con el protocolo técnico, los datos recolectados a través del formulario web serán procesados y almacenados en una base de datos específica para ello. En los cuales se aplicará un proceso ETL así como se emplearán algoritmos de aumento de datos, con la finalidad de obtener una cantidad considerable de elementos con los que trabajar. Esta información será fundamental para la creación de un cubo de Procesamiento Analítico en Línea, del inglés (OLAP), que posibilitará el análisis exhaustivo de dichos datos y, a su vez, permitirá la generación de informes detallados.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis de procesos

a. Análisis del proceso propuesto

En la Figura 2 se presenta el proceso propuesto para él envió de alertas de emergencia con ubicación en tiempo real, en donde:

1. Si el usuario no se encuentra registrado en el sistema:
    (a) El usuario se registra en el sistema.
    (b) El usuario inicia sesión en el sistema.
2. El usuario ingresa miembros a su grupo familiar.
3. El usuario selecciona un tipo de incidente.
4. El usuario envía la alerta de emergencia presionando el botón de pánico durante 3 segundos.
5. Se envía una notificación a los miembros del grupo familiar del usuario y los policías dentro de la zona de emergencia junto con la ubicación en tiempo real.
6. El Ecu 911 recibe la alerta de emergencia y la asigna a la entidad correspondiente.
7. Si el tipo de incidente no es el correcto:
    (a) Ingresar la información correcta sobre el incidente.
    (b) La emergencia es atendida.

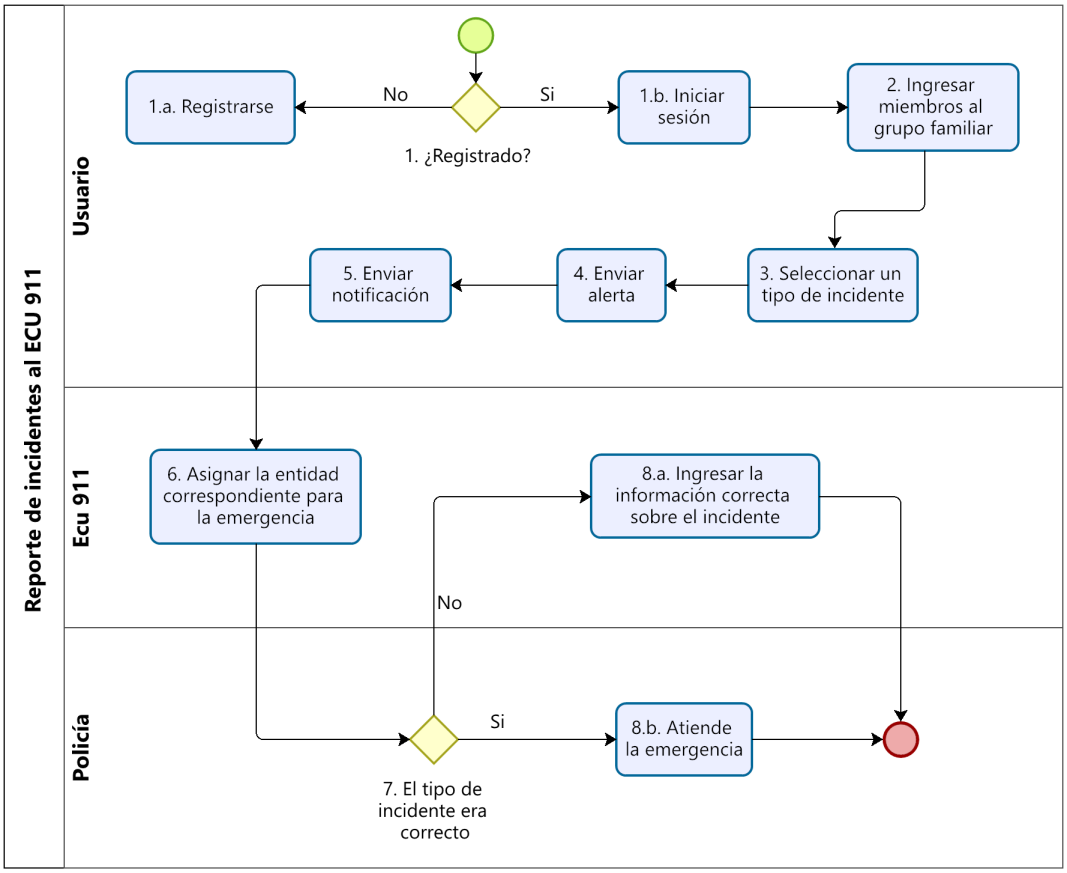


Figura 2. Proceso propuesto de reporte de incidentes al ECU 911.

b. Análisis del proceso de BI

En la Figura 3 se presenta el proceso de Business Intelligence (BI) propuesto para el análisis de los incidentes reportados al ECU 911, en donde:

1. Se seleccionan las preguntas del negocio.
2. Se identifica los indicadores y perspectivas.
3. Se declara el grano.
4. Se identifica las dimensiones y atributos.
5. Se crea el esquema dimensional.
6. Se realiza el proceso ETL.
7. Se crea el cubo OLAP.

8. Se visualiza los datos mediante gráficos y reportes.

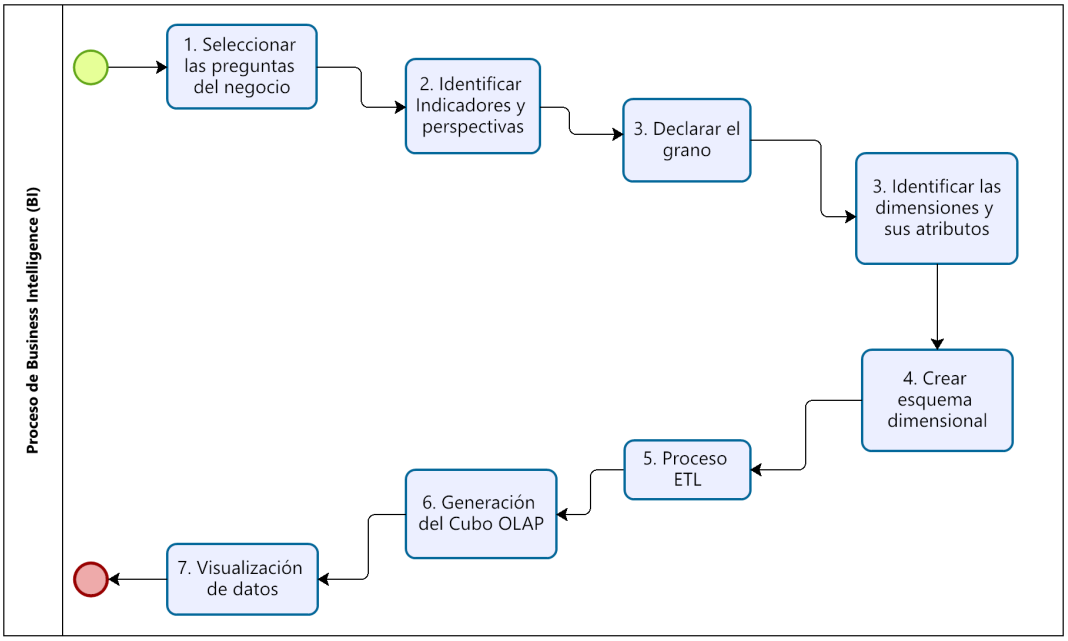


Figura 3. Proceso de Business Intelligence (BI) para el análisis de incidentes reportados al ECU 911.

3.2 Análisis de herramientas de desarrollo

En este apartado se presenta el análisis y selección de las herramientas de desarrollo que se utilizarán para la implementación del proyecto.

3.2.1 Análisis y selección de la metodología de desarrollo

Con el objetivo de seleccionar la metodología ágil más adecuada para el desarrollo del proyecto, se realizó un análisis comparativo entre las metodologías ágiles tomando en cuenta criterios tales como tamaño del proyecto, tamaño del equipo, enfoque, roles, ciclo de vida, ciclo de entrega, entre otros, los cuales se detallan en la Tabla 6, en base a la información proporcionada por [24].

Tabla 6. Análisis y comparación entre metodologías de desarrollo ágil

Criterio	XP	Kanban	Scrum
Tamaño del proyecto	Pequeños y medianos.	Pequeños, medianos y grandes.	Proyectos de toda magnitud y dificultad.
Tamaño del equipo	5 o menos integrantes.	De 5 a 14 integrantes.	De 5 a 9 integrantes.
Enfoque	Se centra en ofrecer un producto de alta calidad y en garantizar la satisfacción del cliente.	Se centra en la mejora continua del proyecto.	Se centra en la colaboración en equipo.
Roles	Programador, cliente, tester, coach, encargado del seguimiento.	No existen roles específicos, aunque algunos equipos pueden recibir apoyo de un coach.	Product owner, scrum master, equipo de desarrollo.
Ciclo de vida	Exploración, planificación, iteraciones, producción, monitoreo.	Inicio, del inglés To-Do, Trabajo en progreso, del inglés Doing, Listo para revisión, del inglés Done.	Sprint, sprint panning, daily scrum, sprint review, sprint retrospective.
Continua en la siguiente página			

Tabla 6 – continuación de la página anterior

Criterio	XP	Kanban	Scrum
Ciclo de entrega	Se centra en hacer entregas regulares y de menor escala conforme se desarrollan nuevas funcionalidades.	Las tareas progresan a lo largo del tablero Kanban a medida que avanzan en el proceso de desarrollo, y una vez terminadas, se entregan directamente al cliente.	Al final de cada iteración o sprint, se realiza una revisión del trabajo y se entrega el avance del producto al cliente o usuario.
Gestión de cambios	Se caracteriza por su capacidad de adaptarse a los cambios. En XP, los cambios son recibidos en todo momento debido a su enfoque en la retroalimentación constante.	Los cambios pueden ser gestionados a medida que son detectados y priorizados.	Si se detecta un cambio que requiere atención, se cancela el Sprint en curso y se inicia uno nuevo con las adaptaciones necesarias.

Continua en la siguiente página

Tabla 6 – continuación de la página anterior

Criterio	XP	Kanban	Scrum
Enfoque en la calidad	Las pruebas unitarias, la integración continua y la refactorización son pilares esenciales en XP que contribuyen a mantener la calidad del software. Planificación y modelado	La identificación temprana de problemas y su resolución inmediata son fundamentales para mantener la calidad. Cualquier defecto identificado es priorizado y se corregido de inmediato. Rápida y flexible	Durante un Sprint, el equipo se dedica a implementar y probar las funcionalidades según lo planificado. Cada funcionalidad debe satisfacer los criterios de aceptación establecidos.

3.2.2 Análisis y selección de herramientas de desarrollo

Mediante el análisis realizo en la Tabla 6, se optó por emplear la metodología ágil xp debido a que se centra en ofrecer un producto de alta calidad, destacando su capacidad de adaptarse a los cambios y su enfoque en la retroalimentación constante. Además, XP es adecuada para proyectos de pequeño y mediano tamaño, lo cual se ajusta a las características del proyecto actual.

3.2.3 Análisis y selección del framework de desarrollo para el servidor web (Backend)

Como lenguaje de desarrollo para el servidor web (Backend) se optó por utilizar Typescript junto al entorno de ejecución de Node.js, debido que este permite el

desarrollo de aplicaciones escalables y de alto rendimiento, impulsado por eventos asíncronos, además del amplio soporte por parte de la comunidad y documentación disponible, así como a la facilidad de uso y la gran cantidad de librerías y frameworks disponibles para el desarrollo de aplicaciones web [25].

Como framework de desarrollo para el servidor web (backend), se optó por el uso de NestJS debido a que proporciona una arquitectura modular y escalable basada en el patrón de diseño de inyección de dependencias. Además, combina elementos de la programación orientada a objetos (POO), programación funcional (PF) y programación funcional reactiva (PFR). NestJS permite utilizar como base dos de los frameworks más populares en el desarrollo web, Express y Fastify, mediante un nivel de abstracción superior que permite exponer las APIs de ambos frameworks de forma directa al desarrollador, lo que proporciona una mayor flexibilidad al momento de incluir paquetes de terceros. Además, destaca por la gran cantidad de módulos y librerías que posee [26].

En la tabla 7 se presenta una comparación entre Express y Fastify tomando en cuenta criterios como el soporte para Typescript, rendimiento, velocidad, documentación, soporte por la comunidad y paquetes/librerías.

Tabla 7. Análisis y comparación entre los frameworks de NodeJS Express y Fastify [27] [28]

Criterio	**Express.js**	**Fastify.js**
Soporte para Typescript	Mediante un paquete externo	Hecho en TypeScript con soporte directo
Rendimiento	11080 peticiones/s	45871 peticiones
Velocidad	Más lento debido a su mayor cantidad de middleware y flexibilidad	Significativamente más rápido debido a su enfoque en la velocidad y eficiencia
Documentación	Documentación detallada y ampliamente utilizada	Documentación completa y fácil de entender
Continua en la siguiente página		

Tabla 7 – continuación de la página anterior

Criterio	Express.js	Fastify.js
Soporte por la comunidad	Gran soporte por parte de la comunidad al ser el framework más usado	Cuenta con un soporte amplio por parte de su comunidad en creciente aumento
Paquetes/librerías	Permite integrar fácilmente tanto librerías propias como de terceros gracias a su arquitectura flexible y su amplia comunidad.	Tiene gran soporte para incorporar librerías personalizadas y de terceros gracias a su creciente popularidad

Considerando el análisis presentado en la Tabla 7, se optó por utilizar Fastify como el framework base para NestJS. Esto se debe a que Fastify es más rápido que Express, gracias a su enfoque en la velocidad y la eficiencia. Fastify ha sido diseñado específicamente con el objetivo de ser rápido y eficiente, lo que lo convierte en una excelente elección para aplicaciones que necesitan un alto rendimiento.

3.2.4 Análisis y selección del framework de desarrollo para el cliente web (Frontend)

Dado que en el lado del servidor se ha optado por utilizar TypeScript, para el cliente web (frontend) se ha optado por emplear el mismo lenguaje de programación. Esto se debe a que permite una integración más fluida entre el cliente y el servidor, además de facilitar la comunicación entre ambos. Para el desarrollo del cliente web (frontend), se han considerado tres de los frameworks más populares en la actualidad: Angular, ReactJS y VueJS. A continuación, se presenta una comparación de estos frameworks en la Tabla 8, en base a la información proporcionada por [24] y [29].

Tabla 8. Análisis y comparación entre los frameworks Angular, ReactJS, VueJS [30]

Criterio	Angular	ReactJS	VueJS
Arquitectura	MVC (Modelo Vista Controlador).	Basada en componentes.	MVVM (Modelo Vista Vista Modelo).
Estadística de uso	17.46% de uso a nivel mundial.	40.58% de uso a nivel mundial.	16.38% de uso a nivel mundial.
Creadores	Google.	Evan You.	Facebook.
Reutilización de componentes	Se centra en la encapsulación y la inyección de dependencias para promover la reutilización.	Se enfoca en la creación de componentes y utiliza High Order Components (HOC), Hooks y providers para reutilizar lógica entre los componentes.	Se enfoca en la creación de componentes y emplea slots y mixins para mejorar la flexibilidad y promover la reutilización eficiente del código.

Continua en la siguiente página

Tabla 8 – continuación de la página anterior

Criterio	Angular	ReactJS	VueJS
Curva de Aprendizaje	Presenta una curva de aprendizaje alta debido a sus conceptos avanzados. No obstante, la documentación oficial y los recursos de la comunidad pueden ser de gran ayuda para reducir esta curva.	Su curva de aprendizaje es baja debido a que utiliza JSX lo cual lo vuelve intuitivo y fácil de usar rápidamente.	Es fácil de aprender dado que utiliza solo tecnologías básicas del lado del cliente, como HTML, CSS y JavaScript.
Lenguajes utilizados	TypeScript.	JSX (JavaScript) y TSX (TypeScript).	JavaScript y TypeScript.

Continua en la siguiente página

Tabla 8 – continuación de la página anterior

Criterio	Angular	ReactJS	VueJS
Rendimiento	Angular cuenta con un sistema eficiente de detección de cambios que facilita la actualización del DOM.	React utiliza una representación virtual del DOM para comparar cambios antes de actualizar el DOM del navegador, en lugar de realizar modificaciones directas.	VueJS actualiza selectivamente las partes del DOM que han cambiado, lo que ayuda a optimizar la velocidad de carga de la aplicación.
Flexibilidad	Establece una estructura definida para las aplicaciones, lo cual puede resultar beneficioso especialmente en proyectos de gran escala.	Es muy adaptable y facilita la integración con otras bibliotecas y frameworks de manera sencilla.	Es flexible y fácil de integrar en proyectos existentes.

Tomando en cuenta el análisis de las características de cada framework, a continuación se presenta una comparación de las ventajas y desventajas de Angular, ReactJS y VueJS, en base a la información proporcionada por [31].

1. **ReactJS**

   (a) **Ventajas**

      i. Emplea un DOM virtual para lograr una eficiencia máxima al actualizar nodos según sea necesario.

ii. La capacidad de renderizar en el servidor es otra ventaja que este framework ofrece, especialmente adecuada para ciertos tipos de implementaciones, como las aplicaciones enfocadas en el contenido.

iii. Reduce la carga de recursos del usuario mediante el respaldo de bundling y tree shaking.

iv. La programación funcional facilita la creación de código que puede ser reutilizado.

v. Ofrece ventajas en términos de SEO en comparación con Angular y Vue.js.

(b) **Desventajas**

i. Es necesario importar bibliotecas adicionales para manejar el estado y el modelo, ya que React no incluye la arquitectura MVC de forma nativa.

ii. Aunque React permite su uso, se distancia de los enfoques basados en clases y puede presentar dificultades para aquellos que prefieren la Programación Orientada a Objetos (POO).

2. **Angular**

(a) **Ventajas**

i. Utiliza el patrón MVVM (Modelo-Vista-Modelo de Vista), el cual permite manipular la misma colección de datos de forma independiente dentro de una misma aplicación.

ii. Su estructura y arquitectura están diseñadas específicamente para mejorar la escalabilidad de los proyectos.

iii. La inyección de dependencias en los componentes ayuda a mejorar la modularidad de la aplicación.

iv. La programación funcional facilita la creación de código que puede ser reutilizado.

(b) **Desventajas**

i. Posee múltiples estructuras como Inyectables, Componentes,

Tuberías, Módulos, entre otros, que suelen presentar un mayor nivel de complejidad para su comprensión.

ii. Experimenta actualizaciones continuas, incorporando mejoras nuevas y significativas de manera constante. Sin embargo, estas actualizaciones pueden plantear desafíos al adaptarse a los cambios.

3. **VueJS**

(a) **Ventajas**

i. Facilita la creación de modelos modulares de gran alcance que pueden renderizarse eficientemente gracias a su estructura fundamental, sin requerir esfuerzos adicionales.

ii. Posee una alta capacidad de respuesta y ofrece una sencilla vinculación de datos entre el código HTML y JavaScript.

iii. Vue gestiona de forma sobresaliente la vinculación de datos bidireccional dinámica. Además, lleva a cabo la manipulación del DOM de manera coherente, lo que lo hace ideal para diversas aplicaciones.

(b) **Desventajas**

i. Su participación en el mercado es moderada, lo que indica que el intercambio de información en esta plataforma se encuentra en sus primeras fases de desarrollo.

ii. Existe el riesgo de que su flexibilidad pueda ser un problema al integrarse en proyectos extensos debido a la carencia de recursos disponibles.

Tomando en cuenta el análisis de las características de cada framework, así como sus ventajas y desventajas, se optó por el uso de ReactJS para el desarrollo del cliente web (frontend). Esto se debe a que ReactJS es una de las librerías más populares y ampliamente utilizadas en la actualidad, lo que significa que cuenta con una gran comunidad de desarrolladores y una amplia variedad de recursos disponibles. Además, ReactJS posee un gran rendimiento y eficiencia gracias a su manejo eficiente del

DOM virtual, lo que lo convierte en una excelente elección para aplicaciones web que requieren una interfaz de usuario rápida y receptiva, así como su capacidad de crear componentes reutilizables, lo que facilita el desarrollo de aplicaciones web complejas y escalables.

3.2.5 Análisis y selección del framework de desarrollo para el cliente móvil (Frontend)

Para el desarrollo del cliente móvil (frontend), se han considerado tres de los frameworks más populares en la actualidad: React Native, Flutter y Ionic. A continuación, se presenta una comparación de estos frameworks en la Tabla 9.

Tabla 9. Análisis y comparación entre los frameworks React Native, Flutter y Xamarin [32] [33]

Criterio	React Native	Flutter	Xamarin
Lanzamiento	2015	2017	2013
Licencia	MIT Licensed	BSD	MIT Licensed
Lenguaje	JavaScript, TypeScript	Dart	C#
Plataformas soportadas	Android, IOS	Android, IOS	Android, IOS
Open source	Si	Si	Si
Paradigma	Declarativo	Declarativo	Imperativo
Recarga en tiempo real	Si	Si	No
Recarga en vivo	Si	Si	No
Administrador de paquetes	NPM, Yarn	Pub	NuGet
Enfoque multiplataforma	Interpretado	Compilado a nativo	Compilado a nativo

Continua en la siguiente página

Tabla 9 – continuación de la página anterior

Criterio	React Native	Flutter	Xamarin
Compatibilidad con 64bits	No en android	Si	Si
Portabilidad	Si	No	No
Geolocalización	Incluido	Mediante paquete de la comunidad	Incluido
Notificaciones	Mediante paquete de la comunidad	Mediante paquete de la comunidad	Mediante paquete de la comunidad
Rendimiento	Alto desempeño al ser un framework ligero	Elevada eficiencia gracias a su propio motor de renderizado	Un desempeño sólido, aunque podría beneficiarse de una capa extra de abstracción.
Integración con APIs y Bibliotecas	Gran cantidad de paquetes de terceros	Una amplia disponibilidad de bibliotecas y paquetes de terceros gracias a su comunidad en constante desarrollo	Gran cantidad de bibliotecas y recursos disponibles, especialmente diseñados para integrarse con los servicios de Microsoft.
Documentación	Amplia documentación	Documentación amplia y detallada, además de gran cantidad de recursos de terceros	Documentación completa y abundante

Continua en la siguiente página

Tabla 9 – continuación de la página anterior

Criterio	React Native	Flutter	Xamarin
Tiempo de desarrollo	Rápida gracias a su hot reload con cambios en tiempo real	Rápido gracias al hot reload que permite visualizar cambios en tiempo real	Tiempo de desarrollo más lento debido a la compilación

Considerando el análisis realizado en la Tabla 9, se optó por utilizar Flutter para el desarrollo del cliente móvil (frontend). Esto se debe a que Flutter es un framework de desarrollo de aplicaciones móviles multiplataforma creado por Google, que permite el desarrollo de aplicaciones móviles nativas de alta calidad para Android e iOS desde una sola base de código. Flutter destaca por su rendimiento y eficiencia, gracias a su propio motor de renderizado, lo que lo convierte en una excelente elección para aplicaciones móviles que requieren un alto rendimiento y una interfaz de usuario rápida y receptiva. Además, Flutter cuenta con una amplia variedad de widgets y herramientas que facilitan el desarrollo de aplicaciones móviles complejas y escalables.

3.2.6 Análisis y selección de la base de datos

Para la base de datos del proyecto se han considerado dos opciones: MySQL y PostgreSQL. A continuación, se presenta una comparación de estas bases de datos en la Tabla 10.

Tabla 10. Análisis y comparación entre las bases de datos MySQL y PostgreSQL [33]

Criterio	MySQL	PostgreSQL
GUI	MySQL Workbench	pgAdmin
Consumo de recursos	Consumo mayor de CPU y memoria	Consumo mayor de CPU y memoria
Continua en la siguiente página		

Tabla 10 – continuación de la página anterior

Criterio	MySQL	PostgreSQL
Tiempo de respuesta de CRUD	Bajo	Alto
Lenguaje de ejecución	C/C++	C
herencia de tablas	No	Si
Tipos de datos	Solo tipos estándar	Estándar, store, arreglos, geográficos, definido por el usuario, etc.
APIs y otros métodos de acceso	ADO.NET, JDBC, biblioteca C nativa, ODBC, API de transmisión para objetos grandes	ADO.NET, JDBC, ODBC, API nativa patentada
Tipos de conexión	Las conexiones son subprocesos del sistema operativo	Las conexiones son subprocesos del sistema operativo
Respaldo	En MySQL, Mysqldump y XtraBackup, las herramientas proporcionan respaldo	PostgreSQL proporciona una copia de seguridad completa en línea.
Lenguajes soportados	C/C++, PHP, Java, Go, Delphi, Lisp, Erlang, Node.js, R, Perl, PHP	Go, C/C++, Java, Delphi, Javascript, Erlang, Lisp, R, .Net, Tcl, Python
Sistemas Operativos Soportados	Windows, Linux, macOS, Oracle Solaris, Fedora, FreeBSD	Windows, macOS, BSD, Linux, Solaris

Tomando en cuenta el análisis presentado en la Tabla 10, se optó por utilizar PostgreSQL como la base de datos del proyecto. Esto se debe a que PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto y de alto rendimiento, que ofrece una amplia variedad de características y funcionalidades avanzadas, como

soporte para tipos de datos geoespaciales, herencia de tablas, tipos de datos definidos por el usuario, entre otros. Además, PostgreSQL cuenta con una arquitectura robusta y escalable, que permite gestionar grandes volúmenes de datos de forma eficiente y fiable, lo que lo convierte en una excelente elección para aplicaciones que requieren una base de datos potente y confiable.

3.3 Análisis de la precisión de los datos georreferenciados

Con la finalidad de evaluar la precisión de los datos georreferenciados obtenidos a través de la aplicación móvil, se desarrolló un prototipo de aplicación que permite obtener la ubicación del dispositivo móvil y compararla con un punto de referencia conocido, el cual será seleccionado en la aplicación de prueba. Para ello, se utilizó el GPS del dispositivo móvil para obtener la ubicación, el framework de desarrollo Flutter para la creación de la aplicación móvil y Google Maps para la visualización de la ubicación en un mapa interactivo.

En la Figura 4 se muestra el prototipo de la aplicación móvil desarrollada para la evaluación de la precisión de los datos georreferenciados. Esta aplicación cuenta con un botón que permite obtener la ubicación actual del dispositivo móvil. Con esto, el usuario puede seleccionar un punto de referencia en el mapa y guardar ambos puntos para su posterior comparación.

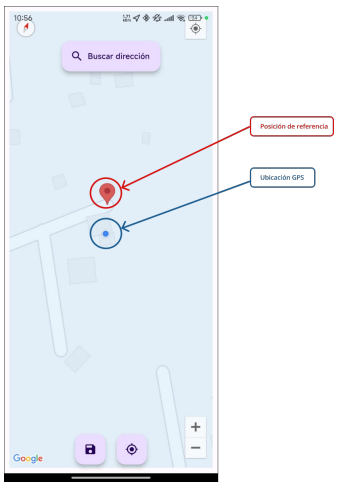


Figura 4. Prototipo de la aplicación móvil para la evaluación de la precisión de los datos georreferenciados.

Para establecer la cantidad de datos necesarios para la evaluación de la precisión de los datos georreferenciados, se optó por utilizar una muestra infinita con población desconocida, aplicando la Ecuación 1, donde:

- n = tamaño de la muestra
- Z = nivel de confianza
- p = probabilidad de éxito o proporción esperada
- q = probabilidad de fracaso
- pq = varianza de la población
- e = error de estimación máximo aceptable

$$n = \frac{Z^2 \cdot p \cdot q}{e^2} \tag{1}$$

Sustituyendo los valores en la Ecuación 1, se obtiene:

- Z = 1.96, con un nivel de confianza del 95% y un error de estimación máximo aceptable del 5%
- p = 0.50
- q = 0.50
- e = 0.05

$$n = \frac{(1.96)^2 \cdot 0.50 \cdot 0.50}{(0.05)^2} \tag{2}$$

$$n = 384.16 \approx 385 \tag{3}$$

Utilizando un nivel de confianza del 95% y un error de estimación máximo aceptable del 5%, se obtiene una muestra aproximada de 385 puntos de referencia para la evaluación de la precisión de los datos georreferenciados.

Los datos obtenidos de la aplicación móvil se almacenaron en una tabla de una base de datos PostgreSQL, la cual se muestra en la Figura 5. Esta tabla contiene la información de los puntos de referencia seleccionados en la aplicación móvil, los cuales son: latitud y longitud real (Punto de referencia) y latitud y longitud obtenida mediante GPS (Punto de ubicación).

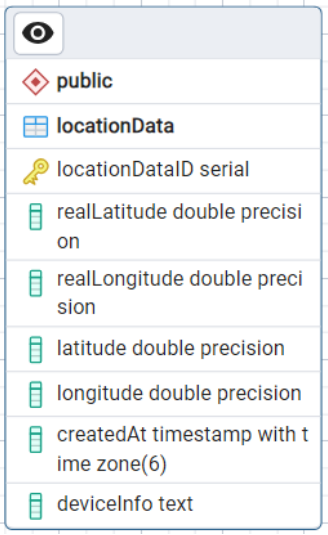


Figura 5. Tabla de datos georreferenciados almacenados en la base de datos PostgreSQL.

Estos datos se traspasaron a un archivo de Excel para su posterior análisis. Para ello, se utilizó un script con Node.js y la librería de manejo de archivos ExcelJS, como se muestra en el Anexo C. La distancia entre los puntos de referencia y los puntos de ubicación se calculó utilizando la fórmula de Haversine, dado que esta toma en cuenta la curvatura de la Tierra y proporciona una mejor precisión [34]. La función de Haversine utilizada se muestra en el Anexo D.

El archivo de Excel generado con los datos georreferenciados se muestra en la Figura 6. El cual contiene la longitud obtenida mediante GPS y la distancia entre la ubicación del dispositivo y el punto de referencia seleccionado.

	A	B	C	D	E	F
1	#	Latitude	Longitude	Haversine distance from center (m)		
2	1	-1.270393597	-78.63184031	6.11		
3	2	-1.328198102	-78.62853147	1.27		
4	3	-1.327951405	-78.62848185	11.29		
5	4	-1.328555076	-78.62942465	15.37		
6	5	-1.329033386	-78.62912256	7.44		
7	6	-1.328822219	-78.62834942	11		
8	7	-1.329500635	-78.62806946	2.54		
9	8	-1.330691885	-78.62756487	3.38		
10	9	-1.330926851	-78.62835713	5.81		
11	10	-1.330667752	-78.62745121	2.77		
12	11	-1.330075479	-78.62705525	13.92		
13	12	-1.330233016	-78.62766981	8.36		
14	13	-1.329764091	-78.62876885	5.58		
15	14	-1.329489909	-78.62798732	16.31		
16	15	-1.329191258	-78.62730436	6.42		
17	16	-1.33049312	-78.62693857	15.73		
18	17	-1.330498483	-78.62687789	16.34		
19	18	-1.329981292	-78.62700596	4.6		
20	19	-1.349407578	-78.61801419	9.2		
21	20	-1.352097086	-78.61803263	4.86		
22	21	-1.352759743	-78.61813355	5.57		
23	22	-1.353402623	-78.61854795	4.65		
24	23	-1.353493793	-78.61809433	14.87		
25	24	-1.355800853	-78.61772519	9.51		
26	25	-1.355151605	-78.61595593	5.03		
27	26	-1.355753593	-78.61568369	9.4		

Figura 6. Archivo de Excel con los datos georreferenciados para la comparación.

Utilizando estos datos se creó un cluster de puntos georreferenciados en python, para ello se utilizó el algoritmo de K-Means, el cual permite agrupar los puntos en clusters en función de su distancia. El código utilizado para la creación del cluster se muestra en el Anexo E. Utilizando el método del codo se determinó el número óptimo de clusters, el cual fue de 2 como se muestra en la Figura 7 en done se puede observar que el codo se encuentra en el punto 2.

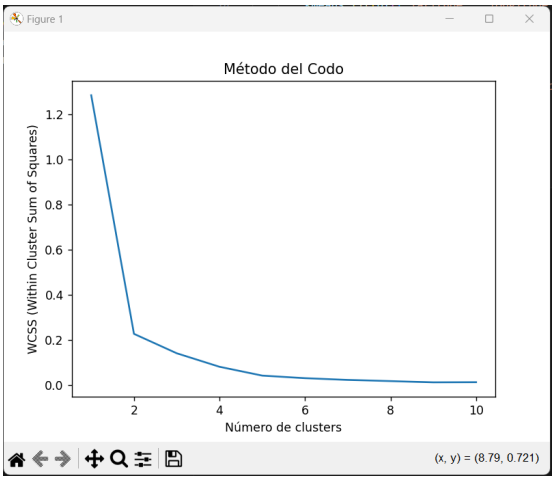


Figura 7. Método del codo para determinar el número óptimo de clusters.

El cluster resultante aplicando el algoritmo de K-Means se muestra en la Figura 8.

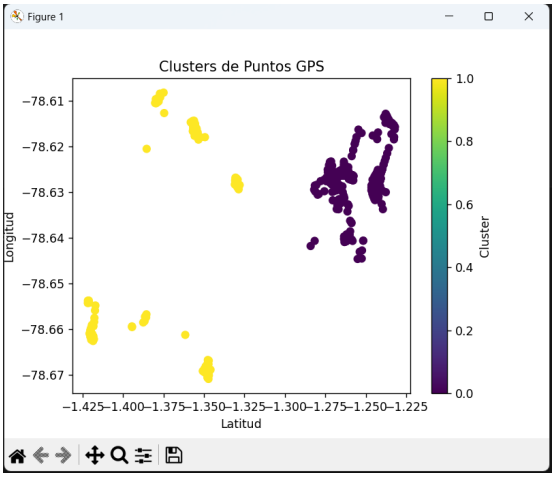


Figura 8. Cluster de puntos georreferenciados obtenidos mediante el algoritmo de K-Means.

Posteriormente se calculó la distancia media de cada cluster, así como la media de las distancias medias de los clusters. Los resultados se muestran a continuación:

- Distancias medias de cada cluster:
  - Cluster 1: 6.32 m
  - Cluster 2: 8.78 m
- Media de las distancias medias de los clusters: 7.55 m

Estos resultados indican que la precisión media de los cluster es de 7.55 metros. Con la finalidad de obtener el intervalo de confianza para esta media, se utilizó la Ecuación 4, donde:

- IC = Intervalo de confianza
- M = Media de las distancias medias de los clusters
- ME = Margen de error

$$IC = M \pm ME \tag{4}$$

El margen de error se calculó utilizando la Ecuación 5, donde:

- ME = Margen de error

- Z = Valor crítico de la distribución normal estándar
- SE = Error estándar

$$ME = Z \times SE \tag{5}$$

El error estándar se calculó utilizando la Ecuación 6, donde:

- SE = Error estándar
- σ = Desviación estándar
- k = Tamaño de la muestra

$$SE = \frac{\sigma}{\sqrt{k}} \tag{6}$$

Sustituyendo los valores en las Ecuaciones 6 y 5, se obtiene:

- Z = 1.96, con un nivel de confianza del 95%
- σ = 4.77
- k = 385

$$SE = \frac{4.77}{\sqrt{385}} = 0.24 \tag{7}$$

$$ME = 1.96 \times 0.24 = 0.47 \tag{8}$$

Sustituyendo los valores en la Ecuación 4 para el intervalo de confianza, se obtiene:

- ME = 0.47
- M = 7.55

$$IC = 7.55 \pm 0.47 = [7.08, 8.02] \tag{9}$$

Por lo tanto, se tiene que para un nivel de confianza del 95%, el error de estimación de la precisión de los datos georreferenciados se encuentra entre 7.08 y 8.02 metros con respecto al punto real de la ubicación.

3.4 Desarrollo de la propuesta

En base a la metodología XP, es esencial seguir una serie de fases que incluyen la planificación, el diseño, la codificación y la prueba de la aplicación y lanzamiento. A continuación, se describen cada una de las fases de la metodología XP:

- **Planificación:** En esta fase, se definen las historias de usuario, definición de roles, tareas, estimación de tiempo y el plan de entregas.
- **Diseño:** En esta fase, se realizan las iteraciones de diseño y se crean las tarjetas CRC (Clase Responsabilidad Colaboración).
- **Codificación:** En esta fase, se realiza la codificación en base a las historias de usuario.
- **Pruebas:** En esta fase, se aplican las pruebas de aceptación de acuerdo a las historias de usuario.
- **Lanzamiento:** En esta fase, finalizadas las pruebas, se realiza la puesta en producción del sistema y se realiza la entrega al cliente

3.4.1 Fase I: Planificación

En esta fase el cliente, el equipo de desarrollo y los gerentes acuerdan el orden de implementación de las historias de usuario y se establecen las entregas de las mismas.

a. Grupos de interés (Stakeholders)

- **Usuarios del aplicativo web:** Usuarios administradores.
- **Usuarios del aplicativo móvil:** Ciudadanos, policías.
- **Desarrollador:** Autor del presente proyecto.

b. Roles del proyecto

La distribución de roles es fundamental para una organización eficaz entre los miembros del equipo y el cliente. En este proyecto, se han establecido los siguientes roles en base a la metodología elegida (XP).

Requerimientos de implementación del aplicativo web.

En la tabla 11 se presentan los requerimientos definidos para el desarrollo del proyecto.

Tabla 11. Definición de requerimientos

N°	Historia	Descripción	Prioridad
Aplicación web			
1	Iniciar sesión	El inicio de sesión se realizará a través de la autenticación del usuario, quien deberá ingresar su correo y contraseña.	Alta
2	Cerrar sesión	El usuario podrá cerrar la sesión en cualquier momento.	Alta
3	Gestionar usuarios	El administrador podrá crear, visualizar, actualizar y deshabilitar usuarios	Alta
4	Gestionar tipos de incidentes	El administrador podrá crear, visualizar, actualizar y deshabilitar tipos de incidentes	Alta
5	Gestionar zonas de vigilancia	El administrador podrá crear, visualizar, actualizar y deshabilitar tipos de incidentes	Alta
6	Asignar policías a las zonas de vigilancia	El administrador podrá asignar y desasignar miembros de la policías a las zonas de vigilancia	Alta
Continua en la siguiente página			

Tabla 11 – continuación de la página anterior

N°	Historia	Descripción	Prioridad
7	Gestionar alertas de incidentes	El administrador podrá visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real	Alta
8	Visualizar mapa de calor	El administrador podrá visualizar mediante un mapa de calor los incidentes delictivos suscitados en las diferentes zonas	Alta
Aplicación móvil			
9	Iniciar sesión	El inicio de sesión se realizará a través de la autenticación del usuario, quien deberá ingresar su correo y contraseña.	Alta
10	Cerrar sesión	El usuario podrá cerrar la sesión en cualquier momento.	Alta
11	Registro de usuario	Los usuarios deberán ingresar sus datos y fotografía mediante un formulario.	Alta
12	Cambiar contraseña	El usuario podrá cambiar su contraseña en cualquier momento.	Media
13	Recuperar contraseña	El usuario podrá recuperar su contraseña en caso de olvidarla mediante su correo electrónico.	Media
14	Asignar miembros al grupo familiar	El usuario podrá Asignar miembros al grupo familiar mediante la cédula de ciudadanía.	Alta

Continua en la siguiente página

Tabla 11 – continuación de la página anterior

N°	Historia	Descripción	Prioridad
15	Enviar alertas de emergencia	El usuario podrá enviar alertas de emergencia seleccionando el tipo de incidente y oprimiendo un botón de pánico durante 3 segundos.	Alta
16	Visualizar alertas de emergencia de familiares	El usuario podrá visualizar mediante un mapa las alertas de emergencia enviadas por los sus familiares así como su posición en tiempo real.	Alta
17	Visualizar alertas de emergencia de los ciudadanos	El usuario policía podrá visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real	Alta

Las historias de usuario en base a su prioridad y las necesidades del proyecto, serán implementadas en el siguiente orden:

Tabla 12. Definición de requerimientos

N°	Historia	Descripción	Prioridad
1	Iniciar sesión sistema web	El inicio de sesión se realizará a través de la autenticación del usuario, quien deberá ingresar su correo y contraseña.	Alta
2	Cerrar sesión sistema web	El usuario podrá cerrar la sesión en cualquier momento.	Alta
3	Gestionar usuarios	El administrador podrá crear, visualizar, actualizar y deshabilitar usuarios	Alta
Continua en la siguiente página			

Tabla 12 – continuación de la página anterior

N°	Historia	Descripción	Prioridad
4	Gestionar tipos de incidentes	El administrador podrá crear, visualizar, actualizar y deshabilitar tipos de incidentes	Alta
5	Gestionar zonas de vigilancia	El administrador podrá crear, visualizar, actualizar y deshabilitar tipos de incidentes	Alta
6	Asignar policías a las zonas de vigilancia	El administrador podrá asignar y desasignar miembros de la policías a las zonas de vigilancia	Alta
7	Gestionar alertas de incidentes	El administrador podrá visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real	Alta
8	Visualizar mapa de calor	El administrador podrá visualizar mediante un mapa de calor los incidentes delictivos suscitados en las diferentes zonas	Alta
9	Iniciar sesión aplicación móvil	El inicio de sesión se realizará a través de la autenticación del usuario, quien deberá ingresar su correo y contraseña.	Alta
10	Cerrar sesión aplicación móvil	El usuario podrá cerrar la sesión en cualquier momento.	Alta
11	Registro de usuario	Los usuarios deberán ingresar sus datos y fotografía mediante un formulario.	Alta

Continua en la siguiente página

Tabla 12 – continuación de la página anterior

N°	Historia	Descripción	Prioridad
12	Asignar miembros al grupo familiar	El usuario podrá Asignar miembros al grupo familiar mediante la cédula de ciudadanía.	Alta
13	Enviar alertas de emergencia	El usuario podrá enviar alertas de emergencia seleccionando el tipo de incidente y oprimiendo un botón de pánico durante 3 segundos.	Alta
14	Visualizar alertas de emergencia de familiares	El usuario podrá visualizar mediante un mapa las alertas de emergencia enviadas por los sus familiares así como su posición en tiempo real.	Alta
15	Visualizar alertas de emergencia de los ciudadanos	El usuario policía podrá visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real	Alta
16	Cambiar contraseña	El usuario podrá cambiar su contraseña en cualquier momento.	Media
17	Recuperar contraseña	El usuario podrá recuperar su contraseña en caso de olvidarla mediante su correo electrónico.	Media
18	Desarrollar el modelo analítico de BI	El administrador podrá visualizar informes detallados sobre los incidentes ocurridos, generados mediante BI.	Media

Para detallar los requisitos funcionales del sistema se empleará un modelo de historia de usuario, el cual se presenta en la tabla 13.

Tabla 13. Modelo de historia de usuario

Historia de Usuario	
Número <Identificador de la historia>	**Usuario** <Persona que proporciona la información>
Nombre de la historia <Título de la historia de usuario>	
Prioridad en negocio <Grado de la necesidad de negocio (Alta, Media, Baja)>	**Riesgo en desarrollo** <Grado del impacto en caso de fallo (Alta, Media, Baja)>
Puntos estimados <Días estimados designados para el desarrollo>	**Iteración asignada** <Número de la iteración de la historia>
Programador responsable <Persona encargada del desarrollo de la historia>	
Descripción: <Descripción de la historia>	
Criterios de aceptación	
<Se describe las condiciones necesarias que se ejecute cada historia de usuario>	

Tabla 14. Historia de usuario 1: Iniciar sesión sistema web

Historia de Usuario	
Número: 1	**Usuario:** Usuario administrador
Nombre de la historia: Iniciar sesión sistema web	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 8	**Iteración asignada:** 1
Programador responsable: José Pazmiño	
Descripción: Como usuario administrador del sistema web, quiero iniciar sesión en el sistema para acceder a sus funcionalidades	
Criterios de aceptación	
Continua en la siguiente página	

Criterio 1: Dado que el usuario administrador ingrese su correo y contraseña de forma correcta cuando presione el botón de inicio de sesión, el sistema deberá redirigirlo a la página principal del sistema. **Criterio 2:** Dado que el usuario administrador ingrese su correo y contraseña de forma incorrecta cuando presione el botón de inicio de sesión, el sistema deberá mostrar un mensaje de error.

Tabla 15. Historia de usuario 2: Cerrar sesión sistema web

Historia de Usuario	
Número: 2	**Usuario:** Usuario administrador
Nombre de la historia: Cerrar sesión sistema web	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 2	**Iteración asignada:** 1
Programador responsable: José Pazmiño	
Descripción: Como usuario del sistema web, quiero cerrar sesión en el sistema para finalizar mi sesión de trabajo	
Criterios de aceptación	
Criterio 1: Dado que el usuario administrador presione el botón de cerrar sesión desde cualquier página, el sistema deberá redirigirlo a la página de inicio de sesión.	

Tabla 16. Historia de usuario 3: Gestionar usuarios

Historia de Usuario	
Número: 3	**Usuario:** Usuario administrador
Nombre de la historia: Gestionar usuarios	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Continua en la siguiente página	

Puntos estimados:en desarrollo: 18	**Iteración asignada:** 1
Programador responsable: José Pazmiño	
Descripción: Como usuario administrador, quiero administrar los usuarios que se encuentran registrados en el sistema	
Criterios de aceptación	
Criterio 1: El usuario administrador debe poder visualizar, crear, actualizar y deshabilitar y habilitar usuarios en el sistema.	

Tabla 17. Historia de usuario 4: Gestionar tipos de incidentes

Historia de Usuario	
Número: 4	**Usuario:** Usuario administrador
Nombre de la historia: Gestionar tipos de incidentes	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 13	**Iteración asignada:** 1
Programador responsable: José Pazmiño	
Descripción: Como usuario administrador, quiero gestionar los tipos de incidentes para definir las categorías de los incidentes	
Criterios de aceptación	
Criterio 1: El usuario administrador debe poder visualizar, crear, actualizar y deshabilitar y habilitar tipos de incidentes en el sistema.	

Tabla 18. Historia de usuario 5: Gestionar zonas de vigilancia

Historia de Usuario	
Número: 5	**Usuario:** Usuario administrador
Nombre de la historia: Gestionar zonas de vigilancia	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Continua en la siguiente página	

Puntos estimados:en desarrollo: 21	**Iteración asignada:** 2
Programador responsable: José Pazmiño	
Descripción: Como usuario administrador, quiero gestionar las zonas de vigilancia para definir las áreas de monitoreo	
Criterios de aceptación	
Criterio 1: El usuario administrador debe poder visualizar, crear, actualizar y deshabilitar y habilitar zonas de vigilancia en el sistema.	

Tabla 19. Historia de usuario 6: Asignar policías a las zonas de vigilancia

Historia de Usuario	
Número: 6	**Usuario:** Usuario administrador
Nombre de la historia: Asignar policías a las zonas de vigilancia	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 13	**Iteración asignada:** 2
Programador responsable: José Pazmiño	
Descripción: Como usuario administrador, quiero asignar y desasignar miembros de la policía a las zonas de vigilancia	
Criterios de aceptación	
Criterio 1: El usuario administrador debe poder visualizar, asignar y desasignar miembros de la policía a las zonas de vigilancia en el sistema.	

Tabla 20. Historia de usuario 7: Gestionar alertas de incidentes

Historia de Usuario	
Número: 7	**Usuario:** Usuario administrador
Nombre de la historia: Gestionar alertas de incidentes	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Continua en la siguiente página	

Puntos estimados:en desarrollo: 23	**Iteración asignada:** 2
Programador responsable: José Pazmiño	
Descripción: Como usuario administrador, quiero visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real	
Criterios de aceptación	
Criterio 1: El usuario administrador debe poder visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real.	

Tabla 21. Historia de usuario 8: Visualizar mapa de calor

Historia de Usuario	
Número: 8	**Usuario:** Usuario administrador
Nombre de la historia: Visualizar mapa de calor	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 11	**Iteración asignada:** 2
Programador responsable: José Pazmiño	
Descripción: Como usuario administrador, quiero visualizar mediante un mapa de calor los incidentes delictivos suscitados en las diferentes zonas	
Criterios de aceptación	
Criterio 1: El usuario administrador debe poder visualizar mediante un mapa de calor los incidentes delictivos suscitados en las diferentes zonas.	

Tabla 22. Historia de usuario 9: Iniciar sesión aplicación móvil

Historia de Usuario	
Número: 9	**Usuario:** Ciudadano
Continua en la siguiente página	

Nombre de la historia: Iniciar sesión sistema móvil	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 8	**Iteración asignada:** 3
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero iniciar sesión en el sistema móvil para acceder a sus funcionalidades	
Criterios de aceptación	
Criterio 1: Dado que el ciudadano ingrese su correo y contraseña de forma correcta cuando presione el botón de inicio de sesión, la aplicación deberá redirigirlo a la pantalla principal de la aplicación. **Criterio 2:** Dado que el ciudadano ingrese su correo y contraseña de forma incorrecta cuando presione el botón de inicio de sesión, la aplicación deberá mostrar un mensaje de error.	

Tabla 23. Historia de usuario 10: Cerrar sesión aplicación móvil

Historia de Usuario	
Número: 10	**Usuario:** Ciudadano
Nombre de la historia: Cerrar sesión aplicación móvil	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 4	**Iteración asignada:** 3
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero cerrar sesión en la aplicación móvil para finalizar mi sesión de trabajo	
Criterios de aceptación	
Criterio 1: Dado que el ciudadano presione el botón de cerrar sesión desde cualquier pantalla, la aplicación deberá redirigirlo a la pantalla de inicio de sesión.	

Tabla 24. Historia de usuario 11: Registro de usuario

Historia de Usuario	
Número: 11	**Usuario:** Ciudadano
Nombre de la historia: Registro de usuario	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 11	**Iteración asignada:** 3
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero ingresar mis datos y fotografía mediante un formulario	
Criterios de aceptación	
Criterio 1: El ciudadano debe poder ingresar sus datos personales y fotografía mediante un formulario.	

Tabla 25. Historia de usuario 12: Asignar miembros al grupo familiar

Historia de Usuario	
Número: 12	**Usuario:** Ciudadano
Nombre de la historia: Asignar miembros al grupo familiar	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 18	**Iteración asignada:** 4
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero asignar miembros al grupo familiar para recibir alertas de emergencia	
Criterios de aceptación	
Criterio 1: El ciudadano debe poder asignar miembros al grupo familiar para recibir alertas de emergencia. **Criterio 2:** El ciudadano debe poder visualizar los miembros asignados al grupo familiar.	

Tabla 26. Historia de usuario 13: Enviar alerta de incidente

Historia de Usuario	
Número: 13	**Usuario:** Ciudadano
Nombre de la historia: Enviar alerta de incidente	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 21	**Iteración asignada:** 4
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero enviar una alerta de incidente para notificar a las autoridades sobre un incidente delictivo	
Criterios de aceptación	
Criterio 1: El ciudadano debe poder enviar una alerta de incidente para notificar a sus familiares y entes de control sobre una emergencia.	

Tabla 27. Historia de usuario 14: Visualizar alertas de emergencia de familiares

Historia de Usuario	
Número: 14	**Usuario:** Ciudadano
Nombre de la historia: Visualizar alertas de emergencia de familiares	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 21	**Iteración asignada:** 4
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero visualizar mediante un mapa las alertas de emergencia enviadas por mis familiares así como su posición en tiempo real	
Criterios de aceptación	
Criterio 1: El ciudadano debe poder visualizar mediante un mapa las alertas de emergencia enviadas por los miembros de su grupo familiar así como su posición en tiempo real.	

Tabla 28. Historia de usuario 15: Visualizar alertas de emergencia de ciudadanos

Historia de Usuario	
Número: 15	**Usuario:** Policía
Nombre de la historia: Visualizar alertas de emergencia de ciudadanos	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 11	**Iteración asignada:** 4
Programador responsable: José Pazmiño	
Descripción: Como policía, quiero visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real y el tipo de incidente	
Criterios de aceptación	
Criterio 1: El policía debe poder visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real y el tipo de incidente.	

Tabla 29. Historia de usuario 16: Cambiar contraseña

Historia de Usuario	
Número: 16	**Usuario:** Ciudadano
Nombre de la historia: Cambiar contraseña	
Prioridad en negocio Media	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 14	**Iteración asignada:** 5
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero cambiar mi contraseña para mejorar la seguridad de mi cuenta	
Criterios de aceptación	
Criterio 1: El ciudadano debe poder cambiar su contraseña para mejorar la seguridad de su cuenta.	

Tabla 30. Historia de usuario 17: Recuperar contraseña

Historia de Usuario	
Número: 17	**Usuario:** Ciudadano
Nombre de la historia: Recuperar contraseña	
Prioridad en negocio Media	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 15	**Iteración asignada:** 5
Programador responsable: José Pazmiño	
Descripción: Como ciudadano, quiero recuperar mi contraseña en caso de olvidarla	
Criterios de aceptación	
Criterio 1: El ciudadano debe poder recuperar su contraseña en caso de olvidarla.	

Tabla 31. Historia de usuario 18: Desarrollar modelo analítico de BI

Historia de Usuario	
Número: 18	**Usuario:** Administrador
Nombre de la historia: Desarrollar modelo analítico de BI	
Prioridad en negocio Alta	**Riesgo en desarrollo:** Medio
Puntos estimados:en desarrollo: 41	**Iteración asignada:** 5
Programador responsable: José Pazmiño	
Descripción: Como administrador, quiero visualizar informes detallados sobre los incidentes ocurridos	
Criterios de aceptación	
Criterio 1: El administrador debe poder visualizar informes detallados sobre los incidentes ocurridos. **Criterio 2:** El administrador debe poder la información al realizar al click en las gráficas.	

Estimación de las historias de usuario.

Tabla 32. Estimación de tiempo para las historias de usuario

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
1	Iniciar sesión sistema web	Diseño de la base de datos	4	1	4
		Desarrollo backend para inicio de sesión	4	1	4
		Diseño de la interfaz para iniciar sesión	4	1	4
		Validación y pruebas para inicio de sesión	4	1	4
2	Cerrar sesión sistema web	Diseño de la interfaz para cerrar sesión	1	1	1
		Validación y pruebas para cerrar de sesión	1	1	1
3	Gestionar usuarios	Diseño de la base de datos para la gestión de usuarios	4	1	4

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
		Desarrollo backend para la gestión de usuarios	5	1	5
		Desarrollo de la interfaz para gestión de usuarios	5	1	5
		Validación y pruebas para gestión de usuarios	4	1	4
4	Gestionar tipos de incidentes	Diseño de la base de datos para la gestión de incidentes	3	1	3
		Desarrollo backend para la gestión de incidentes	3	1	3
		Desarrollo de la interfaz para la gestión de incidentes	4	1	4
		Validación y pruebas para la gestión de incidentes	3	1	3

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
5	Gestionar zonas de vigilancia	Diseño de la base de datos para la gestión de zonas de vigilancia	4	1	4
		Desarrollo backend para la gestión de zonas de vigilancia	5	1	5
		Desarrollo de la interfaz para la gestión de zonas de vigilancia	8	1	8
		Validación y pruebas para la gestión de zonas de vigilancia	4	1	4
6	Asignar policías a las zonas de vigilancia	Diseño de la base de datos para asignar policías a las zonas de vigilancia	3	1	3
		Desarrollo backend para asignar policías a las zonas de vigilancia	3	1	3

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
		Desarrollo de la interfaz para asignar policías a las zonas de vigilancia	4	1	4
		Validación y pruebas para asignar policías a las zonas de vigilancia	3	1	3
7	Gestionar alertas de incidentes	Diseño de la base de datos para Gestionar alertas de incidentes	5	1	5
		Desarrollo backend para Gestionar alertas de incidentes	8	1	8
		Desarrollo de la interfaz para Gestionar alertas de incidentes	6	1	6

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
		Validación y pruebas para Gestionar alertas de incidentes	4	1	4
8	Visualizar mapa de calor	Desarrollo backend para Visualizar mapa de calor	4	1	4
		Desarrollo de la interfaz para Visualizar mapa de calor	4	1	4
		Validación y pruebas para Visualizar mapa de calor	3	1	3
9	Iniciar sesión aplicación móvil	Desarrollo de la interfaz para iniciar sesión en la aplicación móvil	5	1	5
		Validación y pruebas para iniciar sesión en la aplicación móvil	3	1	3

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
10	Cerrar sesión aplicación móvil	Diseño de la interfaz para cerrar sesión en la aplicación móvil	3	1	3
		Validación y pruebas para cerrar sesión en la aplicación móvil	1	1	1
11	Registro de usuario	Desarrollo de la interfaz para el registro de usuario en la aplicación móvil	8	1	8
		Validación y pruebas para el registro de usuario en la aplicación móvil	3	1	3
12	Asignar miembros al grupo familiar	Diseño de la base de datos para Asignar miembros al grupo familiar	4	1	4

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
		Desarrollo backend para Asignar miembros al grupo familiar	5	1	5
		Desarrollo de la interfaz para Asignar miembros al grupo familiar	6	1	6
		Validación y pruebas para Asignar miembros al grupo familiar	3	1	3
13	Enviar alertas de emergencia	Diseño de la base de datos para enviar alertas de emergencia	5	1	5
		Desarrollo backend para enviar alertas de emergencia	8	1	8
		Desarrollo de la interfaz para enviar alertas de emergencia	5	1	5

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
		Validación y pruebas para enviar alertas de emergencia	3	1	3
14	Visualizar alertas de emergencia de familiares	Diseño de la base de datos para visualizar alertas de emergencia de familiares	5	1	5
		Desarrollo backend para visualizar alertas de emergencia de familiares	6	1	6
		Desarrollo de la interfaz para visualizar alertas de emergencia de familiares	7	1	7
		Validación y pruebas para visualizar alertas de emergencia de familiares	3	1	3

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
15	Visualizar alertas de emergencia de los ciudadanos	Desarrollo backend para visualizar alertas de emergencia de los ciudadanos	3	1	3
		Desarrollo de la interfaz para visualizar alertas de emergencia de los ciudadanos	5	1	5
		Validación y pruebas para visualizar alertas de emergencia de los ciudadanos	3	1	3
16	Cambiar contraseña	Diseño de la base de datos para cambiar contraseña	3	1	3
		Desarrollo backend para cambiar contraseña	3	1	3
		Desarrollo de la interfaz para cambiar contraseña	5	1	5

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
		Validación y pruebas para cambiar contraseña	3	1	3
17	Recuperar contraseña	Diseño de la base de datos para recuperar contraseña	4	1	4
		Desarrollo backend para recuperar contraseña	4	1	4
		Desarrollo de la interfaz para recuperar contraseña	4	1	4
		Validación y pruebas para recuperar contraseña	3	1	3
18	Desarrollar el modelo analítico de BI	Seleccionar las preguntas del negocio	4	1	4
		Identificar Indicadores y perspectivas	3	1	3
		Declarar el grano	3	1	3

Continua en la siguiente página

Tabla 32 – continuación de la página anterior

N°	Historia de usuario	Tareas	Puntos estimados	Días	Horas
		Identificar las dimensiones y sus atributos	8	1	8
		Crear esquema dimensional	2	1	2
		Realizar el Proceso ETL	5	1	5
		Creación del Cubo OLAP	5	1	5
		Visualización de datos (Generar reportes)	16	2	15

Plan de entregas

Con la finalidad de llevar a cabo las actividades asignadas a cada historia de usuario, se ha establecido un cronograma que detalla las iteraciones y con su tiempo estimado.

Tabla 33. Asignación de roles en el proyecto en base a la metodología XP

N°	Historia de usuario	Tiempo estimado		Iteración asignada				
		Días	Horas	1	2	3	4	5
1	Iniciar sesión sistema web	1.00	8	X				
2	Cerrar sesión sistema web	0.25	2	X				
3	Gestionar usuarios	2.25	18	X				
4	Gestionar tipos de incidentes	1.63	13	X				
5	Gestionar zonas de vigilancia	2.63	21		X			
6	Asignar policías a las zonas de vigilancia	1.63	13		X			
7	Gestionar alertas de incidentes	2.88	23		X			
8	Visualizar mapa de calor	1.38	11		X			
9	Iniciar sesión aplicación móvil	1.00	8			X		
10	Cerrar sesión aplicación móvil	0.50	4			X		
11	Registro de usuario	1.38	11			X		
12	Asignar miembros al grupo familiar	2.25	18				X	

Continua en la siguiente página

Tabla 33 – continuación de la página anterior

N°	Historia de usuario	Tiempo estimado		Iteración asignada				
		Días	Horas	1	2	3	4	5
13	Enviar alertas de emergencia	2.63	21				X	
14	Visualizar alertas de emergencia de familiares	2.63	21				X	
15	Visualizar alertas de emergencia de los ciudadanos	1.38	11				X	
16	Cambiar contraseña	1.75	14					X
17	Recuperar contraseña	1.88	15					X
18	Desarrollar el modelo analítico de BI	5.13	41					X

3.4.2 Fase II: Diseño

En esta fase se desarrollan las iteraciones, tomando en cuenta que estos son incrementos pequeños y manejables que se realizan en un corto periodo de tiempo.

a. Iteraciones

Para el desarrollo del presente proyecto se han establecido las siguientes iteraciones:

- ***Iteración 1.*** En base a la información establecida en la tabla 33, en el primer periodo, el plan de entrega se divide en 4 historias de usuario, las cuales se detallan a continuación:

Tabla 34. Historias de usuario Iteración N° 1

N°	Historia de usuario	Prioridad	Riesgo en desarrollo
1	Iniciar sesión sistema web	Alta	Medio
2	Cerrar sesión sistema web	Alta	Medio
3	Gestionar usuarios	Alta	Medio
4	Gestionar tipos de incidentes	Alta	Medio

En la tabla 35 se detallan las tareas a realizar en la primera iteración.

Tabla 35. Tareas Iteración N° 1

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
Historia 1. Iniciar sesión sistema web					
1	Diseño de la base de datos	4	Crear esquema de la base de datos	11/03/2024	11/03/2024
2	Desarrollo backend para inicio de sesión	4	Implementar servicios y lógica del servidor para el inicio de sesión	11/03/2024	11/03/2024
3	Diseño de la interfaz para iniciar sesión	4	Crear diseño visual de la pantalla de inicio de sesión	12/03/2024	12/03/2024
Continua en la siguiente página					

Tabla 35 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
4	Validación y pruebas para inicio de sesión	4	Realizar pruebas de aceptación	12/03/2024	12/03/2024
Historia 2. Cerrar sesión sistema web					
1	Diseño de la interfaz para cerrar sesión	1	Crear diseño visual para el cierre de sesión	13/03/2024	13/03/2024
2	Validación y pruebas para cerrar de sesión	1	Realizar pruebas de aceptación	13/03/2024	13/03/2024
Historia 3. Gestionar usuarios					
1	Diseño de la base de datos para la gestión de usuarios	4	Crear esquema de la base de datos para la administración de usuarios	13/03/2024	13/03/2024
2	Desarrollo backend para la gestión de usuarios	5	Implementar servicios y lógica del servidor para la administración de usuarios	14/03/2024	14/03/2024

Continua en la siguiente página

Tabla 35 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
3	Desarrollo de la interfaz para gestión de usuarios	5	Crear diseño visual para la administración de usuarios	15/03/2024	15/03/2024
4	Validación y pruebas para gestión de usuarios	4	Realizar pruebas de aceptación	18/03/2024	18/03/2024
Historia 4. Gestionar tipos de incidentes					
1	Diseño de la base de datos para la gestión de incidentes	3	Crear esquema de la base de datos para la gestión de incidentes	18/03/2024	18/03/2024
2	Desarrollo backend para la gestión de incidentes	3	Implementar servicios y lógica del servidor para la gestión de incidentes	19/03/2024	19/03/2024

Continua en la siguiente página

Tabla 35 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
3	Desarrollo de la interfaz para la gestión de incidentes	4	Crear diseño visual de la pantalla de incidentes	19/03/2024	19/03/2024
4	Validación y pruebas para la gestión de incidentes	3	Realizar pruebas de aceptación	20/03/2024	20/03/2024

En la Figura 9 se muestra el prototipo de la interfaz de usuario propuesta para el inicio de sesión en el sistema web.



Figura 9. Prototipo de la interfaz de inicio de sesión web

En la Figura 10 se puede observar el prototipo de la interfaz de usuario propuesto para el cierre de sesión en el sistema web.



Figura 10. Prototipo de la interfaz para cerrar sesión en el sistema web

En la Figura 11 se puede visualizar el prototipo de la interfaz de usuario propuesta para la pantalla de visualización de usuarios.

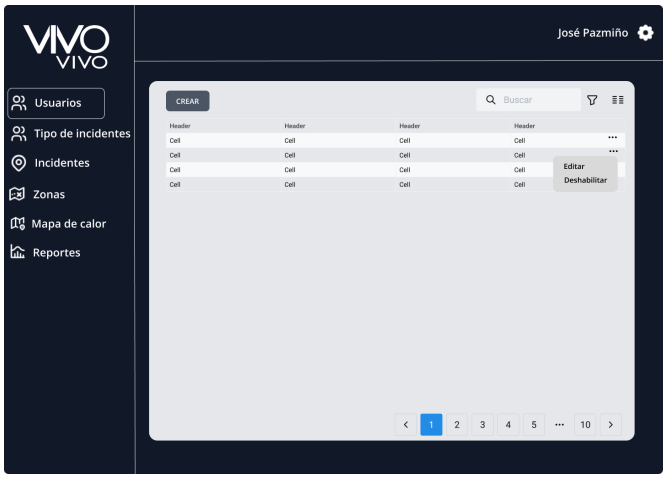


Figura 11. Prototipo de la interfaz para visualizar los usuarios

En la Figura 12 se presenta el prototipo de la interfaz de usuario propuesto para la creación y edición de un usuario en el sistema web.

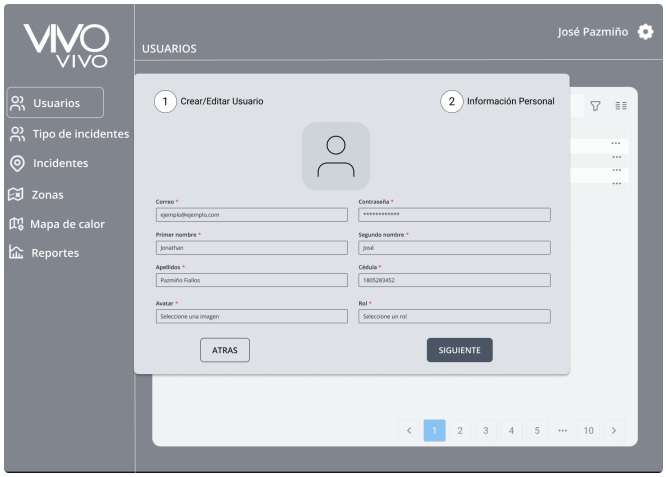


Figura 12. Prototipo de la interfaz para crear/editar usuario

En la Figura 13 se muestra el prototipo de la interfaz de usuario propuesta para la visualización de los tipos de incidentes.

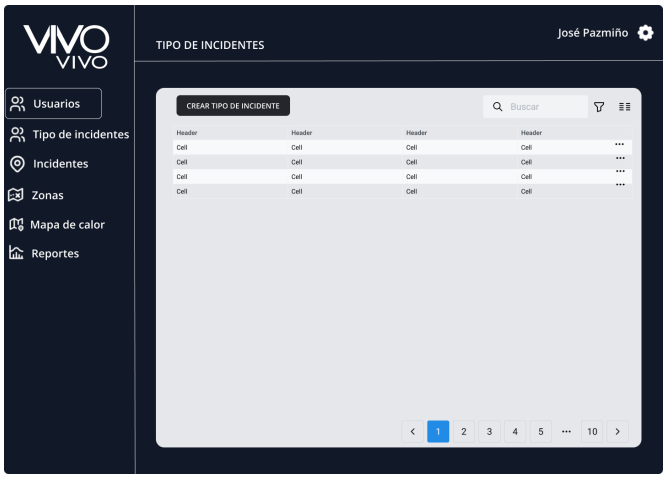


Figura 13. Prototipo de la interfaz para visualizar los tipos de incidentes

En la Figura 14 se puede observar el prototipo de la interfaz de usuario propuesta para la creación y edición de un tipo de incidente en el sistema web.

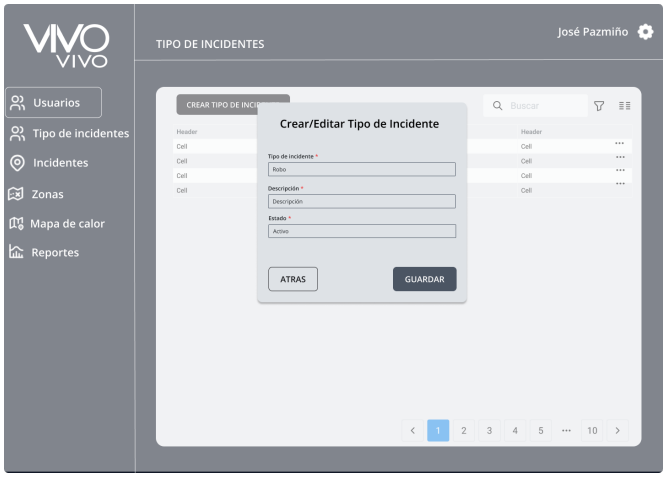


Figura 14. Prototipo de la interfaz para crear/editar un tipo de incidente

- ***Iteración 2.*** En base a la información establecida en la tabla 33, en el segundo periodo, el plan de entrega se divide en 4 historias de usuario, las cuales se detallan a continuación:

Tabla 36. Historias de usuario Iteración N° 2

N°	Historia de usuario	Prioridad	Riesgo en desarrollo
1	Gestionar zonas de vigilancia	Alta	Medio
2	Asignar policías a las zonas de vigilancia	Alta	Medio
3	Gestionar alertas de incidentes	Alta	Medio
4	Visualizar mapa de calor	Alta	Medio

En la tabla 37 se detallan las tareas a realizar en la segunda iteración.

Tabla 37. Tareas Iteración N° 2

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
Historia 1. Gestionar zonas de vigilancia					
1	Diseño de la base de datos para la gestión de zonas de vigilancia	4	Crear esquema de la base de datos para zonas de vigilancia	21/03/2024	21/03/2024
2	Desarrollo backend para la gestión de zonas de vigilancia	5	Implementar servicios y lógica del servidor para zonas de vigilancia	22/03/2024	22/03/2024
3	Desarrollo de la interfaz para la gestión de zonas de vigilancia	8	Crear diseño visual para la gestión de zonas de vigilancia	25/03/2024	26/03/2024
4	Validación y pruebas para la gestión de zonas de vigilancia	4	Realizar pruebas de aceptación	27/03/2024	27/03/2024
Historia 2. Asignar policías a las zonas de vigilancia					
Continua en la siguiente página					

Tabla 37 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
1	Diseño de la base de datos para asignar policías a las zonas de vigilancia	3	Crear esquema de la base de datos para asignación de policías	28/03/2024	28/03/2024
2	Desarrollo backend para asignar policías a las zonas de vigilancia	3	Implementar servicios y lógica del servidor para asignación de policías	28/03/2024	28/03/2024
3	Desarrollo de la interfaz para asignar policías a las zonas de vigilancia	4	Crear diseño visual para asignación de policías	29/03/2024	29/03/2024

Continua en la siguiente página

Tabla 37 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
4	Validación y pruebas para asignar policías a las zonas de vigilancia	3	Realizar pruebas de aceptación	01/04/2024	01/04/2024
Historia 3. Gestionar alertas de incidentes					
1	Diseño de la base de datos para Gestionar alertas de incidentes	5	Crear esquema de la base de datos para alertas de incidentes	02/04/2024	02/04/2024
2	Desarrollo backend para Gestionar alertas de incidentes	8	Implementar servicios y lógica del servidor para alertas de incidentes	03/04/2024	04/04/2024
3	Desarrollo de la interfaz para Gestionar alertas de incidentes	6	Crear diseño visual para visualización de alertas de incidentes	05/04/2024	05/04/2024

Continua en la siguiente página

Tabla 37 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
4	Validación y pruebas para Gestionar alertas de incidentes	4	Realizar pruebas de aceptación	08/04/2024	08/04/2024
Historia 4. Visualizar mapa de calor					
1	Desarrollo backend para visualizar mapa de calor	4	Implementar servicios y lógica del servidor para el mapa de calor	09/04/2024	09/04/2024
2	Desarrollo de la interfaz para visualizar mapa de calor	4	Crear diseño visual para visualización del mapa de calor	10/04/2024	10/04/2024
3	Validación y pruebas para visualizar mapa de calor	3	Realizar pruebas de aceptación	11/04/2024	11/04/2024

En la Figura 15 se muestra el prototipo de la interfaz de usuario propuesta para la visualización de las zonas de vigilancia.



Figura 15. Prototipo de la interfaz para visualizar las zonas de vigilancia

En la Figura 16 se presenta el prototipo de la interfaz de usuario propuesta para la creación y edición de una zona de vigilancia en el sistema web.

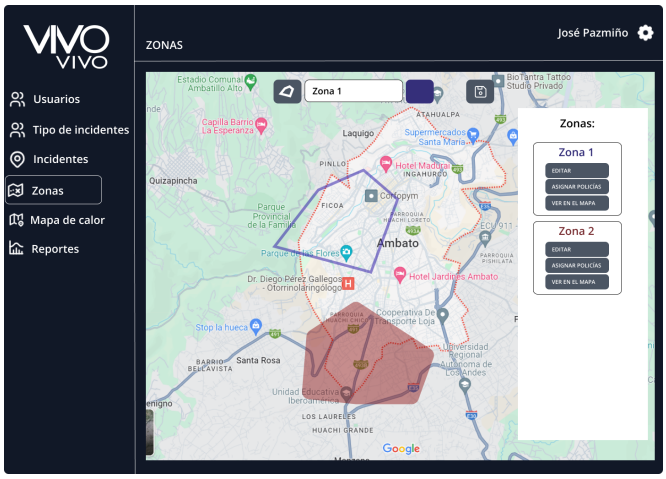


Figura 16. Prototipo de la interfaz para crear/editar una zona de vigilancia

En la Figura 17 se puede observar el prototipo de la interfaz de usuario propuesta para asignar o desasignar policías en la zona de vigilancia.

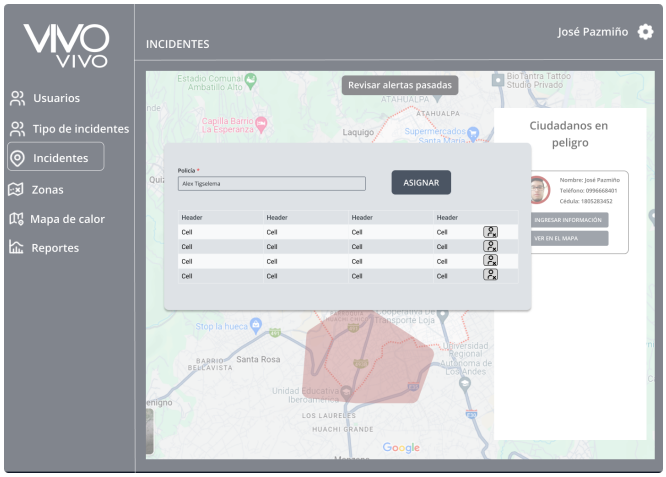


Figura 17. Prototipo de la interfaz para asignar/desasignar policías a la zona de vigilancia

En la Figura 18 se muestra el prototipo de la interfaz de usuario propuesta para la visualización de las alertas de incidentes en el sistema web.

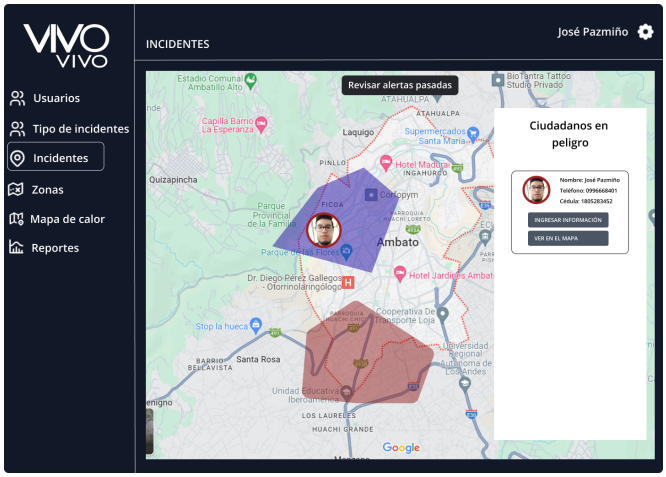


Figura 18. Prototipo de la interfaz para visualizar las alertas de incidentes

En la Figura 19 se presenta el prototipo de la interfaz de usuario propuesta para la visualización del mapa de calor en el sistema web.

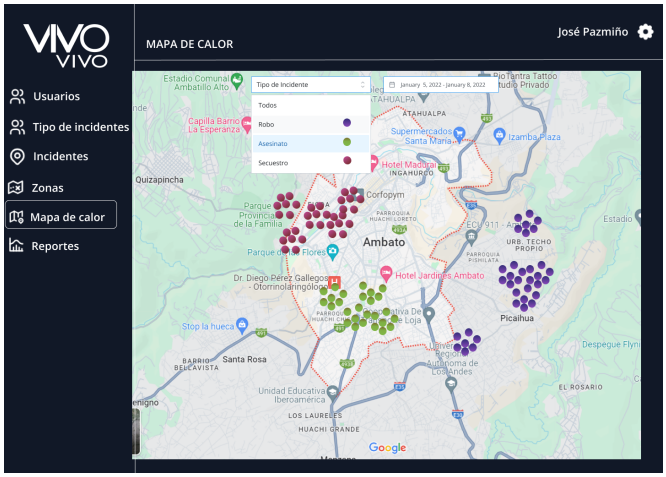


Figura 19. Prototipo de la interfaz para visualizar el mapa de calor

- ***Iteración 3.*** En base a la información establecida en la tabla 33, en el tercer periodo, el plan de entrega se divide en 3 historias de usuario, las cuales se detallan a continuación:

Tabla 38. Historias de usuario Iteración N° 3

N°	Historia de usuario	Prioridad	Riesgo en desarrollo
1	Iniciar sesión aplicación móvil	Alta	Medio
2	Cerrar sesión aplicación móvil	Alta	Medio
3	Registro de usuario	Alta	Medio

En la tabla 39 se detallan las tareas a realizar en la tercera iteración.

Tabla 39. Tareas Iteración N° 3

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
Historia 1. Iniciar sesión aplicación móvil					
Continua en la siguiente página					

Tabla 39 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
1	Desarrollo de la interfaz para iniciar sesión en la aplicación móvil	5	Crear diseño visual para la pantalla de inicio de sesión en la app móvil	12/04/2024	12/04/2024
2	Validación y pruebas para iniciar sesión en la aplicación móvil	3	Realizar pruebas de aceptación	15/04/2024	15/04/2024
Historia 2. Cerrar sesión aplicación móvil					
1	Diseño de la interfaz para cerrar sesión en la aplicación móvil	3	Crear diseño visual para la pantalla de cierre de sesión en la app móvil	16/04/2024	16/04/2024
2	Validación y pruebas para cerrar sesión en la aplicación móvil	1	Realizar pruebas de aceptación	17/04/2024	17/04/2024
Historia 3. Registro de usuario					

Continua en la siguiente página

Tabla 39 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
1	Desarrollo de la interfaz para el registro de usuario en la aplicación móvil	8	Crear diseño visual para la pantalla de registro de usuario en la app móvil	18/04/2024	19/04/2024
2	Validación y pruebas para el registro de usuario en la aplicación móvil	3	Realizar pruebas de aceptación	22/04/2024	22/04/2024

En la Figura 20 se muestra el prototipo de la interfaz de usuario propuesta para el inicio de sesión en la aplicación móvil.



Figura 20. Prototipo de la interfaz para iniciar sesión en la aplicación móvil

En la Figura 21 se presenta el prototipo de la interfaz de usuario propuesta para el cierre de sesión en la aplicación móvil.

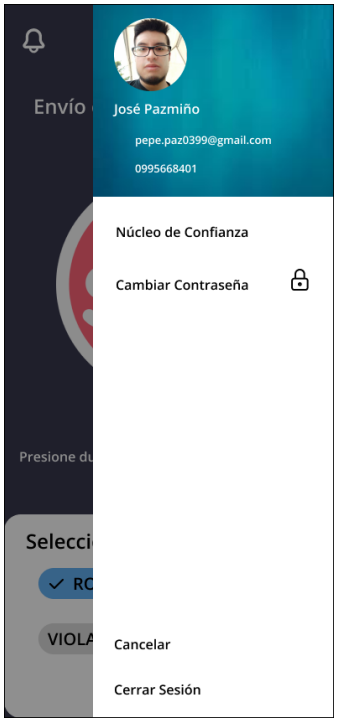


Figura 21. Prototipo de la interfaz para cerrar sesión en la aplicación móvil

En la Figura 22 se puede observar el prototipo de la interfaz de usuario propuesta para

el registro de usuarios en la aplicación móvil.

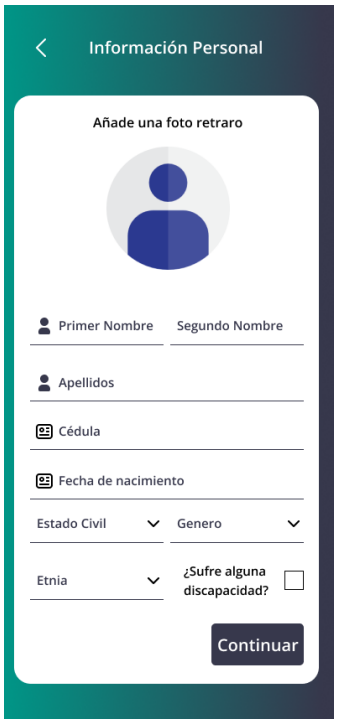


Figura 22. Prototipo de la interfaz para el registro de usuarios en la aplicación móvil

- ***Iteración 4.*** En base a la información establecida en la tabla 33, en el cuarto periodo, el plan de entrega se divide en 4 historias de usuario, las cuales se detallan a continuación:

Tabla 40. Historias de usuario Iteración N° 4

N°	Historia de usuario	Prioridad	Riesgo en desarrollo
1	Asignar miembros al grupo familiar	Alta	Medio
2	Enviar alertas de emergencia	Alta	Medio
3	Visualizar alertas de emergencia de familiares	Alta	Medio

Continua en la siguiente página

Tabla 40 – continuación de la página anterior

N°	Historia de usuario	Prioridad	Riesgo en desarrollo
4	Visualizar alertas de emergencia de los ciudadanos	Alta	Medio

En la tabla 41 se detallan las tareas a realizar en la cuarta iteración.

Tabla 41. Tareas Iteración N° 4

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
Historia 1. Asignar miembros al grupo familiar					
1	Diseño de la base de datos para asignar miembros al grupo familiar	4	Crear esquema de la base de datos para la asignación de miembros al grupo familiar	23/04/2024	23/04/2024
2	Desarrollo backend para asignar miembros al grupo familiar	5	Implementar servicios y lógica del servidor para la asignación de miembros al grupo familiar	24/04/2024	24/04/2024
Continua en la siguiente página					

Tabla 41 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
3	Desarrollo de la interfaz para asignar miembros al grupo familiar	6	Crear diseño visual para la pantalla de asignación de miembros al grupo familiar	25/04/2024	26/04/2024
4	Validación y pruebas para asignar miembros al grupo familiar	3	Realizar pruebas de aceptación	29/04/2024	29/04/2024
Historia 2. Enviar alertas de emergencia					
1	Diseño de la base de datos para enviar alertas de emergencia	5	Crear esquema de la base de datos para el envío de alertas	30/04/2024	30/04/2024
2	Desarrollo backend para enviar alertas de emergencia	8	Implementar servicios y lógica del servidor para el envío de alertas	01/05/2024	02/05/2024

Continua en la siguiente página

Tabla 41 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
3	Desarrollo de la interfaz para enviar alertas de emergencia	5	Crear diseño visual para la pantalla de envío de alertas	03/05/2024	03/05/2024
4	Validación y pruebas para enviar alertas de emergencia	3	Realizar pruebas de aceptación	06/05/2024	06/05/2024
Historia 3. Visualizar alertas de emergencia de familiares					
1	Diseño de la base de datos para visualizar alertas de emergencia de familiares	5	Crear esquema de la base de datos para la visualización de alertas de familiares	07/05/2024	07/05/2024

Continua en la siguiente página

Tabla 41 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
2	Desarrollo backend para visualizar alertas de emergencia de familiares	6	Implementar servicios y lógica del servidor para la visualización de alertas de familiares	08/05/2024	09/05/2024
3	Desarrollo de la interfaz para visualizar alertas de emergencia de familiares	7	Crear diseño visual para la pantalla de visualización de alertas de familiares	10/05/2024	13/05/2024
4	Validación y pruebas para visualizar alertas de emergencia de familiares	3	Realizar pruebas de aceptación	14/05/2024	14/05/2024
Historia 4. Visualizar alertas de emergencia de los ciudadanos					

Continua en la siguiente página

Tabla 41 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
1	Desarrollo backend para visualizar alertas de emergencia de los ciudadanos	3	Implementar servicios y lógica del servidor para la visualización de alertas de ciudadanos	15/05/2024	15/05/2024
2	Desarrollo de la interfaz para visualizar alertas de emergencia de los ciudadanos	5	Crear diseño visual para la pantalla de visualización de alertas de ciudadanos	16/05/2024	16/05/2024
3	Validación y pruebas para visualizar alertas de emergencia de los ciudadanos	3	Realizar pruebas de aceptación	17/05/2024	17/05/2024

En la Figura 23 se muestra el prototipo de la interfaz de usuario propuesta para la visualización de los miembros del grupo familiar en la aplicación móvil.

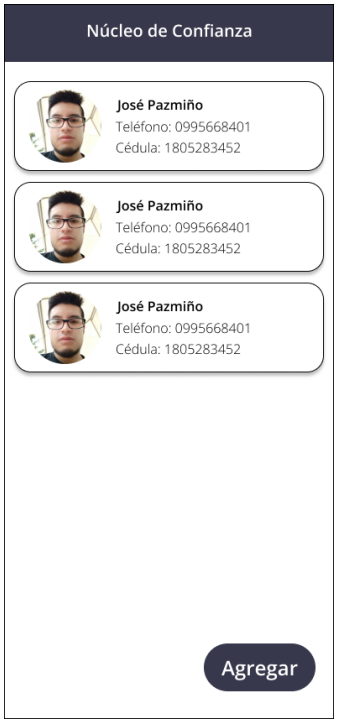


Figura 23. Prototipo de la interfaz para visualizar los miembros del grupo familiar

En la Figura 24 se presenta el prototipo de la interfaz de usuario propuesta para agregar miembros al grupo familiar en la aplicación móvil.



Figura 24. Prototipo de la interfaz para agregar miembros al grupo familiar

En la Figura 25 se muestra el prototipo de la interfaz de usuario propuesta para enviar

una alerta de incidente en la aplicación móvil.

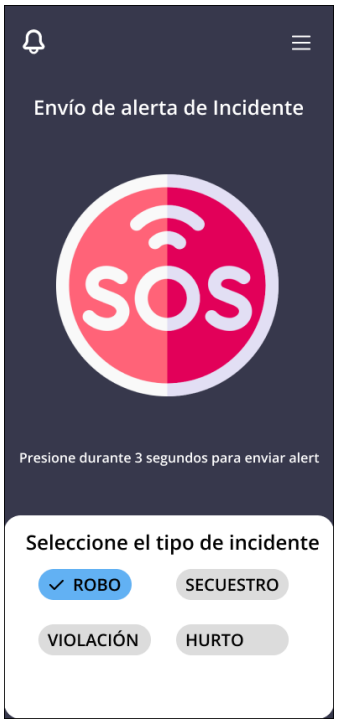


Figura 25. Prototipo de la interfaz para enviar una alerta de incidente

En la Figura 26 se presenta el prototipo de la interfaz de usuario propuesta para visualizar la notificación de alerta de incidente en la aplicación móvil.

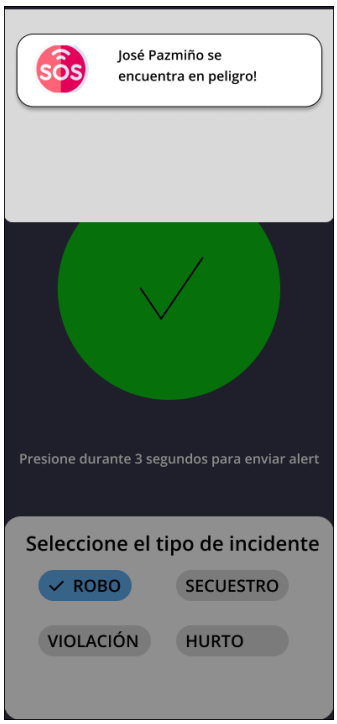


Figura 26. Prototipo de la interfaz para visualizar la notificación de alerta de incidente

En la Figura 27 se muestra el prototipo de la interfaz de usuario propuesta para visualizar las alertas de emergencia de familiares en la aplicación móvil.

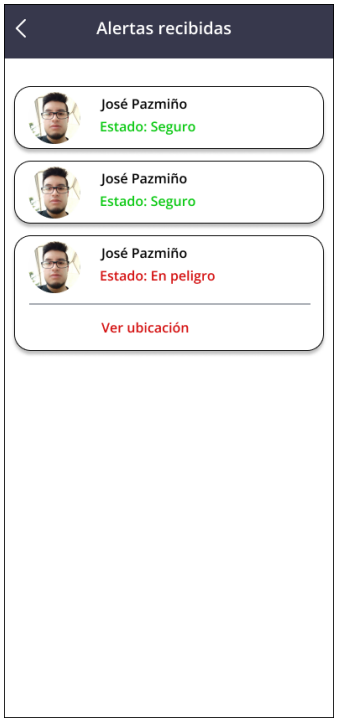


Figura 27. Prototipo de la interfaz para visualizar las alertas de emergencia de familiares

En la Figura 28 se presenta el prototipo de la interfaz de usuario propuesta para visualizar la ubicación de la alerta de emergencia de un familiar en la aplicación móvil.



Figura 28. Prototipo de la interfaz para visualizar la ubicación de la alerta de emergencia de un familiar

En la Figura 29 se muestra el prototipo de la interfaz de usuario propuesta para visualizar las alertas de emergencia de ciudadanos en la aplicación móvil de la policía.

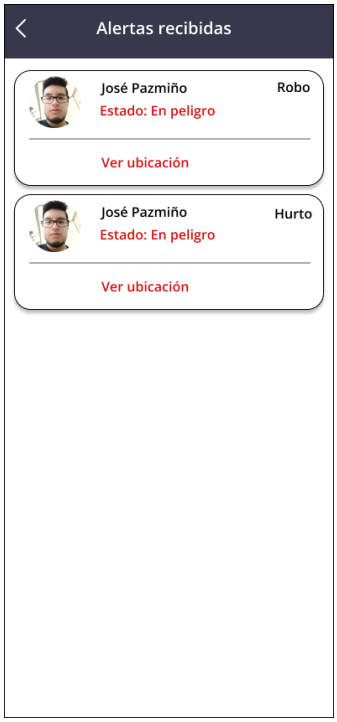


Figura 29. Prototipo de la interfaz para visualizar las alertas de emergencia de ciudadanos

En la Figura 30 se presenta el prototipo de la interfaz de usuario propuesta para visualizar la ubicación de la alerta de emergencia de un ciudadano en la aplicación móvil de la policía.



Figura 30. Prototipo de la interfaz para visualizar la ubicación de la alerta de emergencia de un ciudadano

- ***Iteración 5.*** En base a la información establecida en la tabla 33, en el quinto periodo, el plan de entrega se divide en 3 historias de usuario, las cuales se detallan a continuación:

Tabla 42. Historias de usuario Iteración N° 5

N°	Historia de usuario	Prioridad	Riesgo en desarrollo
1	Cambiar contraseña	Media	Medio
2	Recuperar contraseña	Media	Medio
4	Desarrollar el modelo analítico de BI	Media	Medio

En la tabla 43 se detallan las tareas a realizar en la quinta iteración.

Tabla 43. Tareas Iteración N° 5

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
Historia 1. Cambiar contraseña					
1	Diseño de la base de datos para cambiar contraseña	3	Crear esquema de la base de datos para el cambio de contraseñas	20/05/2024	20/05/2024
2	Desarrollo backend para cambiar contraseña	3	Implementar servicios y lógica del servidor para el cambio de contraseñas	21/05/2024	21/05/2024
3	Desarrollo de la interfaz para cambiar contraseña	5	Crear diseño visual para la pantalla de cambio de contraseñas	22/05/2024	22/05/2024
4	Validación y pruebas para cambiar contraseña	3	Realizar pruebas de aceptación	23/05/2024	23/05/2024
Historia 2. Recuperar contraseña					

Continua en la siguiente página

Tabla 43 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
1	Diseño de la base de datos para recuperar contraseña	4	Crear esquema de la base de datos para la recuperación de contraseñas	24/05/2024	24/05/2024
2	Desarrollo backend para recuperar contraseña	4	Implementar servicios y lógica del servidor para la recuperación de contraseñas	27/05/2024	27/05/2024
3	Desarrollo de la interfaz para recuperar contraseña	4	Crear diseño visual para la pantalla de recuperación de contraseñas	28/05/2024	28/05/2024
4	Validación y pruebas para recuperar contraseña	3	Realizar pruebas de aceptación	29/05/2024	29/05/2024
Historia 3. Desarrollar el modelo analítico de BI					
1	Seleccionar las preguntas del negocio	4	Identificar las preguntas clave que el negocio necesita responder	30/05/2024	30/05/2024

Continua en la siguiente página

Tabla 43 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
2	Identificar indicadores y perspectivas	3	Definir los KPIs y las perspectivas necesarias para el análisis	31/05/2024	31/05/2024
3	Declarar el grano	3	Determinar el nivel de detalle de los datos para el análisis	03/06/2024	03/06/2024
4	Identificar las dimensiones y sus atributos	8	Especificar las dimensiones de análisis y sus atributos relevantes	04/06/2024	05/06/2024
5	Crear esquema dimensional	3	Diseñar el esquema dimensional para el modelo de datos	06/06/2024	06/06/2024
6	Realizar el Proceso ETL	5	Implementar el proceso de extracción, transformación y carga de datos	07/06/2024	07/06/2024
7	Creación del Cubo OLAP	5	Desarrollar el cubo OLAP para el análisis multidimensional	10/06/2024	10/06/2024

Continua en la siguiente página

Tabla 43 – continuación de la página anterior

N°	Tarea	Puntos estimados	Descripción	Fecha de inicio	Fecha de finalización
8	Visualización de datos (Generar reportes)	16	Crear reportes y dashboards para la visualización de datos	11/06/2024	12/06/2024

En la Figura 31 se muestra el prototipo de la interfaz de usuario propuesta para cambiar la contraseña en la aplicación móvil.



Figura 31. Prototipo de la interfaz para cambiar la contraseña en la aplicación móvil

En la Figura 32 se presenta el prototipo de la interfaz de usuario propuesta para recuperar la contraseña en la aplicación móvil.



Figura 32. Prototipo de la interfaz para recuperar la contraseña en la aplicación móvil

b. Tarjetas CRC

Durante la ejecución del proyecto, se emplearon tarjetas CRC (Clase, Responsabilidad y Colaboración) para identificar las responsabilidades y los colaboradores de cada historia de usuario, con el fin de optimizar el diseño del software.

En la tabla 44 se presenta la tarjeta CRC de la historia de usuario 1, la cual corresponde a la funcionalidad de iniciar sesión en el sistema web.

Tabla 44. Tarjeta CRC - Historia 1: Iniciar sesión sistema web

Iniciar sesión sistema web	
Responsabilidades	**Colaboradores**
Diseño de la base de datos	Usuario
Desarrollo backend para inicio de sesión	Roles
Diseño de la interfaz para iniciar sesión	

Observaciones: Ninguna

En la tabla 45 se presenta la tarjeta CRC de la historia de usuario 2, la cual corresponde a la funcionalidad de cerrar sesión en el sistema web.

Tabla 45. Tarjeta CRC - Historia 2: Cerrar sesión sistema web

Cerrar sesión sistema web	
Responsabilidades	**Colaboradores**
Diseño de la interfaz para cerrar sesión	Usuario
Observaciones: Ninguna	

En la tabla 46 se presenta la tarjeta CRC de la historia de usuario 3, la cual corresponde a la funcionalidad de gestionar usuarios.

Tabla 46. Tarjeta CRC - Historia 3: Gestionar usuarios

Gestionar usuarios	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para la gestión de usuarios	Usuario
Desarrollo backend para la gestión de usuarios	Roles
Desarrollo de la interfaz para gestión de usuarios	
Observaciones: Ninguna	

En la tabla 47 se presenta la tarjeta CRC de la historia de usuario 4, la cual corresponde a la funcionalidad de gestionar tipos de incidentes.

Tabla 47. Tarjeta CRC - Historia 4: Gestionar tipos de incidentes

Gestionar tipos de incidentes	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para la gestión de incidentes	Tipo de incidente
Desarrollo backend para la gestión de incidentes	Policía
Desarrollo de la interfaz para la gestión de incidentes	
Observaciones: Ninguna	

En la tabla 48 se presenta la tarjeta CRC de la historia de usuario 5, la cual corresponde a la funcionalidad de gestionar zonas de vigilancia.

Tabla 48. Tarjeta CRC - Historia 5: Gestionar zonas de vigilancia

Gestionar zonas de vigilancia	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para la gestión de zonas de vigilancia	Zonas de vigilancia
Desarrollo backend para la gestión de zonas de vigilancia	Policía
Desarrollo de la interfaz para la gestión de zonas de vigilancia	
Observaciones: Ninguna	

En la tabla 49 se presenta la tarjeta CRC de la historia de usuario 6, la cual corresponde a la funcionalidad de asignar policías a las zonas de vigilancia.

Tabla 49. Tarjeta CRC - Historia 6: Asignar policías a las zonas de vigilancia

Asignar policías a las zonas de vigilancia	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para asignar policías a las zonas de vigilancia	Zonas de vigilancia
Desarrollo backend para asignar policías a las zonas de vigilancia	Policía
Desarrollo de la interfaz para asignar policías a las zonas de vigilancia	Zonas de vigilancia
Observaciones: Ninguna	

En la tabla 50 se presenta la tarjeta CRC de la historia de usuario 7, la cual corresponde a la funcionalidad de gestionar alertas de incidentes.

Tabla 50. Tarjeta CRC - Historia 7: Gestionar alertas de incidentes

Gestionar alertas de incidentes	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para Gestionar alertas de incidentes	Tipo de incidente
Desarrollo backend para Gestionar alertas de incidentes	Policía
Desarrollo de la interfaz para Gestionar alertas de incidentes	
Observaciones: Ninguna	

En la tabla 51 se presenta la tarjeta CRC de la historia de usuario 8, la cual corresponde

a la funcionalidad de visualizar mapa de calor.

Tabla 51. Tarjeta CRC - Historia 8: Visualizar mapa de calor

Visualizar mapa de calor	
Responsabilidades	**Colaboradores**
Desarrollo backend para visualizar mapa de calor	Zonas de vigilancia
Desarrollo de la interfaz para visualizar mapa de calor	
Observaciones: Ninguna	

En la tabla 52 se presenta la tarjeta CRC de la historia de usuario 9, la cual corresponde a la funcionalidad de iniciar sesión en la aplicación móvil.

Tabla 52. Tarjeta CRC - Historia 9: Iniciar sesión aplicación móvil

Iniciar sesión aplicación móvil	
Responsabilidades	**Colaboradores**
Desarrollo de la interfaz para iniciar sesión en la aplicación móvil	Usuario
Observaciones: Ninguna	

En la tabla 53 se presenta la tarjeta CRC de la historia de usuario 10, la cual corresponde a la funcionalidad de cerrar sesión en la aplicación móvil.

Tabla 53. Tarjeta CRC - Historia 10: Cerrar sesión aplicación móvil

Cerrar sesión aplicación móvil	
Responsabilidades	**Colaboradores**
Diseño de la interfaz para cerrar sesión en la aplicación móvil	Usuario

Observaciones: Ninguna

En la tabla 54 se presenta la tarjeta CRC de la historia de usuario 11, la cual corresponde a la funcionalidad de registro de usuario.

Tabla 54. Tarjeta CRC - Historia 11: Registro de usuario

Registro de usuario	
Responsabilidades	**Colaboradores**
Desarrollo de la interfaz para el registro de usuario en la aplicación móvil	Usuario
Observaciones: Ninguna	

En la tabla 55 se presenta la tarjeta CRC de la historia de usuario 12, la cual corresponde a la funcionalidad de asignar miembros al grupo familiar.

Tabla 55. Tarjeta CRC - Historia 12: Asignar miembros al grupo familiar

Asignar miembros al grupo familiar	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para asignar miembros al grupo familiar	Grupo familiar
Desarrollo backend para asignar miembros al grupo familiar	Usuario, Roles
Desarrollo de la interfaz para asignar miembros al grupo familiar	
Observaciones: Ninguna	

En la tabla 56 se presenta la tarjeta CRC de la historia de usuario 13, la cual corresponde a la funcionalidad de enviar alertas de emergencia.

Tabla 56. Tarjeta CRC - Historia 13: Enviar alertas de emergencia

Enviar alertas de emergencia	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para enviar alertas de emergencia	Alarma
Desarrollo backend para enviar alertas de emergencia	Policía
Desarrollo de la interfaz para enviar alertas de emergencia	Usuario
Observaciones: Ninguna	

En la tabla 57 se presenta la tarjeta CRC de la historia de usuario 14, la cual corresponde a la funcionalidad de visualizar alertas de emergencia de familiares.

Tabla 57. Tarjeta CRC - Historia 14: Visualizar alertas de emergencia de familiares

Visualizar alertas de emergencia de familiares	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para visualizar alertas de emergencia de familiares	Grupo familiar
Desarrollo backend para visualizar alertas de emergencia de familiares	Usuario
Desarrollo de la interfaz para visualizar alertas de emergencia de familiares	
Observaciones: Ninguna	

En la tabla 58 se presenta la tarjeta CRC de la historia de usuario 15, la cual corresponde a la funcionalidad de visualizar alertas de emergencia de los ciudadanos.

Tabla 58. Tarjeta CRC - Historia 15: Visualizar alertas de emergencia de los ciudadanos

Visualizar alertas de emergencia de los ciudadanos	
Responsabilidades	**Colaboradores**
Desarrollo backend para visualizar alertas de emergencia de los ciudadanos	Alarma
Desarrollo de la interfaz para visualizar alertas de emergencia de los ciudadanos	Usuario
Observaciones: Ninguna	

En la tabla 59 se presenta la tarjeta CRC de la historia de usuario 16, la cual corresponde a la funcionalidad de cambiar contraseña.

Tabla 59. Tarjeta CRC - Historia 16: Cambiar contraseña

Cambiar contraseña	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para cambiar contraseña	Usuario
Desarrollo backend para cambiar contraseña	
Desarrollo de la interfaz para cambiar contraseña	
Observaciones: Ninguna	

En la tabla 60 se presenta la tarjeta CRC de la historia de usuario 17, la cual corresponde a la funcionalidad de recuperar contraseña.

Tabla 60. Tarjeta CRC - Historia 17: Recuperar contraseña

Recuperar contraseña	
Responsabilidades	**Colaboradores**
Diseño de la base de datos para recuperar contraseña	Usuario
Desarrollo backend para recuperar contraseña	
Desarrollo de la interfaz para recuperar contraseña	
Observaciones: Ninguna	

En la tabla 61 se presenta la tarjeta CRC de la historia de usuario 18, la cual corresponde a la funcionalidad de desarrollar el modelo analítico de BI.

Tabla 61. Tarjeta CRC - Historia 18: Desarrollar el modelo analítico de BI

Desarrollar el modelo analítico de BI	
Responsabilidades	**Colaboradores**
Seleccionar las preguntas del negocio	Usuario
Identificar indicadores y perspectivas	
Declarar el grano	
Identificar las dimensiones y sus atributos	
Crear esquema dimensional	
Realizar el Proceso ETL	
Creación del Cubo OLAP	
Visualización de datos (Generar reportes)	
Observaciones: Ninguna	

c. Diseño modelo de base de datos

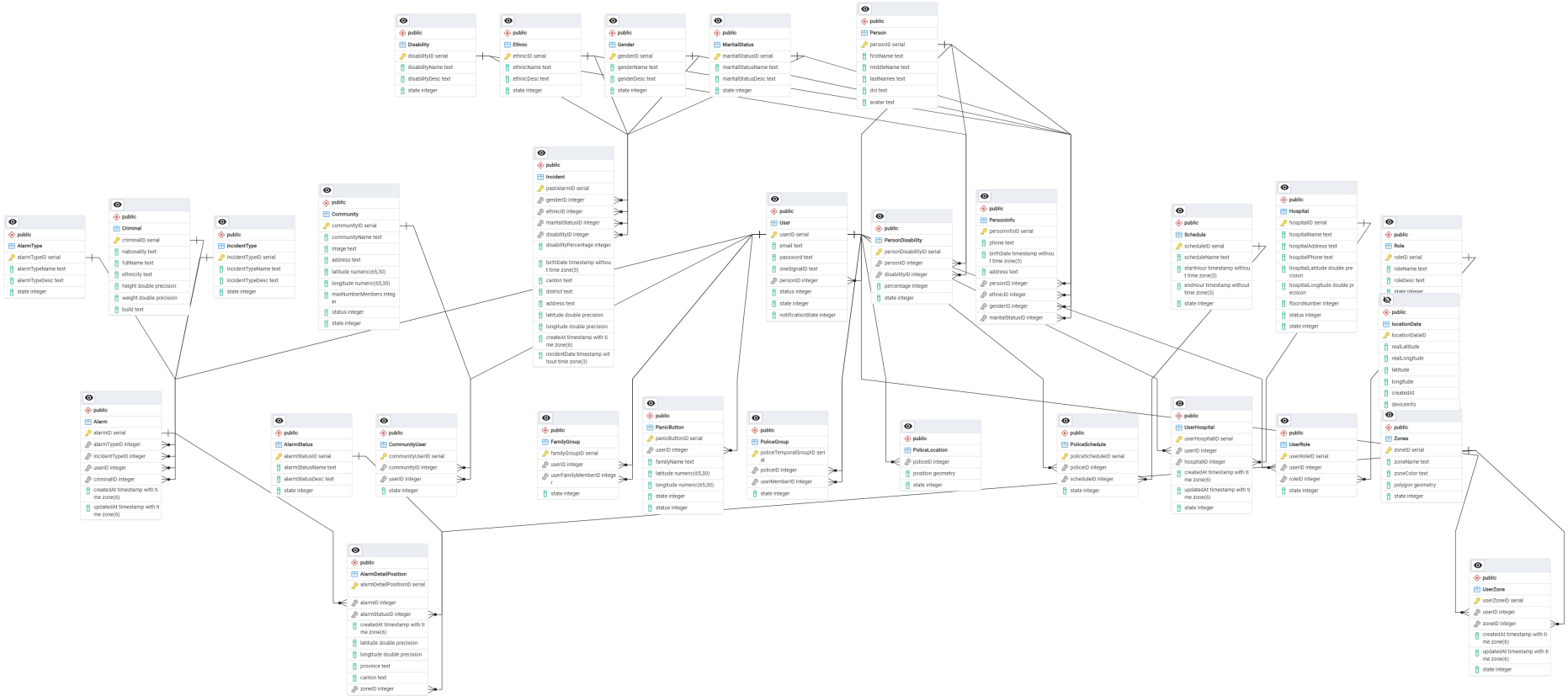


Figura 33. Esquema de la base de datos

3.4.3 Fase III: Codificación

En esta sección se describe la construcción de la propuesta del sistema, la cual se divide en dos partes fundamentales: la API y la interfaz de usuario.

a. Configuración API (Backend)

- ***Dependecias de la API.*** Las dependencias utilizadas en la API se gestionaron mediante npm (Node Package Manager), el cual es un gestor de paquetes para el lenguaje de programación JavaScript, el cual permite instalar, actualizar y eliminar paquetes mediante un archivo de configuración llamado package.json. En en Anexo F muestra el archivo package.json con las dependencias utilizadas en la API.

- ***Configurar variables de entorno de la API.*** Para la configuración de las variables de entorno de la API se utilizó un archivo .env, el cual contiene las propiedades de la aplicación, como la URL de conexión a la base de datos, la configuración para el envío de correos electrónicos, la configuración del servicio de almacenamiento de imágenes, la configuración del servicio de notificaciones push y la configuración del servicio de Google Maps y el secret key para la generación de tokens JWT. En el Anexo G se muestra el archivo .env con las variables de entorno de la API.

- ***Configurción de la seguridad de la API.*** Para la configuración de la seguridad de la API, se utilizó una estrategia de autenticación basada en JWT (Json Web Token) y el sistema de Guards que proporciona NestJS. La estrategia de autenticación JWT se implementó mediante un middleware que verifica la validez del token JWT en cada solicitud realizada a la API. Este middleware se encarga de extraer el token del encabezado de autorización y verificar la firma del token con la clave secreta, además de comprobar si el usuario tiene los permisos necesarios para acceder a los recursos protegidos. En el Anexo H se muestra la configuración de la seguridad de la API.

- ***Accesso y persistencia de datos.*** Para administrar el acceso y la persistencia de datos en la API, se utilizó un ORM (Object-Relational Mapping) llamado PrismaJS, que emplea un lenguaje de modelado de datos propio para definir las entidades de la aplicación mediante un archivo ”schema.prisma”, el cual contiene los modelos y

relaciones de la base de datos, como se muestra en el Anexo I. PrismaJS facilita la interacción con la base de datos PostgreSQL mediante consultas SQL generadas automáticamente a partir de las operaciones CRUD (Create, Read, Update, Delete) realizadas en la API, lo que permite mantener una sincronía entre TypeScript y la base de datos. De este modo, cualquier cambio en el esquema de la base de datos se reflejará automáticamente en el código de la API.

- ***Validación de datos.*** Para la validación de datos en la API, se utilizó la librería class-validator, la cual permite definir reglas de validación mediante decoradores en las clases de los modelos de datos. Estos decoradores se encargan de validar los datos de entrada en las solicitudes realizadas a la API, como se muestra en el Anexo J.

- ***Envío de correos electrónicos.*** Para el envío de correos electrónicos en la API se utilizó la librería de nest mailer, la cual trabaja sobre la librería nodemailer para el envío de correos electrónicos. Para la configuración del servicio de envío de correos se utilizó el SMTP (Simple Mail Transfer Protocol) de Gmail, este permite enviar correos electrónicos utilizando la cuenta de Gmail del usuario administrador, como se muestra en el Anexo K.

Para plantilla de correo electrónico se utilizó el sistema de plantillas de handlebars, el cual permite generar HTML dinámico a partir de un archivo de plantilla y datos de contexto en formato JSON. En el Anexo O se muestra la plantilla de correo electrónico utilizada para recuperar la contraseña de un usuario.

- ***Configuración y uso del servicio de almacenamiento de imágenes.*** Para el almacenamiento de imágenes en la API se utilizó el servicio de Cloudinary, el cual permite almacenar y gestionar imágenes en la nube. Cloudinary provee una librería para NodeJS que facilita la subida de imágenes, para la configuración se requiere obtener una api key, api secret y cloud name, como se muestra en el Anexo L.

Una vez configurado el servicio de Cloudinary, se procedió a implementar la subida de imágenes en la API. Para ello, se creó un servicio que abstrae la lógica de la librería de Cloudinary y permite subir imágenes a Cloudinary y obtener la URL de la imagen subida, como se muestra en el Anexo M. Este servicio se inyectó en el controlador de la

API y se utilizó en el endpoint de creación de usuarios para subir la fotografía de perfil del usuario, como se muestra en el Anexo N.

- ***Configuración y uso del servicio de notificaciones push.*** Para el envío de notificaciones push en la API, se utilizó el servicio de OneSignal, que permite enviar notificaciones a dispositivos móviles y navegadores web. OneSignal provee una librería para Node.js que facilita el envío de notificaciones push. Para la configuración, se requiere obtener una app ID y una API key, como se muestra en el Anexo P.

Con la configuración realizada, se procedió a implementar una abstracción del API de la librería de OneSignal en un servicio que proporciona una instancia del cliente para el envío de notificaciones, como se puede visualizar en el Anexo P. Este servicio se inyectó en el controlador del módulo de notificaciones push, en el cual se implementó un endpoint para enviar notificaciones a los miembros del grupo familiar del usuario y a los policías dentro de la zona de emergencia, como se puede observar en el Anexo Q.

- ***Modulos.*** NestJS permite organizar la aplicación en módulos, los cuales contienen controladores, servicios y dto (Data Transfer Object). Los módulos se encargan de importar y exportar los componentes de la aplicación, lo que facilita la reutilización de código y la separación de responsabilidades. En el Anexo R se muestra la estructura de los módulos de la API.

- ***Controladores.*** Los controladores en NestJS son clases que se encargan de gestionar las solicitudes HTTP y devolver una respuesta al cliente. Cada controlador contiene una serie de métodos que se corresponden con los diferentes endpoints de la API además de los decoradores que definen los permisos de acceso a los recursos protegidos. En el Anexo S se muestra la estructura de los controladores de la API.

- ***Servicios.*** Los servicios en NestJS son clases que contienen la lógica de negocio de la aplicación y se encargan de interactuar con la base de datos y otros servicios externos. Los servicios se inyectan en los controladores y otros servicios mediante la inyección de dependencias, lo que permite reutilizar la lógica de negocio en diferentes partes de la aplicación. En el Anexo T se muestra la estructura de los servicios de la API.

- ***DTO (Data Transfer Object).*** Los DTO en NestJS son clases que se utilizan para transferir datos entre los controladores y los servicios de la aplicación. Los DTO definen la estructura de los datos que se envían y reciben en las solicitudes HTTP, lo que facilita la validación de datos y la prevención de errores en la aplicación. En el Anexo U se muestra la estructura de los DTO de la API.

- ***Documentación de la API.*** Para la documentación de la API se utilizó la librería Swagger, la cual permite generar una documentación interactiva de la API a partir de los decoradores y comentarios en el código fuente. La documentación de la API se puede visualizar en la ruta /api-docs, como se muestra en la Figura 34.

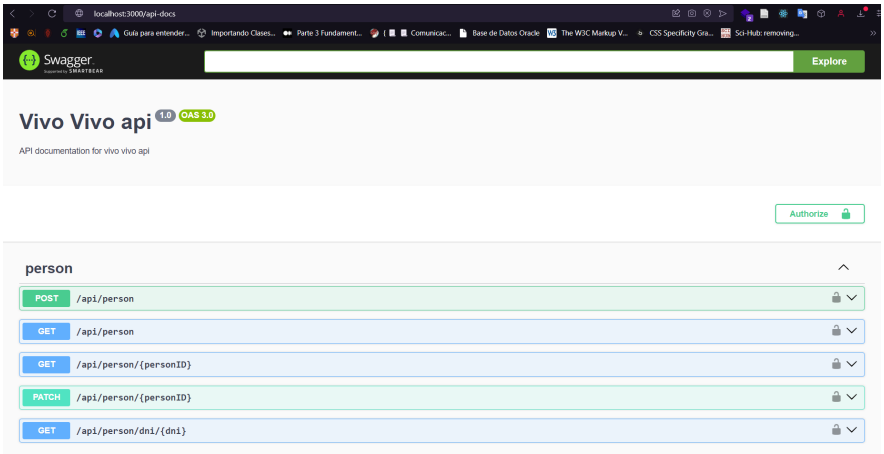


Figura 34. Documentación de la API con Swagger.

b. Configuración Interfaz de usuario (Frontend)

Sistema web

- ***Dependencias del sistema web.*** Para crear el proyecto de NextJS se utilizó el comando ”npx create-next-app@latest”, el cual crea una aplicación de NextJS con una estructura de carpetas y archivos predefinida. Las dependencias utilizadas en el sistema web se gestionaron mediante npm y el archivo de configuración package.json. En el Anexo V se muestra dependencias utilizadas en el sistema web.

- ***Configuración de variables de entorno del sistema web.*** Para la configuración de las variables de entorno del sistema web se utilizó un archivo .env.local, el cual contiene las propiedades de la aplicación, como la URL de la API, el api key de Google Maps, la

url y secret para la autenticación y la url para obtener las imágenes de Cloudinary. En el Anexo W se muestra el archivo .env.local con las variables de entorno del sistema web.

- ***Configuración de la aplicación.*** En Next.js, la configuración global de la aplicación se realiza mediante Providers y Contexts, los cuales permiten compartir datos y funcionalidades entre los componentes. En el Anexo X se muestra la configuración para los proveedores de sesión, componentes de UI, notificaciones y gestión de datos en el sistema web.

Las rutas de la aplicación en Next.js se crean mediante el gestor de archivos, el cual permite crear rutas dinámicas y estáticas siguiendo una estructura ”carpeta/page.ts”, donde ”carpeta” es el nombre de la ruta y ”page.ts” es el archivo de la página. En el Anexo Y se muestra la estructura de las rutas de la aplicación web.

Aplicación móvil

- ***Dependencias de la aplicación móvil.*** Para crear el proyecto de Flutter se utilizó el comando ”flutter create”, el cual crea una aplicación de Flutter con una estructura de carpetas y archivos predefinida. Las dependencias utilizadas en la aplicación móvil se gestionaron mediante flutter pub y el archivo de configuración pubspec.yaml. En el Anexo AB se muestra las dependencias utilizadas en la aplicación móvil.

- ***Configuración de variables de entorno de la aplicación móvil.*** Para la configuración de las variables de entorno de la aplicación móvil se utilizó un archivo .env, el cual contiene las propiedades de de la aplicación, como la URL de la API, el api key de Google Maps, el api key de OneSignal y la URL para obtener las imágenes de Cloudinary. En el Anexo AC se muestra el archivo .env con las variables de entorno de la aplicación móvil.

- ***Configuración de la aplicación.*** En Flutter, la configuración global de la aplicación se realiza mediante Providers y ChangeNotifier, los cuales permiten compartir datos y funcionalidades entre los widgets. En el Anexo AD se muestra la configuración para los proveedores de sesión, sockets, localización y gestión de datos en la aplicación móvil.

c. Iteración N°1

En base a las historias de usuario definidas en la iteración N°1 de la Tabla 34 se procedió a la implementación de las funcionalidades del sistema.

- ***Inicio de sesión sistema web.*** El inicio de sesión en el sistema web se realiza mediante un formulario en el cual el usuario ingresa su correo electrónico y contraseña, como se muestra en la Figura 35. Estos campos son validados mediante la librería Zod, la cual permite definir esquemas de validación para los datos de entrada en los formularios, como se puede observar en el Anexo Z. Una vez validados los datos, se envía una solicitud POST a la API para autenticar al usuario y obtener un token JWT, el cual se almacena en la sesión del navegador para mantener la sesión activa, en el Anexo AA se muestra el código para guardar el token en la sesión del navegador.



Figura 35. Inicio de sesión en el sistema web.

- ***Cerrar sesión sistema web.*** El cierre de sesión en el sistema web se realiza al hacer clic en la opción de cerrar sesión en el menú de usuario, como se muestra en la Figura 36. En este menú también se encuentra la opción de cambiar contraseña, la cual permite al usuario administrador cambiar su contraseña de acceso al sistema web.

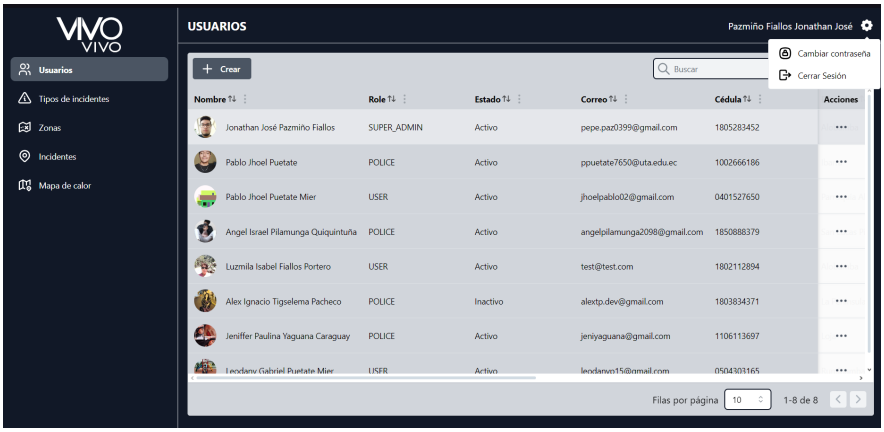


Figura 36. Menú de usuario en el sistema web.

- ***Gestión de usuarios.*** La gestión de usuarios en el sistema web se realiza mediante una tabla de entradas, la cual permite al usuario administrador visualizar los usuarios registrados en el sistema, como se muestra en la Figura 37. En esta tabla se muestran los campos de la información de los usuarios, como la fotografía, nombres, apellidos, correo electrónico, etnia, género, entre otros.

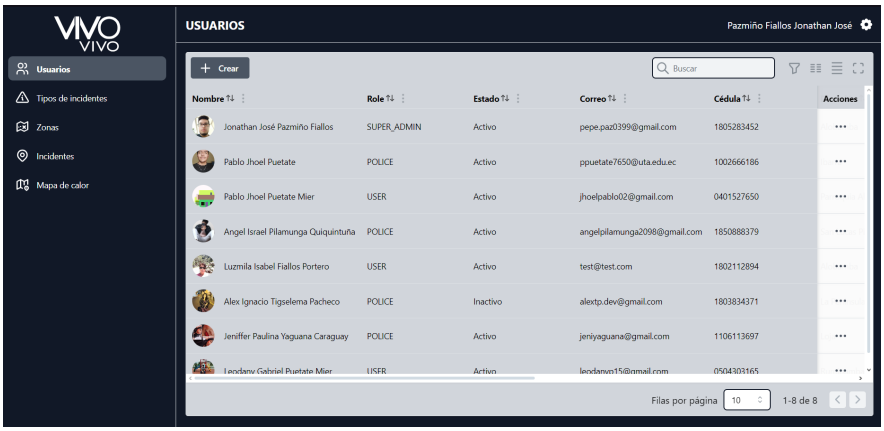


Figura 37. Tabla de usuarios en el sistema web.

El usuario administrador puede realizar acciones como crear, editar y deshabilitar usuarios, como se puede observar en la Figura 38.

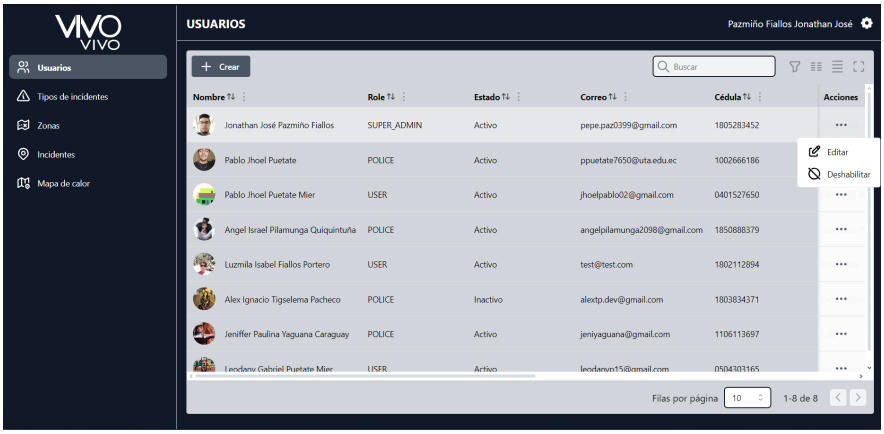


Figura 38. Menú de opciones de la tabla de usuarios en el sistema web.

El formulario de registro de usuarios permite al usuario administrador ingresar la información necesaria para crear un nuevo usuario, como se puede visualizar en las Figuras 39 y 40.

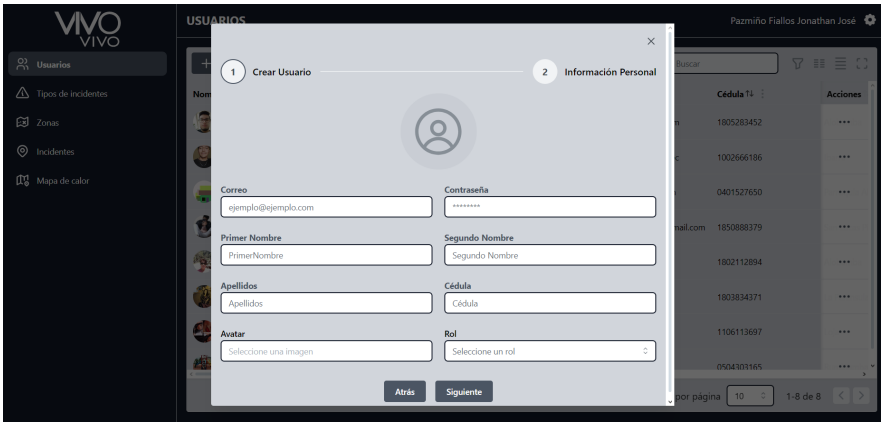


Figura 39. Formulario para crear/editar usuarios en el sistema web (Parte 1).

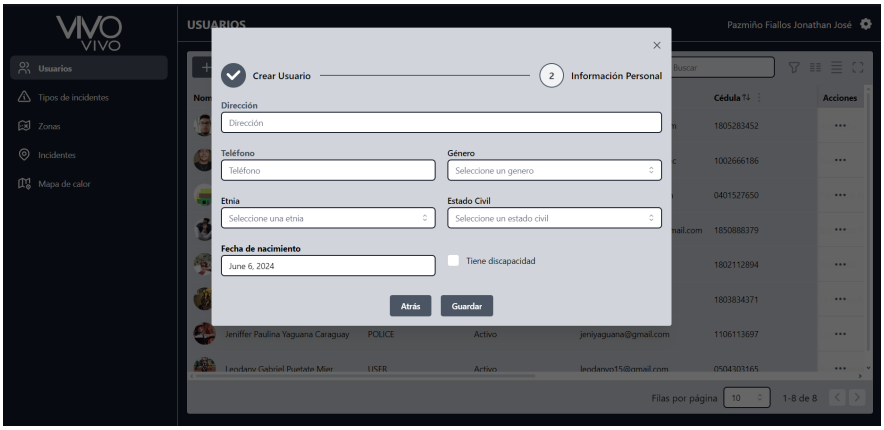


Figura 40. Formulario para crear/editar usuarios en el sistema web (Parte 2).

La tabla de gestión de usuarios cuenta con varios filtros de búsqueda, los cuales

permiten al usuario administrador buscar usuarios por columnas especificas o por un filtro general, como se muestra en la Figura 41.

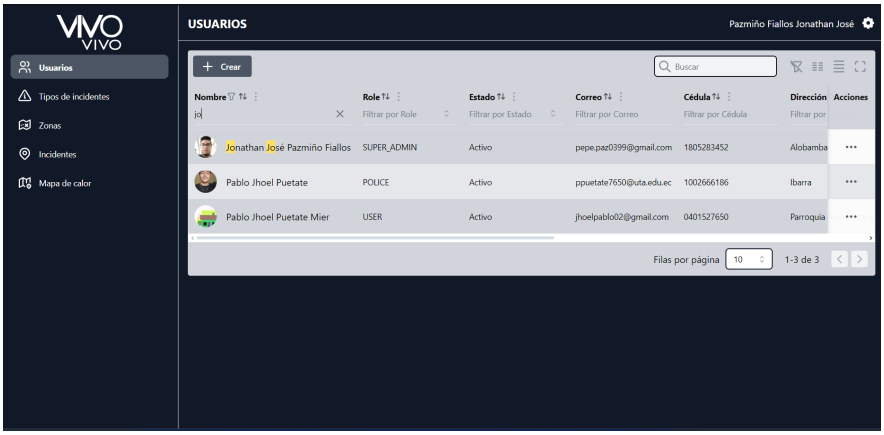


Figura 41. Filtros de búsqueda en la tabla de usuarios en el sistema web.

- ***Gestión de tipos de incidentes.*** La gestión de tipos de incidentes en el sistema web se realiza mediante una tabla de entradas, en la cual el usuario administrador puede visualizar los tipos de incidentes registrados en el sistema, como se muestra en la Figura 42. En esta tabla se muestran los campos de la información de los tipos de incidentes, como el nombre, descripción y estado.



Figura 42. Tabla de tipos de incidentes en el sistema web.

El usuario administrador puede realizar acciones como crear, editar y deshabilitar tipos de incidentes, como se puede observar en la Figura 43.

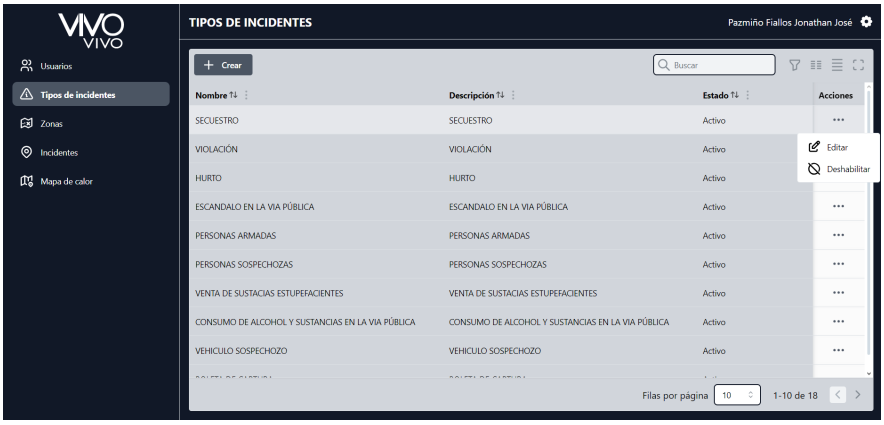


Figura 43. Menú de opciones de la tabla de tipos de incidentes en el sistema web.

El formulario de registro de tipos de incidentes permite al usuario administrador ingresar la información necesaria para crear un nuevo tipo de incidente, como se puede visualizar en la Figura 44.

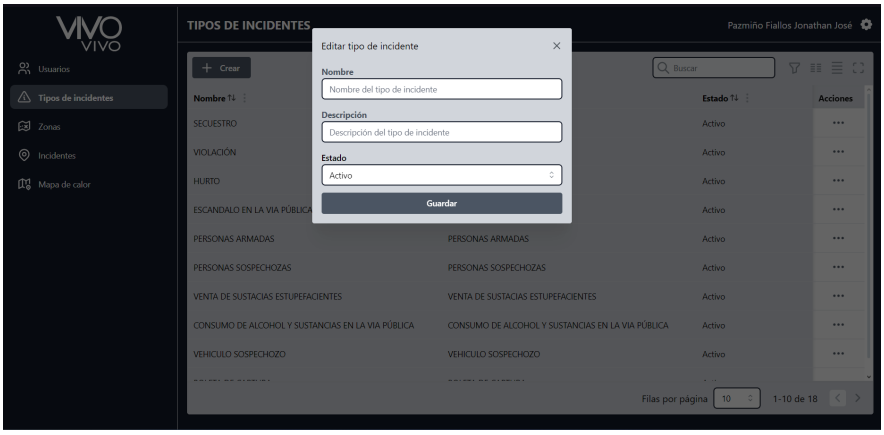


Figura 44. Formulario para crear/editar tipos de incidentes en el sistema web.

La tabla de gestión de tipos de incidentes cuenta con varios filtros de búsqueda, los cuales permiten al usuario administrador buscar tipos de incidentes por columnas especificas o por un filtro general, como se muestra en la Figura 45.

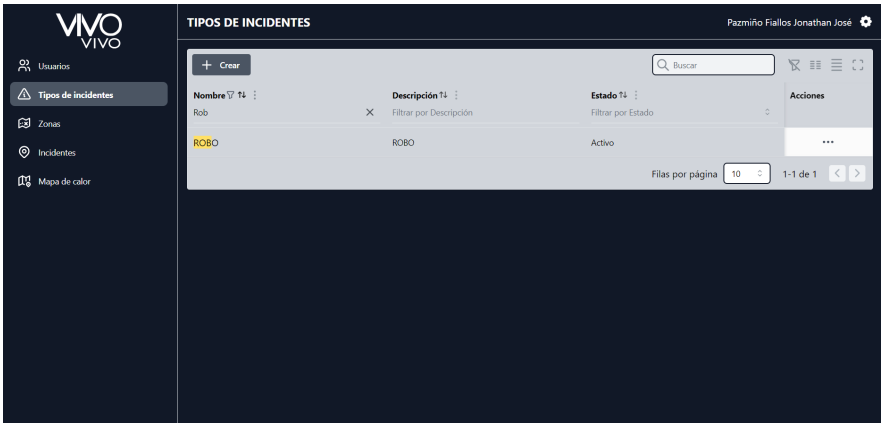


Figura 45. Filtros de búsqueda en la tabla de tipos de incidentes en el sistema web.

d. Iteración N°2

En base a las historias de usuario definidas en la iteración N°2 de la Tabla 36, se procedió a la implementación de las funcionalidades del sistema.

- ***Gestión de zonas de vigilancia.*** La gestión de zonas de vigilancia en el sistema web se realiza por medio de un mapa interactivo, en el cual el usuario administrador puede visualizar las zonas de vigilancia mediante polígonos, como se muestra en la Figura 46.

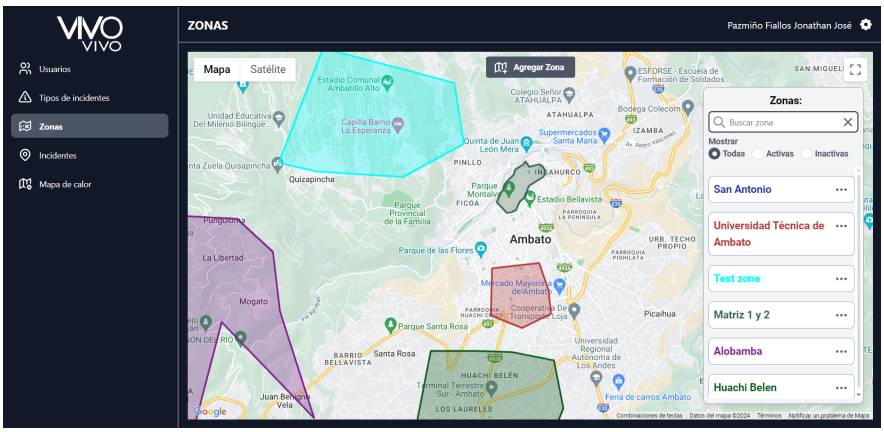


Figura 46. Mapa de zonas de vigilancia en el sistema web.

El usuario administrador puede realizar acciones como crear, editar y deshabilitar zonas de vigilancia, como se puede observar en la Figura 47.

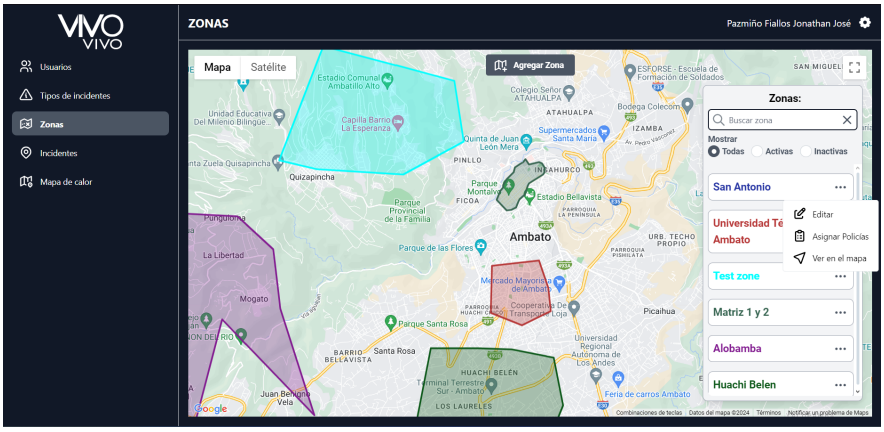


Figura 47. Menú de opciones del mapa de zonas de vigilancia en el sistema web.

El formulario de registro de zonas de vigilancia permite al usuario administrador dibujar un polígono en el mapa para definir una zona de vigilancia, como se puede visualizar en la Figura 48.

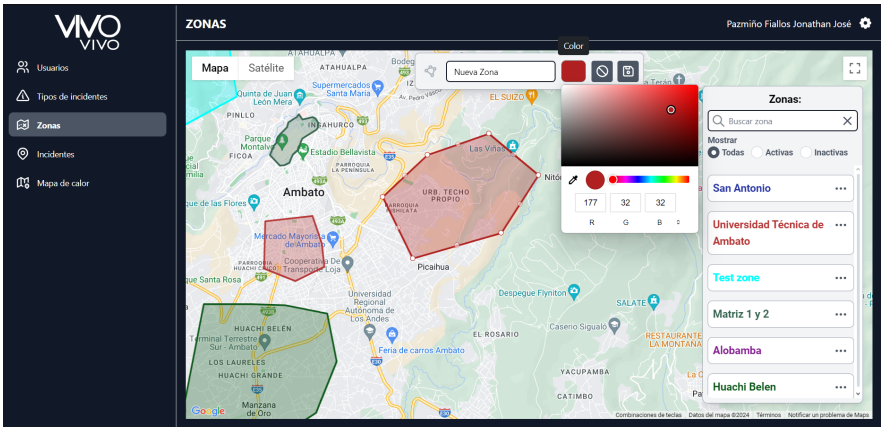


Figura 48. Formulario para crear/editar zonas de vigilancia en el sistema web.

- ***Asignar policías a las zonas de vigilancia.*** El usuario administrador podrá asignar policías a las zonas de vigilancia seleccionado la opción ”Asignar policías” en el menú de opciones de la zona de vigilancia, esto abrirá un modal en el cual el usuario administrador podrá seleccionar los policías que desea asignar a la zona de vigilancia, como se muestra en la Figura 49.

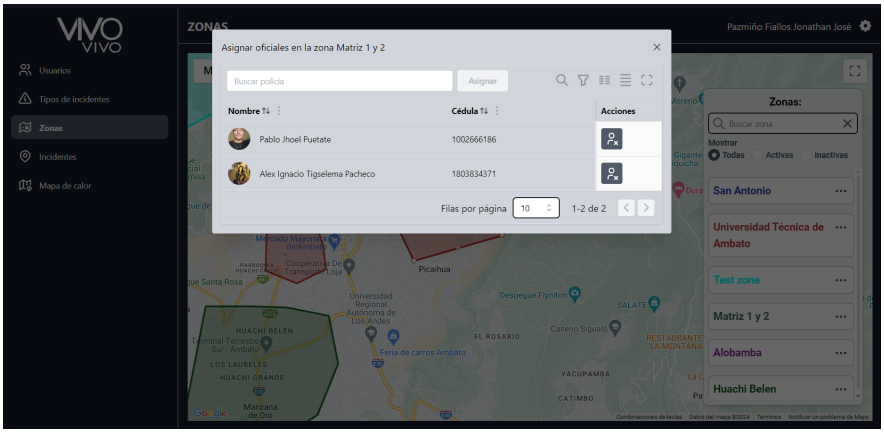


Figura 49. Asignar policías a zonas de vigilancia en el sistema web.

- ***Gestionar alertas de incidentes.*** La gestión de incidentes en el sistema web se realiza empleando un mapa interactivo, en el cual el usuario administrador puede visualizar los incidentes reportados junto con la ubicación en tiempo real de la víctima, así como la ubicación de los policías en la zona de emergencia, como se muestra en la Figura 50.

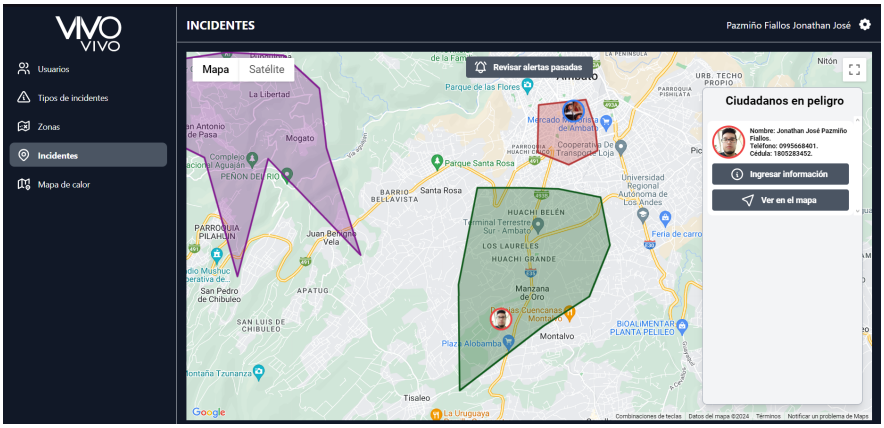


Figura 50. Mapa de incidentes en el sistema web.

El usuario administrador podrá visualizar el tipo de incidente y modificarlo en caso de ser necesario. Además, podrá visualizar información de la víctima y de los policías asignados, como se puede observar en la Figura 51.

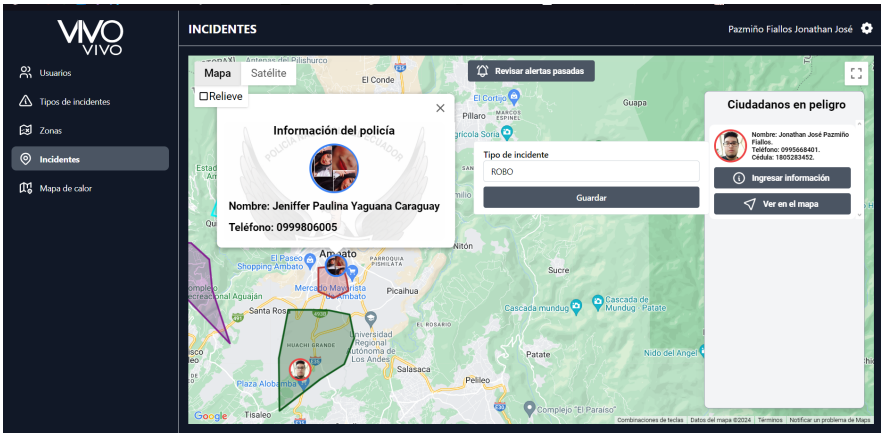


Figura 51. Detalles de incidente en el sistema web.

El usuario administrador podrá revisar las alertas de incidentes pasados mediante una tabla de entradas, en la cual se muestran los campos de la información de los incidentes, como el tipo de incidente, la fecha y la ubicación, como se muestra en la Figura 52.

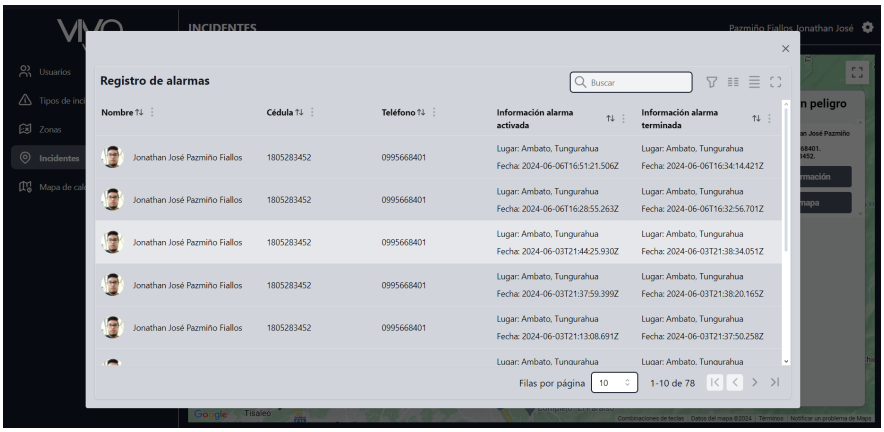


Figura 52. Tabla de incidentes en el sistema web.

- ***Mapa de calor.*** El mapa de calor en el sistema web permite al usuario administrador visualizar la densidad de incidentes reportados en un mapa mediante un gradiente de colores, así como filtrar los incidentes por tipo y fecha, como se muestra en la Figura 53.

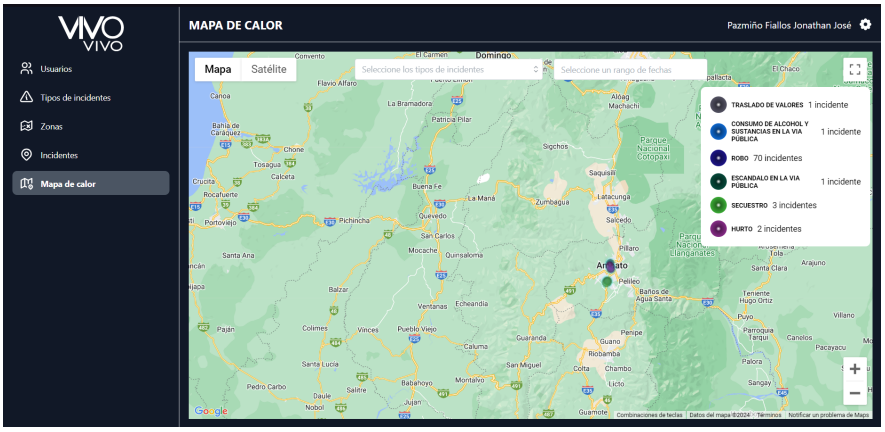


Figura 53. Mapa de calor en el sistema web.

e. Iteración N°3

En base a las historias de usuario definidas en la iteración N°3 de la Tabla 38, se procedió a la implementación de las funcionalidades del sistema.

- ***Inicio de sesión aplicación móvil.*** El inicio de sesión en la aplicación móvil se realiza mediante un formulario en el cual el usuario ingresa su correo electrónico y contraseña, como se muestra en la Figura 54. Estas credenciales son enviadas a la API mediante una solicitud POST para autenticar al usuario y obtener un token JWT, el cual se almacena en el almacenamiento local del dispositivo para mantener la sesión activa, en el Anexo AE se muestra el código para guardar el token en el almacenamiento local del dispositivo.

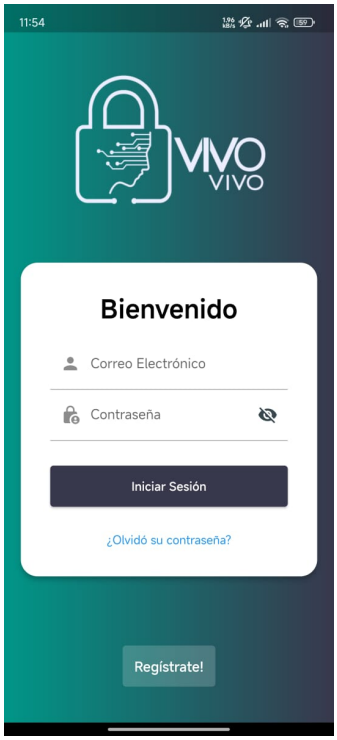


Figura 54. Inicio de sesión en la aplicación móvil.

- ***Cerrar sesión aplicación móvil.*** El cierre de sesión en la aplicación móvil se realiza al hacer clic en la opción de cerrar sesión en el menú de usuario, como se muestra en la Figura 55. En este menú también se encuentra la opción de cambiar contraseña, la cual permite al usuario cambiar su contraseña de acceso a la aplicación móvil y la opción de gestionar el grupo familiar.

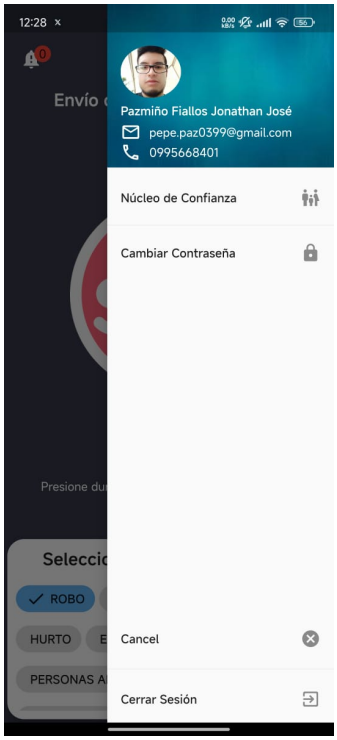


Figura 55. Menú de usuario en la aplicación móvil.

- ***Registro de usuario.*** El registro de usuario en la aplicación móvil se realiza mediante un formulario en el cual el usuario ingresa su información personal, como nombres, apellidos, correo electrónico, contraseña, género, etnia, entre otros, como se muestra en las Figuras 56 y 57.

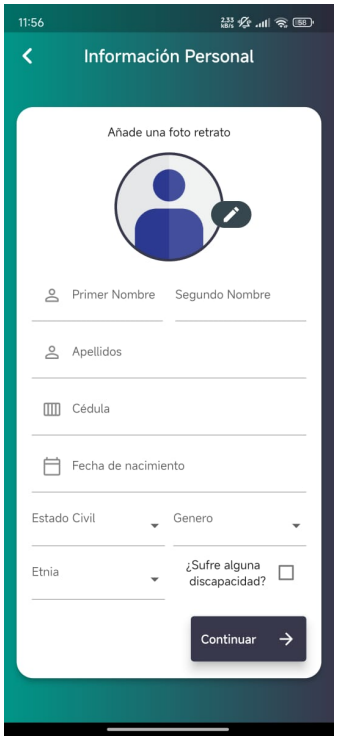


Figura 56. Registro de usuario en la aplicación móvil (Parte 1).

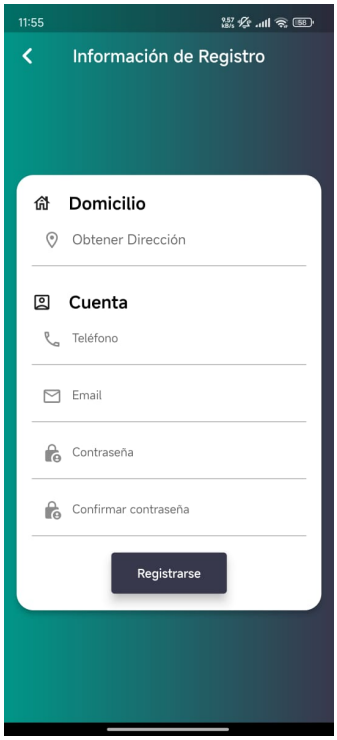


Figura 57. Registro de usuario en la aplicación móvil (Parte 2).

Para ingresar la dirección del usuario, se utilizó un campo de búsqueda de direcciones que permite al usuario buscar su dirección en un mapa interactivo y seleccionarla, como se muestra en la Figura 58.

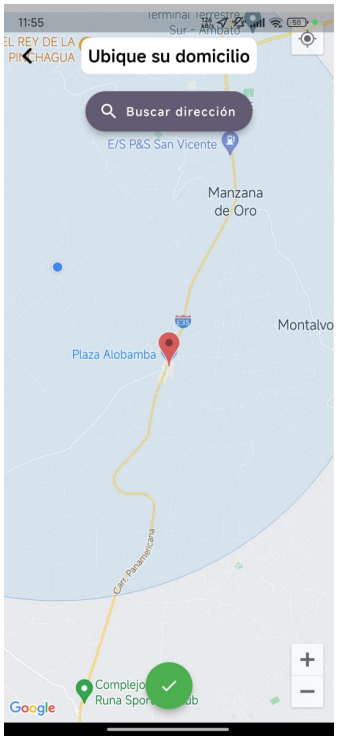


Figura 58. Registro de usuario en la aplicación móvil (Parte 3).

f. Iteración N°4

En base a las historias de usuario definidas en la iteración N°4 de la Tabla 40, se procedió a la implementación de las funcionalidades del sistema.

- ***Gestión de grupo familiar.*** La gestión del grupo familiar en la aplicación móvil se realiza mediante una lista en la cual el usuario podrá visualizar a los miembros de su grupo familiar, como se muestra en la Figura 59.

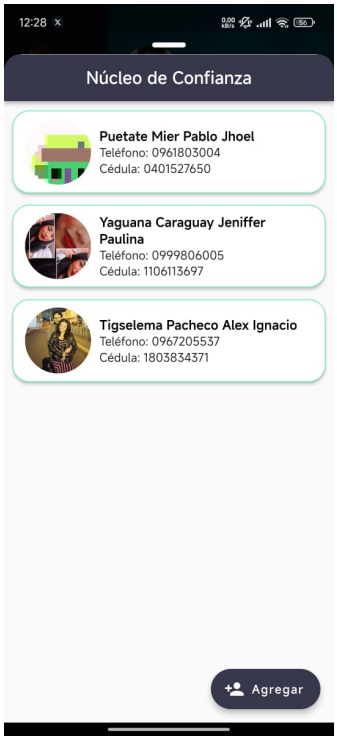


Figura 59. Gestión de grupo familiar en la aplicación móvil.

El usuario podrá agregar miembros a su grupo familiar mediante un formulario en el cual ingresará la cédula de identidad del usuario que desee agregar, como se muestra en la Figura 60.

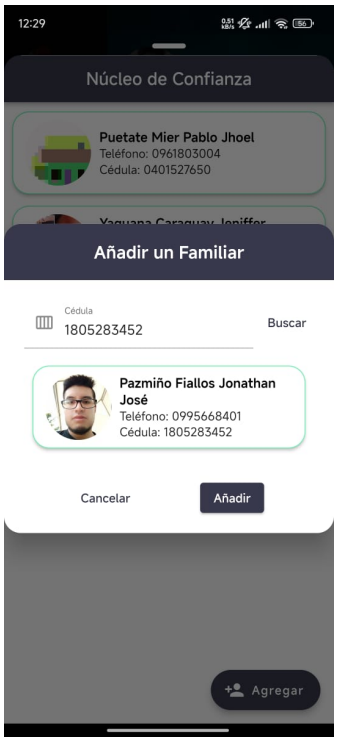


Figura 60. Agregar miembro al grupo familiar en la aplicación móvil.

- ***Envió de alertas de emergencia.*** En la pantalla principal de la aplicación móvil muestra el botón de pánico en la parte central de la pantalla, el cual permite al usuario enviar una alerta de emergencia a los miembros de su grupo familiar y a los policías en la zona de emergencia presionando el botón durante 3 segundos, también el usuario puede seleccionar el tipo de incidente mediante un check, como se muestra en la Figura 61.



Figura 61. Pantalla principal de la aplicación móvil.

- ***Visualizar alertas de emergencia de familiares.*** Las alertas recibidas en la aplicación móvil se visualizan mediante una lista en la cual se muestra el estado de cada miembro del grupo familiar. Cuando un miembro del grupo familiar envía una alerta de emergencia, se muestra una notificación en el dispositivo del usuario, como se puede observar en la Figura 62.

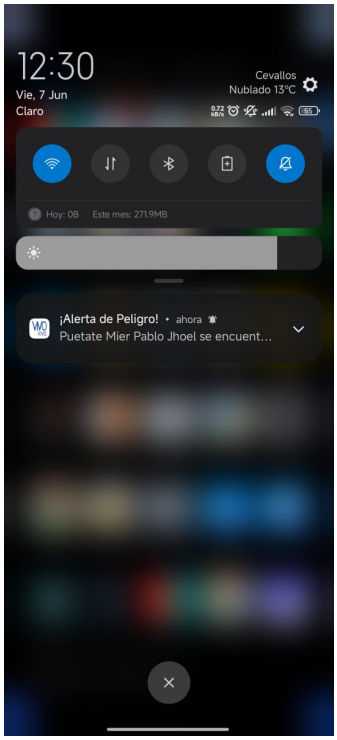


Figura 62. Notificación de alerta recibida en la aplicación móvil.

Una vez enviada la alerta el estado del miembro del grupo familiar cambia a ”En peligro”, como se puede observar en la Figura 63.



Figura 63. Alerta recibida en la aplicación móvil.

El usuario podrá visualizar la ubicación en tiempo real del miembro en peligro seleccionando el botón ”Seguir ubicación”, lo cual mostrará la en un mapa la ubicación del miembro en peligro y la ubicación del usuario junto al radio de error obtenido mediante el análisis de la precisión de la ubicación realizado en la Sección 3.3, como se muestra en la Figura 64.

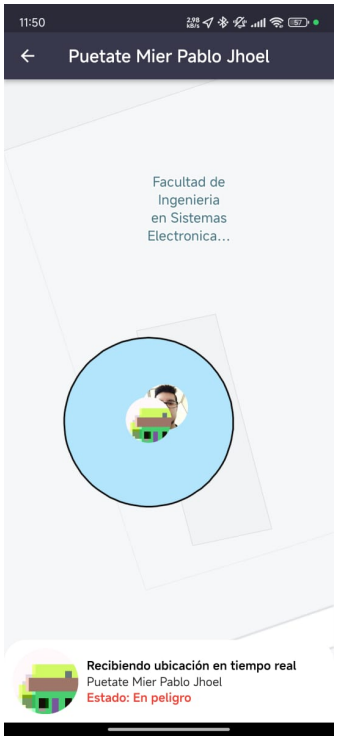


Figura 64. Seguir ubicación en la aplicación móvil.

- ***Visualizar alertas de emergencia de los ciudadanos.*** Las alertas recibidas en la aplicación móvil del policía se visualizan mediante una lista en la cual se muestra el estado de cada ciudadano que envió una alerta de emergencia. Cuando un ciudadano envía una alerta de emergencia, se muestra una notificación en el dispositivo del policía, como se puede observar en la Figura 65.

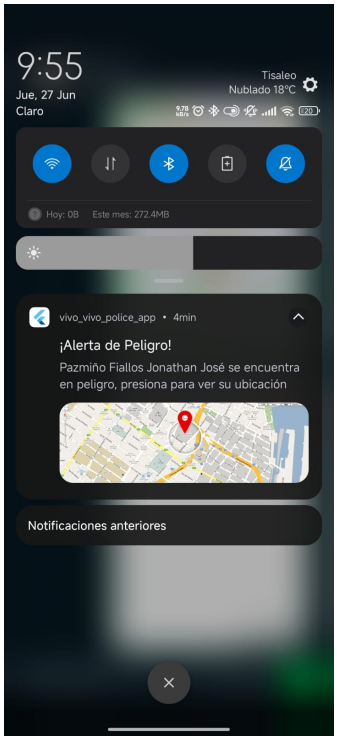


Figura 65. Notificación de alerta recibida en la aplicación móvil del policía.

Una vez enviada la alerta el estado del ciudadano cambia a ”En peligro”, como se puede observar en la Figura 66.

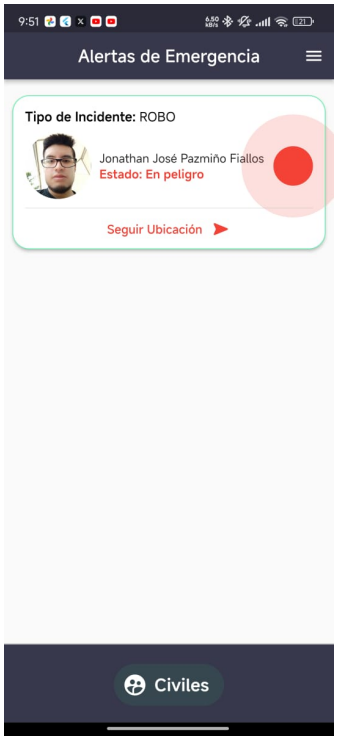


Figura 66. Alerta recibida en la aplicación móvil del policía.

El policía podrá visualizar la ubicación en tiempo real del ciudadano en peligro seleccionando el botón ”Seguir ubicación”, lo cual mostrará la en un mapa la ubicación del ciudadano en peligro y la ubicación del policía junto al radio de error obtenido mediante el análisis de la precisión de la ubicación realizado en la Sección 3.3, como se muestra en la Figura 67.

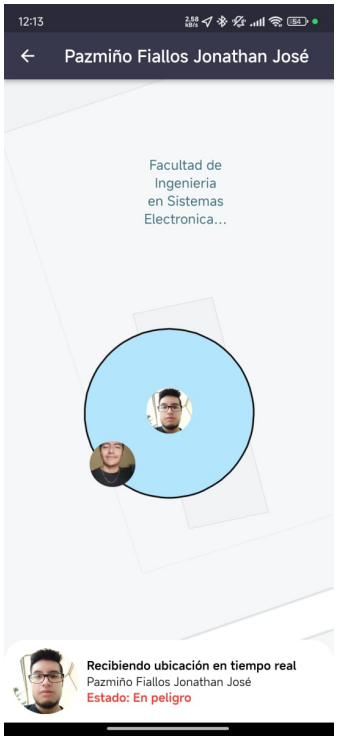


Figura 67. Seguir ubicación en la aplicación móvil del policía.

g. Iteración N°5

En base a las historias de usuario definidas en la iteración N°5 de la Tabla 42, se procedió a la implementación de las funcionalidades del sistema.

- ***Cambiar contraseña.*** La opción de cambiar contraseña en la aplicación móvil permite al usuario modificar su contraseña ingresando la contraseña actual, la nueva contraseña y la confirmación de la nueva contraseña, como se muestra en la Figura 68.



Figura 68. Cambiar contraseña en la aplicación móvil.

- ***Recuperar contraseña.*** La recuperación de contraseña en la aplicación móvil se realiza mediante un formulario en el cual el usuario ingresa su correo electrónico, como se muestra en la Figura 69.

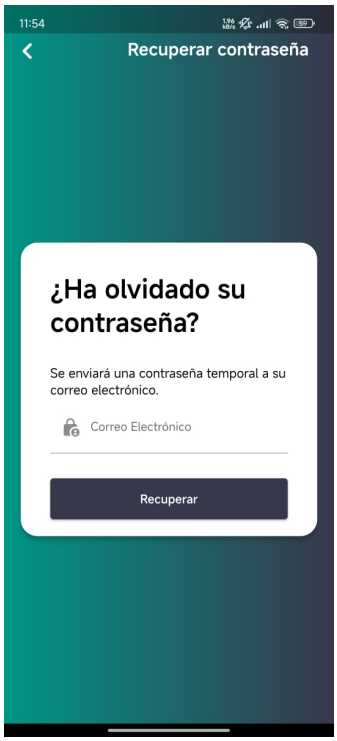


Figura 69. Recuperar contraseña en la aplicación móvil.

Una vez ingresado el correo electrónico, se envía una contraseña provisional al correo electrónico del usuario con la cual podrá iniciar sesión y cambiar su contraseña, en la Figura 70 se muestra el correo electrónico de recuperación de contraseña.

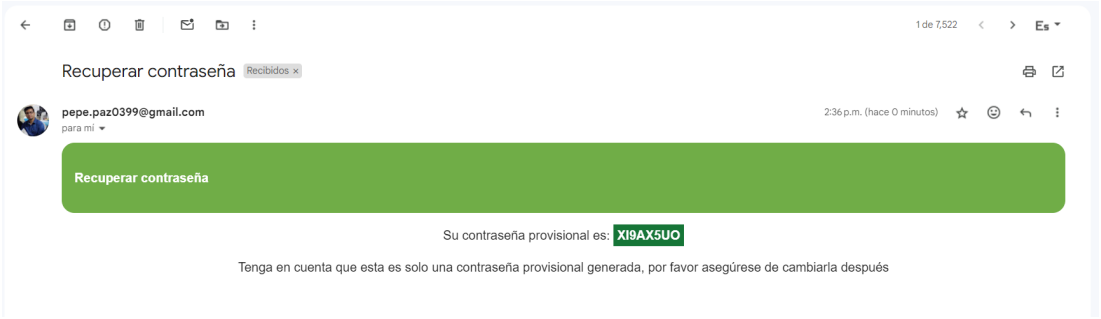


Figura 70. Correo electrónico de recuperación de contraseña.

Modelo analítico de BI

Para el diseño del modelo analítico de BI se utilizó la metodología Hefesto, ya que esta facilita la construcción de un Data Warehouse y aporta información útil para mejorar el rendimiento [35].

La carta de diseño para el modelo dimensional se utiliza para describir el proceso de diseño de un modelo dimensional siguiendo los pasos de la metodología Hefesto. Estos pasos incluyen la selección del proceso de negocio, la declaración del grano, la identificación de las dimensiones y sus atributos, la identificación de las tablas de hechos y sus métricas, y el diseño del esquema dimensional. A continuación, se presenta cada uno de estos pasos.

- Seleccionar el proceso de negocio:
  - Identificar las pregustas del negocio.
  - Identificar indicadores y perspectivas.
  - Diseñar el modelo conceptual.
- Declarar el grano.
- Identificar las dimensiones y sus atributos.
- Identificar las tablas de hechos y sus métricas.
- Diseñar el esquema dimensional

- ***Seleccionar el proceso de negocio.***

- ***Preguntas del negocio.*** El objetivo principal del sistema de reportería de incidentes delictivos es recolectar y analizar información sobre los incidentes delictivos reportados a través de un sistema de alarma, así como detalles de los usuarios y datos geográficos relevantes para apoyar la toma de decisiones de las autoridades policiales.

- ***Identificar indicadores y perspectivas.***

  - Determinar el número de incidentes por tipo de incidente y zona de vigilancia.
  - Analizar la distribución de incidentes por género, discapacidad, etnia y edad de los usuarios.
  - Identificar las zonas con mayor número de incidentes para optimizar la vigilancia policial.

Tabla 62. Indicadores y perspectivas en base las pregustas de negocio

Indicadores	Perspectivas
Número de incidentes	Tipo de Alarma
Número de incidentes	Tipo de Incidente
Número de incidentes	Zona de Vigilancia
Número de incidentes	Género, Discapacidad, Etnia, Estado Civil

En base a los indicadores y perspectivas identificados, se estableció como KPI principal el número de incidentes delictivos reportados, ya que este indicador permite medir la frecuencia de los delitos reportados en una determinada área o periodo de tiempo.

- ***Declarar el grano.*** El grano del modelo dimensional es un único incidente delictivo reportado. Esto implica que cada registro en la tabla de hechos representa un incidente específico.

- ***Identificar las dimensiones y sus atributos.*** A continuación, se presentan las dimensiones y sus atributos identificados para el modelo dimensional.

En la Tabla 63 se presenta los atributos seleccionados para la dimensión de tiempo.

Tabla 63. Dimensión de tiempo con sus atributos

Campo	Descripción
fechaID	Identificador único para cada fecha
fecha	Fecha completa del incidente (YYYY-MM-DD)
anio	Año en que ocurrió el incidente
mes	Mes en que ocurrió el incidente
día	Día en que ocurrió el incidente
trimestre	Trimestre en que ocurrió el incidente
semestre	Semestre en que ocurrió el incidente
hora	Hora en que ocurrió el incidente

En la Tabla 64 se presenta los atributos seleccionados para la dimensión de usuarios.

Tabla 64. Dimensión de usuarios con sus atributos

Campo	Descripción
usuarioID	Identificador único del usuario
género	Género del usuario
discapacidad	Estado de discapacidad del usuario
etnia	Etnia del usuario
estadoCivil	Estado civil del usuario
edad	Edad del usuario

En la Tabla 65 se presenta los atributos seleccionados para la dimensión de tipo de alarma.

Tabla 65. Dimensión de tipo de alarma con sus atributos

Campo	Descripción
tipoAlarmaID	Identificador único del tipo de alarma
nombreTipoAlarma	Nombre del tipo de alarma

En la Tabla 66 se presenta los atributos seleccionados para la dimensión de tipo de incidente.

Tabla 66. Dimensión de tipo de incidente con sus atributos

Campo	Descripción
tipoIncidenteID	Identificador único del tipo de incidente
nombreTipoIncidente	Nombre del tipo de incidente

En la Tabla 67 se presenta los atributos seleccionados para la dimensión de zonas de vigilancia.

Tabla 67. Dimensión de zonas de vigilancia con sus atributos

Campo	Descripción
zonaVigilanciaID	Identificador único de la zona de vigilancia
polígono	Representación geográfica de la zona de vigilancia
nombreZonaVigilancia	Nombre de la zona de vigilancia

En la Tabla 68 se presenta los atributos seleccionados para la dimensión de ubicación.

Tabla 68. Dimensión de ubicación con sus atributos

Campo	Descripción
ubicacionID	Identificador único de la ubicación
Continua en la siguiente página	

Tabla 68 – continuación de la página anterior

Campo	Descripción
canton	Cantón del lugar del incidente
ciudad	Ciudad del lugar del incidente

- ***Identificar las tablas de hechos y sus métricas.*** En la Tabla 69 se presentan los hechos de incidentes delictivos y sus atributos.

Tabla 69. Hechos de incidentes delictivos con sus atributos

Campo	Descripción
incidenteID	Clave primaria de la tabla de hechos incidentes delictivos
fechaID	Clave foránea a la dimensión de tiempo
usuarioID	Clave foránea a la dimensión de usuarios
tipoAlarmaID	Clave foránea a la dimensión de tipo de alarma
tipoIncidenteID	Clave foránea a la dimensión de tipo de incidente
zonaVigilanciaID	Clave foránea a la dimensión de zonas de vigilancia
ubicacionID	Clave foránea a la dimensión de ubicación geográfica
numeroIncidentes	Número de incidentes reportados

- ***Esquema dimensional.*** En la Figura 71 se muestra el esquema del modelo dimensional propuesto para el sistema de reportería de incidentes delictivos.

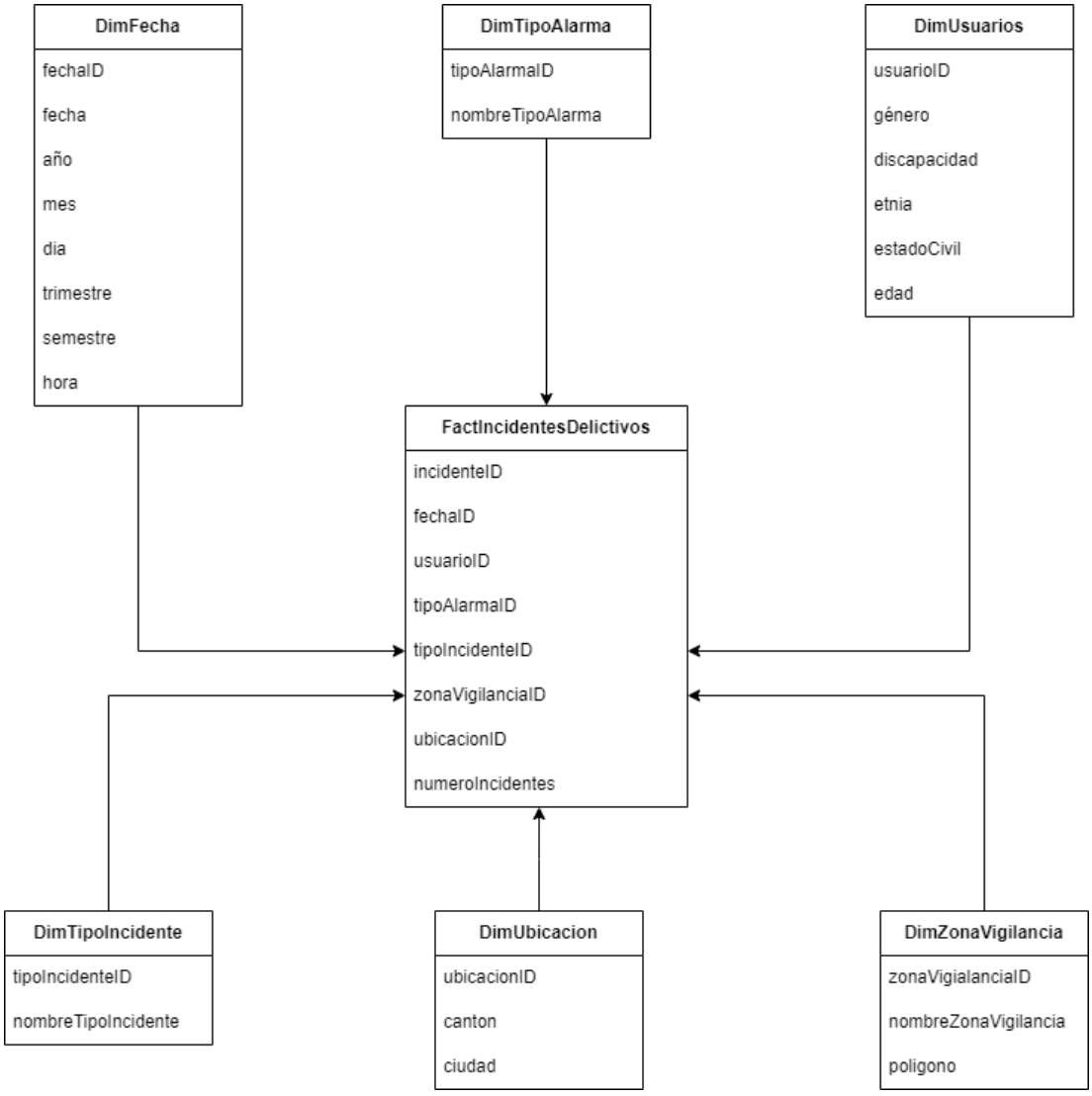


Figura 71. Esquema del modelo dimensional para el sistema de reportería de incidentes delictivos

- ***Proceso ETL.*** Una vez definido el modelo dimensional, se procedió a diseñar el proceso de extracción, transformación y carga (ETL) de los datos. Para ello, se utilizó la herramienta de ETL de visual studio, la cual permite extraer datos de diferentes fuentes, transformarlos y cargarlos en el modelo dimensional. En la Figura 72 se muestra el diseño del proceso ETL para el modelo dimensional.

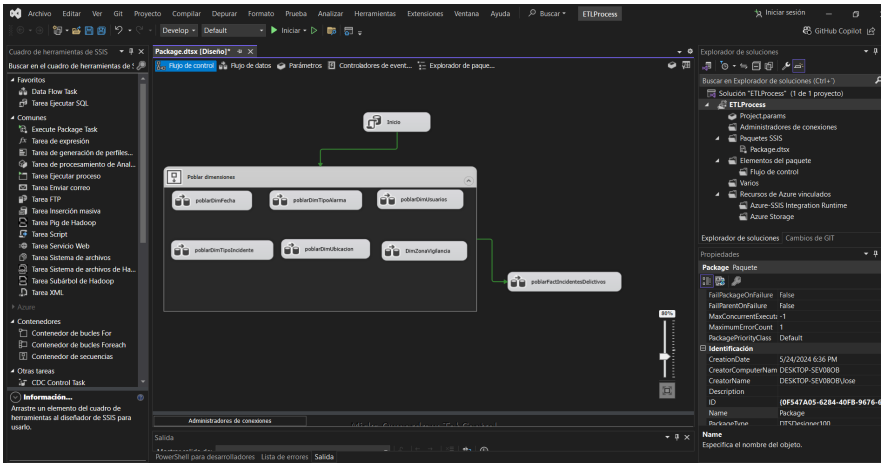


Figura 72. Diseño del proceso ETL para el modelo dimensional.

El proceso ETL inicia con la limpieza de las tablas que componen el modelo dimensional. Esto se realiza mediante la ejecución de un script SQL que elimina los registros de las tablas de hechos y dimensiones, como se puede observar en la Figura73. Una vez limpias las tablas, se procede a extraer los datos de las fuentes de datos. En este caso, se utilizó un origen de datos de ADO.NET para extraer los datos desde PostgreSQL. Además, se utilizó el componente de Data Conversion para convertir los datos a un formato compatible con el destino en una base de datos de SQL Server. Finalmente, se cargan los datos mediante el componente de OLE DB Destination, como se puede observar en la Figura 74.

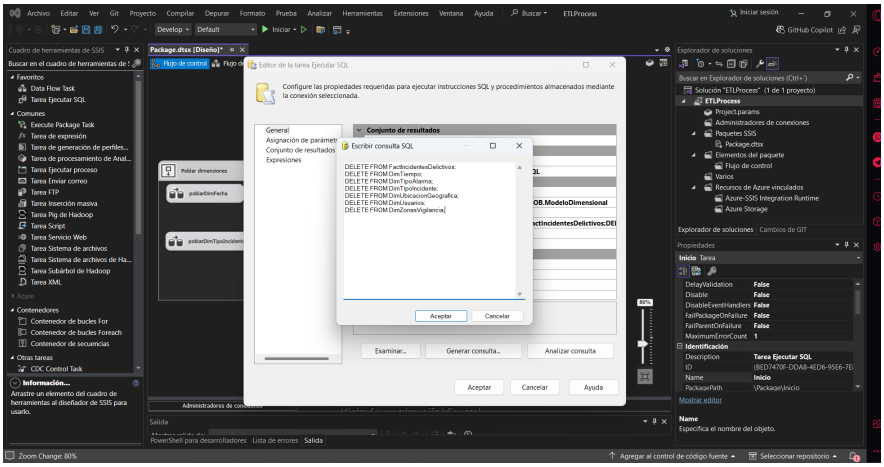


Figura 73. Limpieza de las tablas del modelo dimensional.

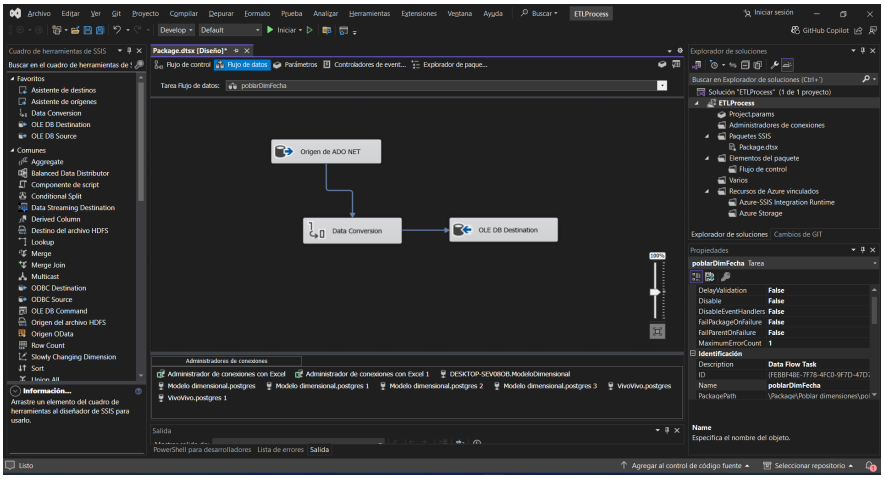


Figura 74. Extracción de datos para el modelo dimensional.

Al ejecutar el proceso ETL, se comienza con la extracción siguiendo el flujo de datos definido en el proceso ETL. En caso de no existir ningún error, el sistema mostrará un mensaje de □□xito, como se puede observar en la Figura 75.

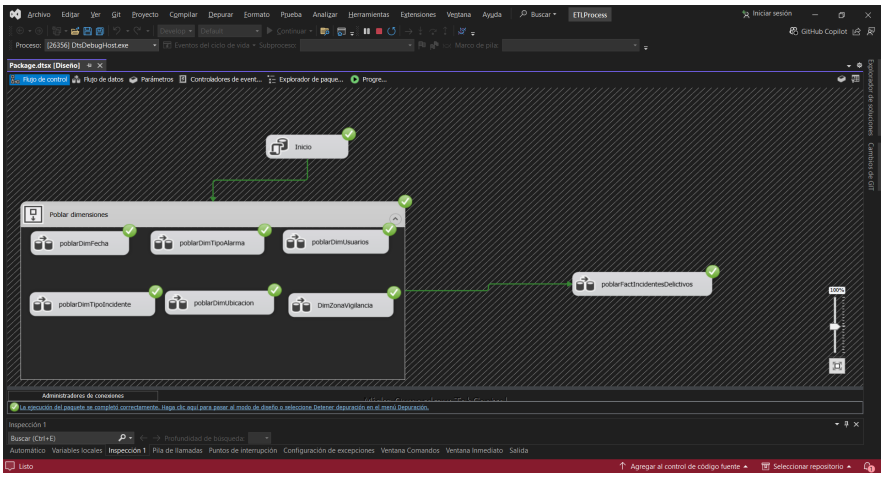


Figura 75. Mensaje de éxito al ejecutar el proceso ETL.

- ***Cubo OLAP.*** Para la construcción del cubo OLAP se empleó Visual Studio Community 2022, creando un proyecto de Analysis Services.

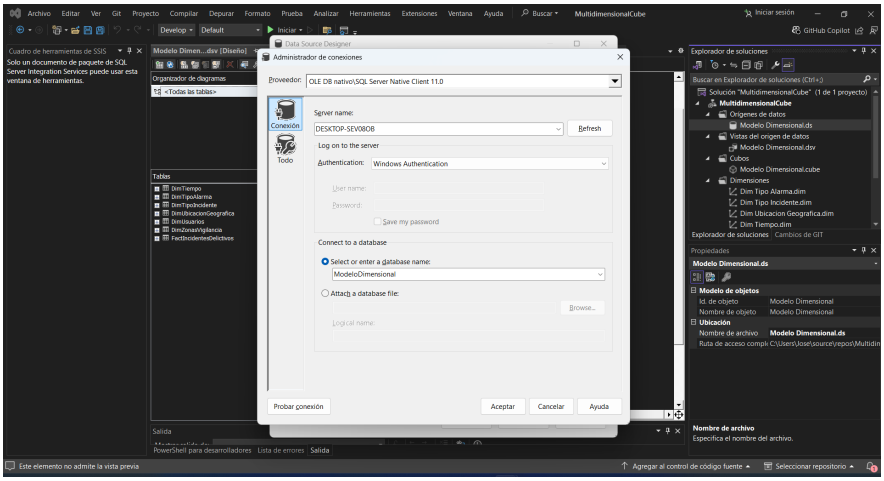


Figura 76. Conexión a la base de datos de SQL Server en el proyecto de Analysis Services.

En este proyecto, se definió la conexión a la base de datos de SQL Server, como se muestra en la Figura 76, se creó un origen de datos, como se puede observar en la Figura 77.

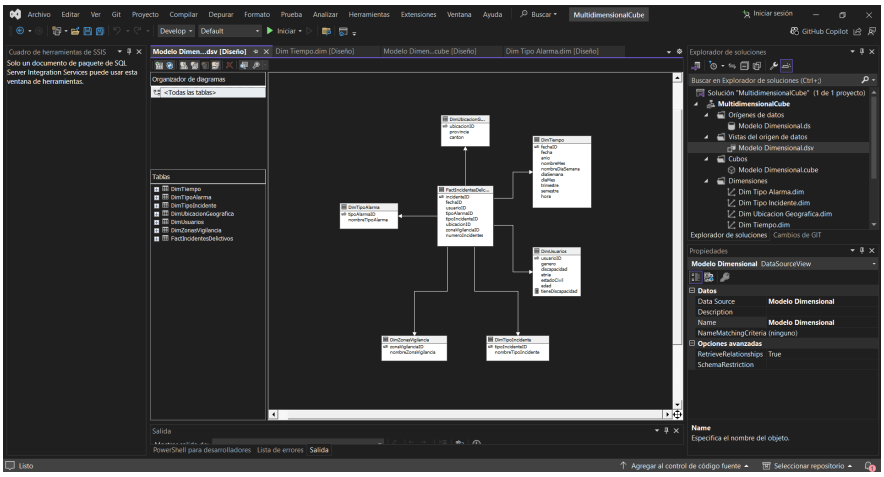


Figura 77. Origen de datos en el proyecto de Analysis Services.

Una vez configurada la conexión a la base de datos, se procedió a diseñar el cubo OLAP, definiendo las medidas e indicadores del modelo dimensional, como se puede observar en la Figura 78.

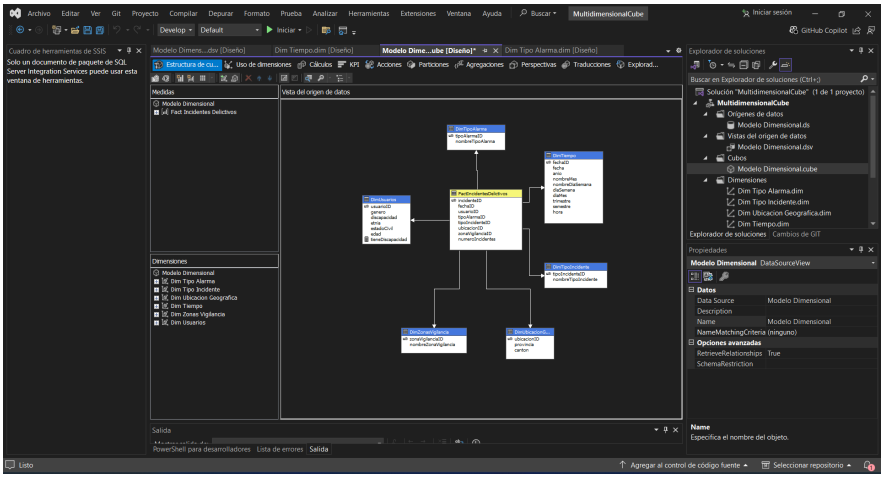


Figura 78. Diseño del cubo OLAP en el proyecto de Analysis Services.

En el origen de datos se definieron campos creados mediante cálculos con nombre que permiten agregar medidas e información adicional al cubo OLAP, como se puede observar en la Figura 79.

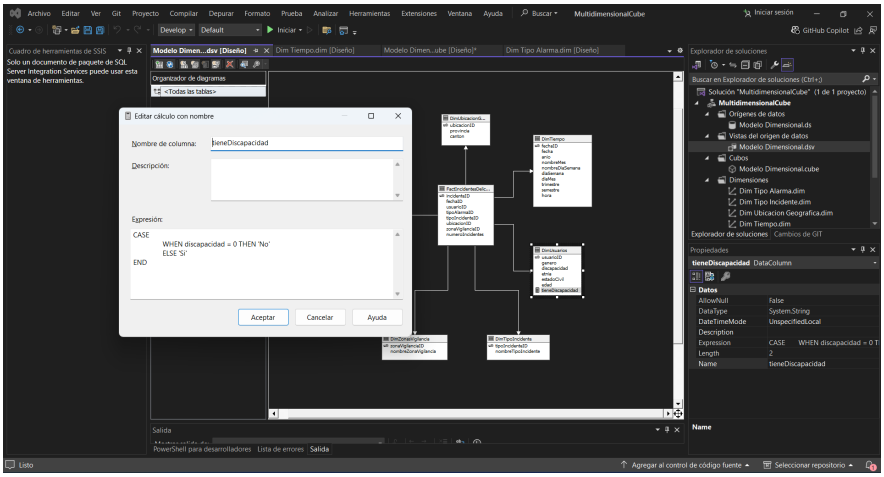


Figura 79. Campos en el origen de datos del proyecto de Analysis Services.

En la dimensión de tiempo se definieron jerarquías de tiempo que permiten visualizar los datos de forma secuencial en años, meses, días y horas, como se puede observar en la Figura 80.

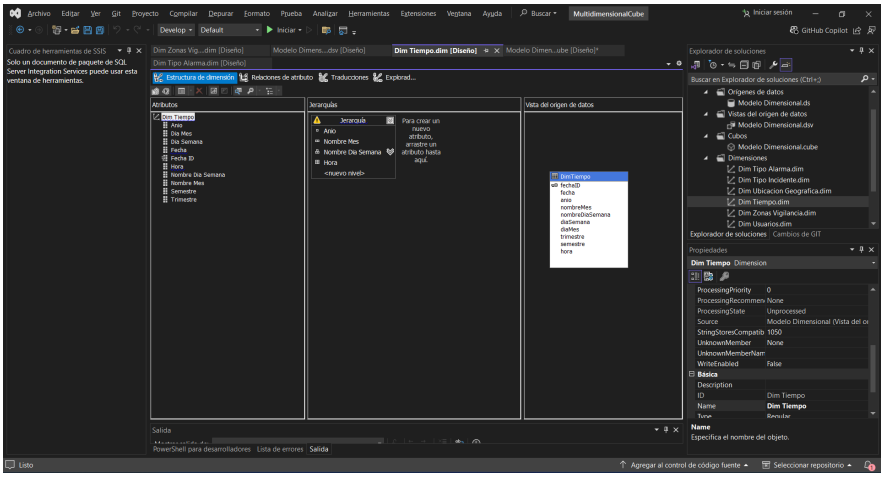


Figura 80. Jerarquías de tiempo en la dimensión de tiempo del proyecto de Analysis Services.

Una vez definido el cubo OLAP, se procedió a procesarlo para generar las medidas e indicadores definidos en el modelo dimensional, terminado el proceso de procesamiento se muestra un mensaje de éxito y se muestra el examinador de cubos en el cual se pueden realizar consultas a los datos del cubo OLAP, como se puede observar en la Figura 81.

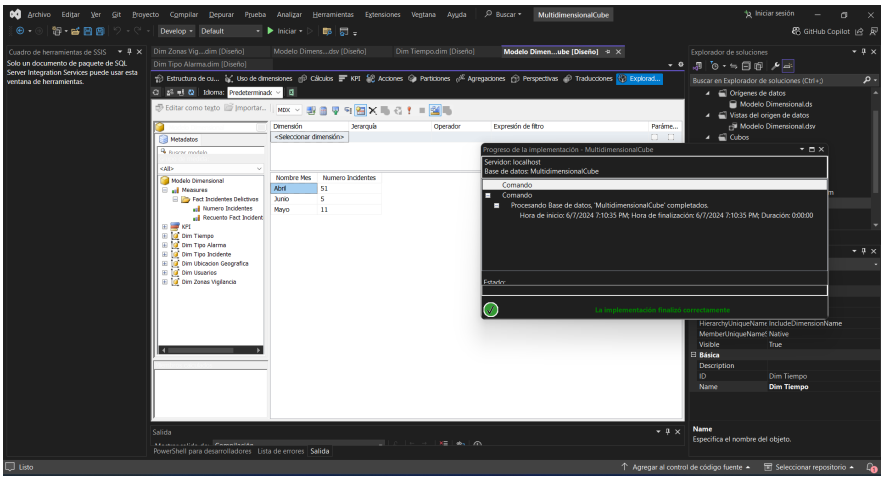


Figura 81. Mensaje de éxito al procesar el cubo OLAP.

- ***Visualización de datos.*** Para la visualización de datos se utilizó Power BI, el cual permite conectar a diferentes fuentes de datos, como SQL Server, Analysis Services, Excel, entre otros. En este caso se lo conecto al cubo OLAP de Analysis Services, como se muestra en la Figura 82. Una vez realizada la conexión En Power BI se creó un informe en el cual se visualizan los datos del cubo mediante gráficos, tablas y mapas de calor que permiten al usuario analizar la información de forma interactiva, como se

muestra en la Figura 83.

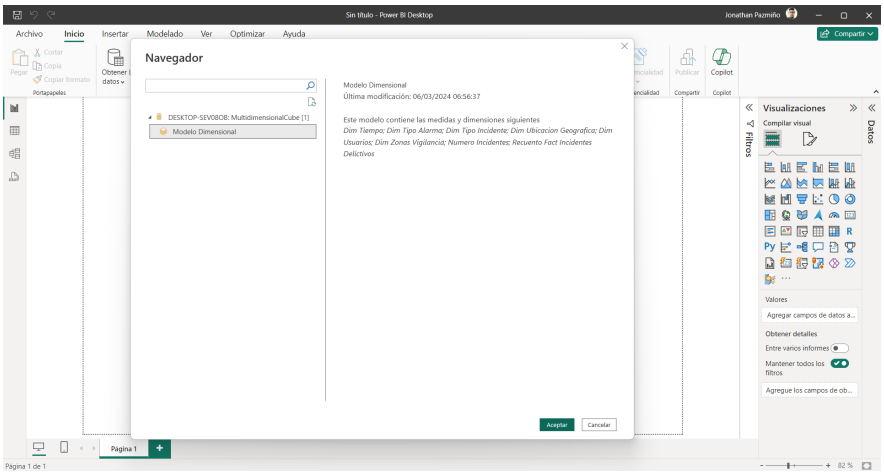


Figura 82. Conexión al cubo OLAP de Analysis Services en Power BI.

Se generaron dos tipos de informes en Power BI, el primero representa la información general de los incidentes reportados, con información que muestra el número de incidentes por tipo de incidente, zona de vigilancia, género, discapacidad, etnia y edad de los usuarios, como se puede observar en la Figura 83.

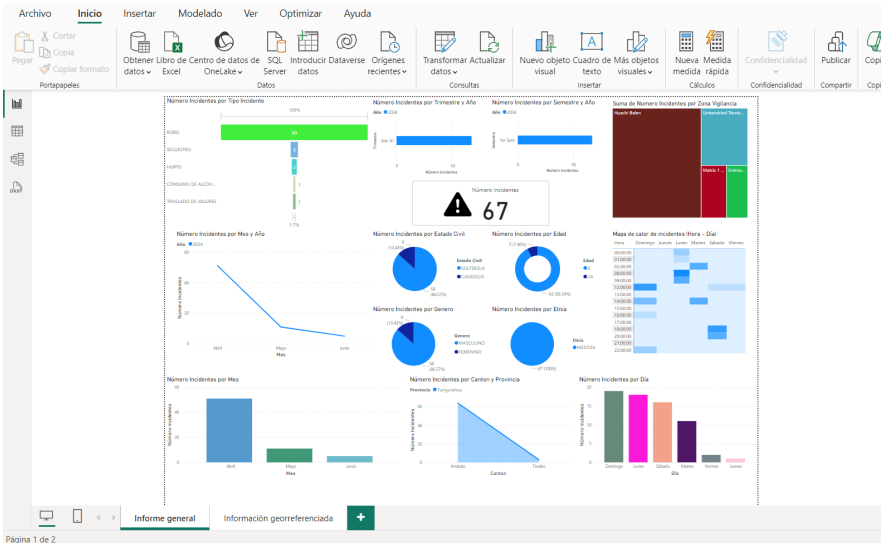


Figura 83. Informe en Power BI con los datos del cubo OLAP.

Al realizar click sobre un gráfico, tabla o mapa, los demás elementos del informe se actualizan automáticamente para mostrar la información relacionada con el elemento seleccionado, permitiendo al usuario filtrar los datos de forma interactiva, como se puede observar en la Figura 84.

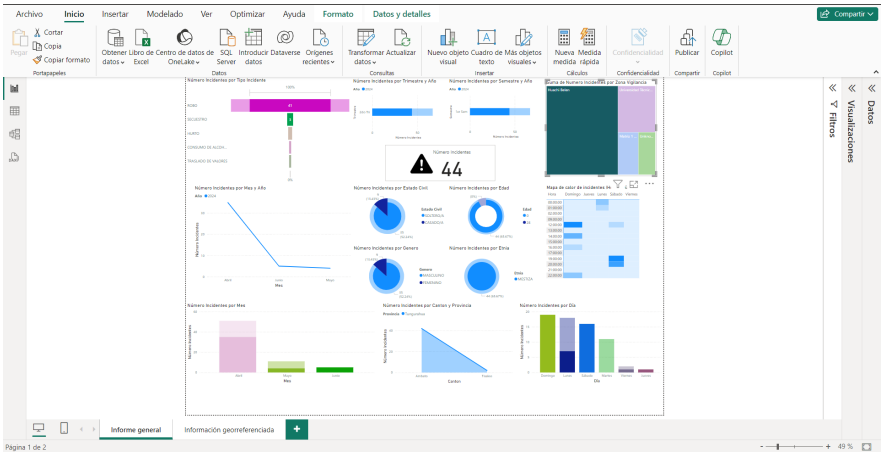


Figura 84. Informe en Power BI con los datos filtrados.

El segundo informe en Power BI se centra en la información de forma georreferenciada, mostrando la cantidad de incidentes reportados en un mapa interactivo en el cual se visualiza las zonas de vigilancia y la cantidad de incidentes reportados en cada una, como se puede observar en la Figura 85.

Este informe también muestra uh mapa de calor de las zonas con mayor densidad de incidentes por hora del día, así como información demográfica de los usuarios que reportaron los incidentes, como género y edad, así como la distribución de los incidentes por año y mes.

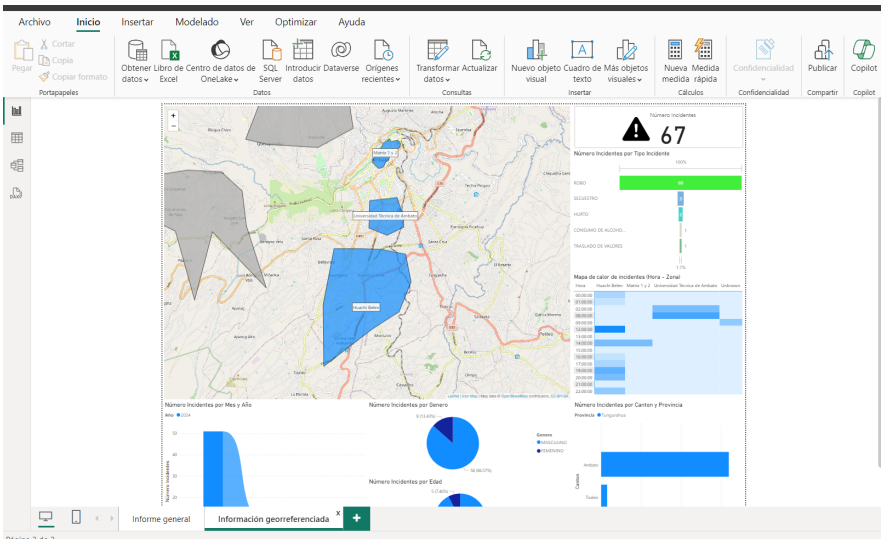


Figura 85. Informe en Power BI con los datos del cubo OLAP.

En este informe, al igual que en el anterior, se pueden realizar un filtrado de los datos de forma interactiva al hacer clic sobre un gráfico, tabla o mapa, los demás elementos del

informe se actualizan automáticamente para mostrar la información relacionada con el elemento seleccionado, como se puede observar en la Figura 86.

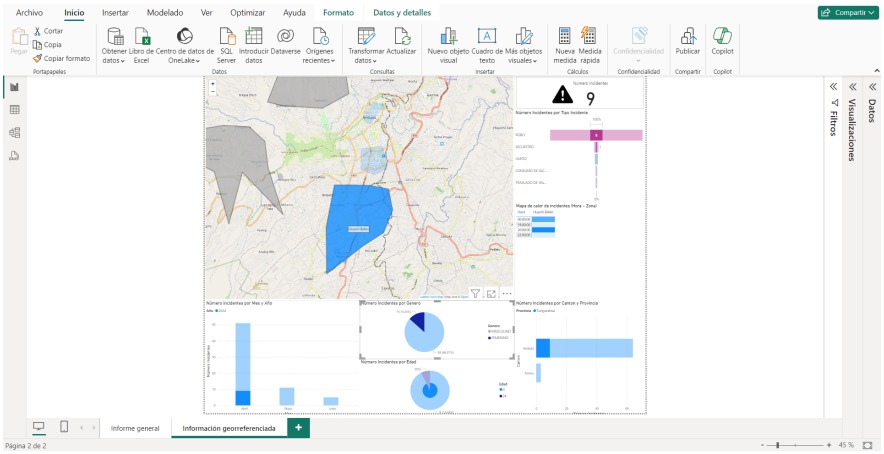


Figura 86. Informe en Power BI con los datos filtrados.

La información mostrada en estos reportes permite responder a las preguntas del negocio planteadas en la sección anterior, como determinar el número de incidentes por tipo de incidente y zona de vigilancia, analizar la distribución de incidentes por género, discapacidad, etnia y edad de los usuarios, e identificar las zonas con mayor número de incidentes para optimizar la vigilancia policial.

3.4.4 Fase IV: Pruebas

En esta fase se realizaron las pruebas de aceptación en base a las historias de usuario definidas en la fase de planificación. En la tabla 70 se presenta el modelo de pruebas de aceptación que será utilizado para validar los criterios de aceptación de las historias de usuario.

Tabla 70. Modelo de prueba de aceptación

Prueba de aceptación	
Número <Número de la prueba de aceptación>	**Historia de usuario:** <Número de la historia de usuario>
Nombre de la historia: <Título de la historia de usuario>	
Continua en la siguiente página	

Descripción: <Se describe con lenguaje natural sobre lo que debe hacer cada historia>
Condiciones de ejecución: <Se especifican las condiciones necesarias que deben estar satisfechas para llevar a cabo la historia de usuario>
Entrada/pasos de ejecución: <Descripción de las acciones que el usuario lleva a cabo para completar la historia de usuario>
Resultado esperado: <Descripción del resultado esperado según los criterios de aceptación establecidos para la historia de usuario>
Evaluación de prueba: <Resultado final de la prueba>

a. Pruebas de aceptación

A continuación, se presentan las pruebas de aceptación realizadas en la fase de pruebas. En la tabla 71 se presenta la prueba de aceptación 1, la cual corresponde a la historia de usuario 1. En esta prueba se evalúa la funcionalidad de iniciar sesión en el sistema web.

Tabla 71. Prueba de aceptación 1: Iniciar sesión sistema web

Prueba de aceptación	
Número 1	**Historia de usuario:** 1
Nombre de la historia: Iniciar sesión sistema web	
Descripción: Como usuario administrador del sistema web, quiero iniciar sesión en el sistema para acceder a sus funcionalidades	
Condiciones de ejecución: Ser un usuario administrador	
Entrada/pasos de ejecución: El usuario debe ingresar sus credenciales de acceso	
Resultado esperado: Dado que el usuario ingrese su correo y contraseña y seleccione ingresar, entonces el sistema web le permitirá ingresar a sus funcionalidades	
Evaluación de prueba: Satisfactoria	

En la tabla 72 se presenta la prueba de aceptación 2, la cual corresponde a la historia

de usuario 2. En esta prueba se evalúa la funcionalidad de cerrar sesión en el sistema web.

Tabla 72. Prueba de aceptación 2: Cerrar sesión sistema web

Prueba de aceptación	
Número 2	**Historia de usuario:** 2
Nombre de la historia: Cerrar sesión sistema web	
Descripción: Como usuario del sistema web, quiero cerrar sesión en el sistema para finalizar mi sesión de trabajo	
Condiciones de ejecución: Ser un usuario administrador y acceder al sistema	
Entrada/pasos de ejecución: El usuario debe seleccionar la opción ”cerrar sesión” en el menú de opciones	
Resultado esperado: Dado que el usuario seleccione la opción ”cerrar sesión”, entonces el sistema debe redirigirlo a la pantalla de ingreso	
Evaluación de prueba: Satisfactoria	

En la tabla 73 se presenta la prueba de aceptación 3, la cual corresponde a la historia de usuario 3. En esta prueba se evalúa la funcionalidad de gestionar usuarios en el sistema web.

Tabla 73. Prueba de aceptación 3: Gestionar usuarios

Prueba de aceptación	
Número 3	**Historia de usuario:** 3
Nombre de la historia: Gestionar usuarios	
Descripción: Como usuario administrador, quiero administrar los usuarios que se encuentran registrados en el sistema	
Condiciones de ejecución: Ser un usuario administrador	
Entrada/pasos de ejecución: El usuario administrador debe haber accedido al sistema e ingresar en la pantalla de usuarios	
Continua en la siguiente página	

Resultado esperado: Dado que el usuario administrador ingrese a la pantalla de usuarios, cuando quiera ingresar, modificar o deshabilitar un usuario, entonces el sistema permitirá la actualización de la información en la base de datos
Evaluación de prueba: Satisfactoria

En la tabla 74 se presenta la prueba de aceptación 4, la cual corresponde a la historia de usuario 4. En esta prueba se evalúa la funcionalidad de gestionar tipos de incidentes en el sistema web.

Tabla 74. Prueba de aceptación 4: Gestionar tipos de incidentes

Prueba de aceptación	
Número 4	**Historia de usuario:** 4
Nombre de la historia: Gestionar tipos de incidentes	
Descripción: Como usuario administrador, quiero gestionar los tipos de incidentes para definir las categorías de los incidentes	
Condiciones de ejecución: Ser un usuario administrador	
Entrada/pasos de ejecución: El usuario administrador debe haber accedido al sistema e ingresar en la pantalla de tipos de incidentes	
Resultado esperado: Dado que el usuario administrador ingrese a la pantalla de tipos de incidentes, cuando quiera visualizar, crear, actualizar, deshabilitar o habilitar un tipo de incidente, entonces el sistema permitirá la actualización de la información en la base de datos	
Evaluación de prueba: Satisfactoria	

En la tabla 75 se presenta la prueba de aceptación 5, la cual corresponde a la historia de usuario 5. En esta prueba se evalúa la funcionalidad de gestionar zonas de vigilancia en el sistema web.

Tabla 75. Prueba de aceptación 5: Gestionar zonas de vigilancia

Prueba de aceptación	
Número 5	**Historia de usuario:** 5
Nombre de la historia: Gestionar zonas de vigilancia	
Descripción: Como usuario administrador, quiero gestionar las zonas de vigilancia para definir las áreas de monitoreo	
Condiciones de ejecución: Ser un usuario administrador	
Entrada/pasos de ejecución: El usuario administrador debe haber accedido al sistema e ingresar en la pantalla de zonas de vigilancia	
Resultado esperado: Dado que el usuario administrador ingrese a la pantalla de zonas de vigilancia, cuando quiera visualizar, crear, actualizar, deshabilitar o habilitar una zona de vigilancia, entonces el sistema permitirá la actualización de la información en la base de datos	
Evaluación de prueba: Satisfactoria	

En la tabla 76 se presenta la prueba de aceptación 6, la cual corresponde a la historia de usuario 6. En esta prueba se evalúa la funcionalidad de asignar policías a las zonas de vigilancia en el sistema web.

Tabla 76. Prueba de aceptación 6: Asignar policías a las zonas de vigilancia

Prueba de aceptación	
Número 6	**Historia de usuario:** 6
Nombre de la historia: Asignar policías a las zonas de vigilancia	
Descripción: Como usuario administrador, quiero asignar y desasignar miembros de la policía a las zonas de vigilancia	
Condiciones de ejecución: Ser un usuario administrador	
Entrada/pasos de ejecución: El usuario administrador debe haber accedido al sistema e ingresar en la pantalla de asignación de policías a zonas de vigilancia	
Continua en la siguiente página	

Resultado esperado: Dado que el usuario administrador ingrese a la pantalla de asignación de policías a zonas de vigilancia, cuando quiera visualizar, asignar o desasignar un miembro de la policía a una zona de vigilancia, entonces el sistema permitirá la actualización de la información en la base de datos
Evaluación de prueba: Satisfactoria

En la tabla 77 se presenta la prueba de aceptación 7, la cual corresponde a la historia de usuario 7. En esta prueba se evalúa la funcionalidad de gestionar alertas de incidentes en el sistema web.

Tabla 77. Prueba de aceptación 7: Gestionar alertas de incidentes

Prueba de aceptación	
Número 7	**Historia de usuario:** 7
Nombre de la historia: Gestionar alertas de incidentes	
Descripción: Como usuario administrador, quiero visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real	
Condiciones de ejecución: Ser un usuario administrador	
Entrada/pasos de ejecución: El usuario administrador debe haber accedido al sistema e ingresar en la pantalla de visualización de alertas de incidentes	
Resultado esperado: Dado que el usuario administrador ingrese a la pantalla de visualización de alertas de incidentes, cuando quiera visualizar las alertas de emergencia en un mapa junto con su posición en tiempo real, entonces el sistema mostrará dicha información correctamente	
Evaluación de prueba: Satisfactoria	

En la tabla 78 se presenta la prueba de aceptación 8, la cual corresponde a la historia de usuario 8. En esta prueba se evalúa la funcionalidad de visualizar el mapa de calor de incidentes delictivos en el sistema web.

Tabla 78. Prueba de aceptación 8: Visualizar mapa de calor

Prueba de aceptación	
Número 8	**Historia de usuario:** 8
Nombre de la historia: Visualizar mapa de calor	
Descripción: Como usuario administrador, quiero visualizar mediante un mapa de calor los incidentes delictivos suscitados en las diferentes zonas	
Condiciones de ejecución: Ser un usuario administrador	
Entrada/pasos de ejecución: El usuario administrador debe haber accedido al sistema e ingresar en la pantalla de visualización del mapa de calor de incidentes delictivos	
Resultado esperado: Dado que el usuario administrador ingrese a la pantalla de visualización del mapa de calor, cuando quiera visualizar los incidentes delictivos en un mapa mediante un mapa de calor, entonces el sistema mostrará dicha información de manera clara y representativa	
Evaluación de prueba: Satisfactoria	

En la tabla 79 se presenta la prueba de aceptación 9, la cual corresponde a la historia de usuario 9. En esta prueba se evalúa la funcionalidad de iniciar sesión en la aplicación móvil.

Tabla 79. Prueba de aceptación 9: Iniciar sesión aplicación móvil

Prueba de aceptación	
Número 9	**Historia de usuario:** 9
Nombre de la historia: Iniciar sesión aplicación móvil	
Descripción: Como ciudadano, quiero iniciar sesión en el sistema móvil para acceder a sus funcionalidades	
Condiciones de ejecución: Ser un ciudadano con cuenta registrada en el sistema móvil	
Entrada/pasos de ejecución: El ciudadano ingresa su correo electrónico y contraseña en la pantalla de inicio de sesión del sistema móvil	
Continua en la siguiente página	

Resultado esperado: Dado que el ciudadano ingrese correctamente su correo y contraseña y presione el botón de inicio de sesión, la aplicación lo redirigirá a la pantalla principal del sistema móvil. En caso de ingresar incorrectamente los datos, la aplicación mostrará un mensaje de error adecuado.
Evaluación de prueba: Satisfactoria

En la tabla 80 se presenta la prueba de aceptación 10, la cual corresponde a la historia de usuario 10. En esta prueba se evalúa la funcionalidad de cerrar sesión en la aplicación móvil.

Tabla 80. Prueba de aceptación 10: Cerrar sesión aplicación móvil

Prueba de aceptación	
Número 10	**Historia de usuario:** 10
Nombre de la historia: Cerrar sesión aplicación móvil	
Descripción: Como ciudadano, quiero cerrar sesión en la aplicación móvil para finalizar mi sesión de trabajo	
Condiciones de ejecución: Ser un ciudadano con sesión iniciada en la aplicación móvil	
Entrada/pasos de ejecución: El ciudadano presiona el botón de cerrar sesión desde cualquier pantalla de la aplicación móvil	
Resultado esperado: Dado que el ciudadano presione el botón de cerrar sesión, la aplicación lo redirigirá a la pantalla de inicio de sesión, finalizando correctamente su sesión activa.	
Evaluación de prueba: Satisfactoria	

En la tabla 81 se presenta la prueba de aceptación 11, la cual corresponde a la historia de usuario 11. En esta prueba se evalúa la funcionalidad de registro de usuario en la aplicación móvil.

Tabla 81. Prueba de aceptación 11: Registro de usuario

Prueba de aceptación	
Número 11	**Historia de usuario:** 11
Nombre de la historia: Registro de usuario	
Descripción: Como ciudadano, quiero ingresar mis datos y fotografía mediante un formulario	
Condiciones de ejecución: Ser un ciudadano que desea registrarse en el sistema	
Entrada/pasos de ejecución: 1. El ciudadano accede a la pantalla de registro en la aplicación móvil. 2. Ingresa sus datos personales (nombre, correo electrónico, etc.) y selecciona una fotografía. 3. Completa y envía el formulario de registro.	
Resultado esperado: Dado que el ciudadano complete y envíe el formulario de registro con sus datos personales y fotografía, el sistema deberá almacenar correctamente la información en la base de datos, permitiendo así el registro exitoso del usuario en el sistema.	
Evaluación de prueba: Satisfactoria	

En la tabla 82 se presenta la prueba de aceptación 12, la cual corresponde a la historia de usuario 12. En esta prueba se evalúa la funcionalidad de asignar miembros al grupo familiar en la aplicación móvil.

Tabla 82. Prueba de aceptación 12: Asignar miembros al grupo familiar

Prueba de aceptación	
Número 12	**Historia de usuario:** 12
Nombre de la historia: Asignar miembros al grupo familiar	
Descripción: Como ciudadano, quiero asignar miembros al grupo familiar para recibir alertas de emergencia	
Continua en la siguiente página	

Condiciones de ejecución: Ser un ciudadano registrado en el sistema y tener acceso a la funcionalidad de gestión de grupo familiar
Entrada/pasos de ejecución: 1. El ciudadano accede a la pantalla de gestión de grupo familiar en la aplicación móvil. 2. Selecciona la opción para añadir un nuevo miembro al grupo. 3. Ingresa los datos del nuevo miembro (nombre, correo electrónico, etc.). 4. Confirma la asignación del nuevo miembro.
Resultado esperado: Dado que el ciudadano complete los pasos para añadir un nuevo miembro al grupo familiar, el sistema deberá registrar correctamente al nuevo miembro como parte del grupo, permitiendo así que reciba alertas de emergencia junto con el ciudadano.
Evaluación de prueba: Satisfactoria

En la tabla 83 se presenta la prueba de aceptación 13, la cual corresponde a la historia de usuario 13. En esta prueba se evalúa la funcionalidad de enviar alerta de incidente en la aplicación móvil.

Tabla 83. Prueba de aceptación 13: Enviar alerta de incidente

Prueba de aceptación	
Número 13	**Historia de usuario:** 13
Nombre de la historia: Enviar alerta de incidente	
Descripción: Como ciudadano, quiero enviar una alerta de incidente para notificar a las autoridades sobre un incidente delictivo	
Condiciones de ejecución: Ser un ciudadano registrado en el sistema y tener acceso a la funcionalidad de enviar alertas de incidentes.	
Continua en la siguiente página	

Entrada/pasos de ejecución: 1. El ciudadano abre la aplicación móvil y accede a la pantalla principal. 2. Selecciona el tipo de incidente (robo, agresión, etc.) y proporciona una descripción breve. 3. Presiona el botón de alerta durante 3 para confirmar el envió de la alerta.
Resultado esperado: Dado que el ciudadano complete los pasos para enviar una alerta de incidente, el sistema deberá registrar la alerta y notificar a los entes de control y a los miembros del grupo familiar.
Evaluación de prueba: Satisfactoria

Tabla 84. Prueba de aceptación 14: Visualizar alertas de emergencia de familiares

Prueba de aceptación	
Número 14	**Historia de usuario:** 14
Nombre de la historia: Visualizar alertas de emergencia de familiares	
Descripción: Como ciudadano, quiero visualizar mediante un mapa las alertas de emergencia enviadas por mis familiares así como su posición en tiempo real	
Condiciones de ejecución: Ser un ciudadano con miembros de grupo familiar registrados	
Entrada/pasos de ejecución: El ciudadano debe iniciar sesión en el sistema y acceder a la sección de alertas recibidas.	
Resultado esperado: Dado que el ciudadano ingrese al apartado de alertas recibidas, el sistema deberá mostrar las alertas de emergencia enviadas por los miembros de su grupo familiar, así como la ubicación en tiempo real en el mapa de alertas.	
Evaluación de prueba: Satisfactoria	

En la tabla 85 se presenta la prueba de aceptación 15, la cual corresponde a la historia de usuario 15. En esta prueba se evalúa la funcionalidad de visualizar alertas de emergencia de ciudadanos en la aplicación móvil.

Tabla 85. Prueba de aceptación 15: Visualizar alertas de emergencia de ciudadanos

Prueba de aceptación	
Número 15	**Historia de usuario:** 15
Nombre de la historia: Visualizar alertas de emergencia de ciudadanos	
Descripción: Como policía, quiero visualizar mediante un mapa las alertas de emergencia enviadas por los ciudadanos así como su posición en tiempo real y el tipo de incidente	
Condiciones de ejecución: Ser un policía autenticado en el sistema	
Entrada/pasos de ejecución: El policía accede a la pantalla principal de la aplicación.	
Resultado esperado: Dado que el policía ingrese a la aplicación, el sistema deberá mostrar las alertas de emergencia enviadas por los ciudadanos, así como la ubicación en tiempo real en el mapa de alertas	
Evaluación de prueba: Satisfactoria	

En la tabla 86 se presenta la prueba de aceptación 16, la cual corresponde a la historia de usuario 16. En esta prueba se evalúa la funcionalidad de cambiar la contraseña en la aplicación móvil.

Tabla 86. Prueba de aceptación 16: Cambiar contraseña

Prueba de aceptación	
Número 16	**Historia de usuario:** 16
Nombre de la historia: Cambiar contraseña	
Descripción: Como ciudadano, quiero cambiar mi contraseña para mejorar la seguridad de mi cuenta	
Condiciones de ejecución: Ser un ciudadano autenticado en el sistema	
Entrada/pasos de ejecución: El ciudadano accede a la sección de cambio de contraseña en el menú, ingresa su contraseña actual, la nueva contraseña y confirma el cambio.	
Continúa en la siguiente página	

Resultado esperado: Dado que el ciudadano ingrese su contraseña actual correctamente y las nuevas contraseñas coincidan, el sistema deberá actualizar la contraseña del ciudadano y confirmar el cambio exitoso.
Evaluación de prueba: Satisfactoria

En la tabla 87 se presenta la prueba de aceptación 17, la cual corresponde a la historia de usuario 17. En esta prueba se evalúa la funcionalidad de recuperar contraseña en la aplicación móvil.

Tabla 87. Prueba de aceptación 17: Recuperar contraseña

Prueba de aceptación	
Número 17	**Historia de usuario:** 17
Nombre de la historia: Recuperar contraseña	
Descripción: Como ciudadano, quiero recuperar mi contraseña en caso de olvidarla	
Condiciones de ejecución: El ciudadano debe estar registrado en el sistema y tener un correo electrónico válido asociado a su cuenta	
Entrada/pasos de ejecución: 1. El ciudadano selecciona la opción ”Olvidé mi contraseña” en la pantalla de inicio de sesión. 2. El ciudadano ingresa su dirección de correo electrónico asociada a su cuenta. 3. El sistema envía un correo electrónico con una clave provisional. 4. El ciudadano ingresa al sistema con su clave provisional. 5. El ciudadano cambia su contraseña.	
Continúa en la siguiente página	

Resultado esperado: Dado que el ciudadano complete los pasos para recuperar la contraseña, el sistema deberá permitirle establecer una nueva contraseña y notificarle del cambio exitoso.
Evaluación de prueba: Satisfactoria

En la tabla 88 se presenta la prueba de aceptación 18, la cual corresponde a la historia de usuario 18. En esta prueba se evalúa la funcionalidad de desarrollar un modelo analítico de BI en Power BI.

Tabla 88. Prueba de aceptación 18: Desarrollar modelo analítico de BI en Power BI

Prueba de aceptación	
Número 18	**Historia de usuario:** 18
Nombre de la historia: Desarrollar modelo analítico de BI en Power BI	
Descripción: Como administrador, quiero visualizar informes detallados sobre los incidentes ocurridos usando Power BI	
Condiciones de ejecución: Ser un usuario administrador autenticado en el sistema con acceso a Power BI	
Entrada/pasos de ejecución: 1. El administrador inicia sesión en Power BI. 2. El administrador accede al panel de informes de incidentes. 3. El administrador selecciona el tipo de informe que desea visualizar (Informe general, información georreferenciada). 4. Power BI genera y muestra el informe detallado con gráficos interactivos. 5. El administrador realiza clic en las gráficas para obtener información específica.	
Continúa en la siguiente página	

Resultado esperado: ▪ Power BI debe generar y mostrar informes detallados sobre los incidentes ocurridos con gráficos interactivos. ▪ Al realizar clic en las gráficas, Power BI debe mostrar información relacionada con los datos seleccionados.
Evaluación de prueba: Satisfactoria

3.4.5 Fase V: Lanzamiento

En esta fase se realizó la puesta en producción de la aplicación web, móvil y el API junto al modelo analítico de BI. Para la puesta en producción de la aplicación web y el API se utilizó Railway, un servicio de alojamiento de aplicaciones web que permite desplegar aplicaciones de forma sencilla y rápida manejando el proceso de CI/CD al desplegar los sistemas desde su repositorio en GitHub, en las Figuras 87 y 88 se muestran los servicios desplegados en Railway.

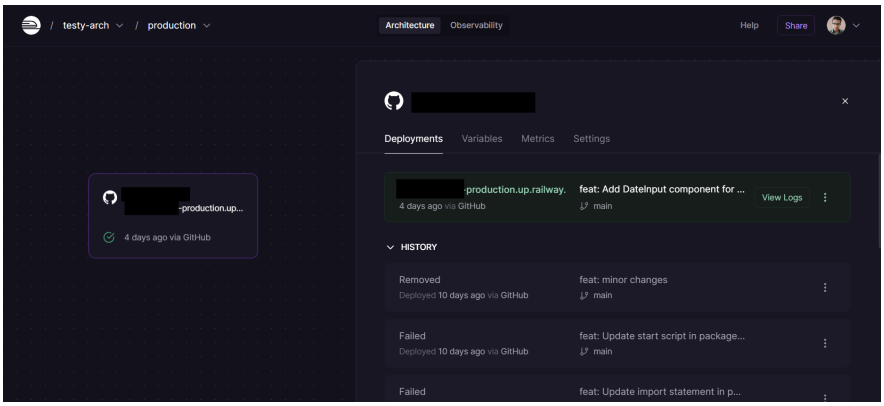


Figura 87. Despliegue de la aplicación web en Railway.

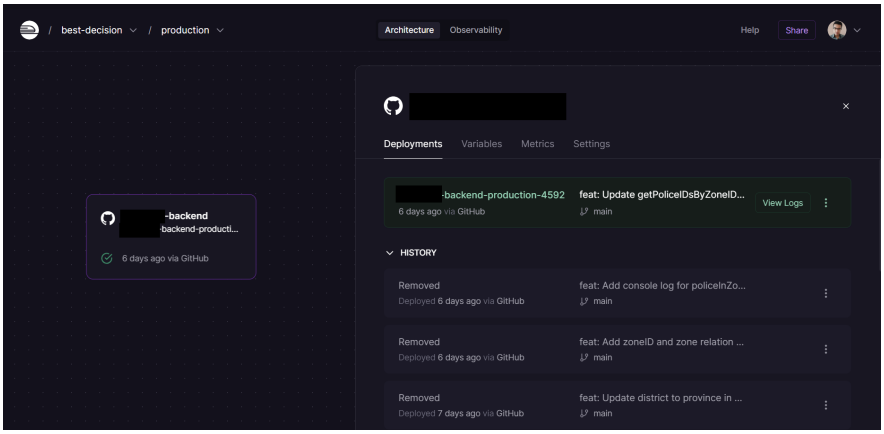


Figura 88. Despliegue del API en Railway.

Para la aplicación móvil se creo un archivo APK que permite instalar la aplicación en dispositivos Android, en la Figura 89 se muestra el archivo APK generado para la aplicación móvil.

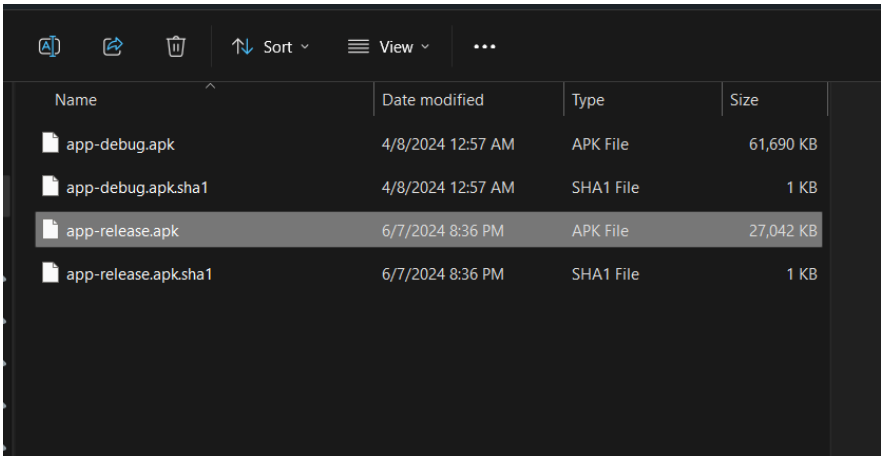


Figura 89. Archivo APK de la aplicación móvil.

Para la puesta en producción del modelo analítico de BI se lo mantuvo en el servidor de SQL Server Analysis Services de forma local, esto debido a que no se contaba con un servidor de Analysis Services en la nube ni las licencias necesarias para obtener uno.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

El presente proyecto de titulación se desarrolló con el objetivo de implementar una solución tecnológica que permita mejorar la recolección de datos georreferenciados de delitos en la ciudad de Ambato. Con estos datos se diseñará un modelo analítico de BI orientado al seguimiento de las tendencias de siniestros, mediante un sistema web y móvil que facilite la obtención de información precisa para su posterior análisis.

4.1 Conclusiones

- En base a la entrevista realizada al coronel de la policía, se logró identificar el proceso actual que se lleva a cabo al momento de reportar un siniestro, lo cual permitió determinar las necesidades y requerimientos necesarios para la implementación del sistema de obtención de datos georreferenciados. La información obtenida a través de la entrevista realizada al ingeniero experto en BI permitió identificar las características y funcionalidades necesarias que deben tener el modelo analítico de BI y el sistema de reportería, así como los medios y formatos correctos para visualizar la información obtenida.

- La aplicación de frameworks de desarrollo demuestran su uso como una herramienta eficaz la cual permite acelerar el proceso creación de software. Estos ofrecen un marco de trabajo que facilita la implementación de aplicaciones. En el caso de este proyecto se utilizaron tecnologías basadas en JavaScript y Dart, las cuales cuales cuentan con una amplia comunidad de desarrolladores y una documentación extensa los cuales pueden ofrecer soporte en caso de errores o dudas. Sin embargo, la elección de un framework debe ser cuidadosa, ya que que este debe adaptarse a las necesidades del proyecto y las capacidades del equipo de desarrollo.

- La precisión de los datos georreferenciados es un factor importante en la obtención de información precisa para el análisis de datos. Con la finalidad de evaluar la precisión de los datos obtenidos, se emplearon técnicas estadísticas

para determinar la fiabilidad de la información. Mediante el empleo de un intervalo de confianza se pudo determinar que en un 95% de los casos, la precisión de los datos georreferenciados se encuentra en un rango entre 7.08 y 8.02 metros con respecto a la ubicación real del incidente.

- La implementación de un modelo analítico de BI, empleando la metodología Hefesto para la construcción de un Data Warehouse, ha demostrado ser una estrategia efectiva para optimizar el rendimiento en la gestión de datos. El diseño del proceso ETL, llevado a cabo con la herramienta de Visual Studio, ha permitido la consolidación eficiente de datos provenientes de diversas fuentes, asegurando su compatibilidad y adecuada inserción en el modelo dimensional. La creación del cubo OLAP y su visualización mediante Power BI han facilitado un análisis detallado de la tendencia de siniestros.

- La metodología de desarrollo ágil XP permitió un desarrollo rápido y eficiente, gracias a su enfoque iterativo y a la participación activa de los usuarios, lo cual facilitó la adaptación de los requerimientos a medida que se avanzaba en el desarrollo del sistema.

4.2 Recomendaciones

- Si el sistema se implementa para su uso en conjunto con la policía, se recomienda realizar una capacitación previa a los oficiales de policía, con el fin de que puedan utilizar la herramienta de manera eficiente y efectiva, así como una concienciación a los ciudadanos para evitar el mal uso de la herramienta.

- Para mejorar la calidad de la información así como reducir las falsas alertas se recomienda la integración de tecnologías para streaming de video, audio e imágenes, y con ello poder de realizar un análisis en tiempo real de las emergencias.

- Dados el tipo de datos que se manejan y la cantidad de información que se debe procesar, se recomienda utilizar servicios de computación en la nube como azure

analysis services, con el fin de mejorar el rendimiento y la escalabilidad de la solución del cubo OLAP.

- Cuando la cantidad de datos incremente de forma significativa, se recomienda realizar un análisis para la incorporación de técnicas de inteligencia artificial, con el fin de realizar predicciones y análisis más avanzados.

REFERENCIAS BIBLIOGRÁFICAS

[1] A. Rummens, T. Snaphaan, N. Van De Weghe, D. Van Den Poel, L. J. R. Pauwels, y W. Hardyns, "Do Mobile Phone Data Provide a Better Denominator in Crime Rates and Improve Spatiotemporal Predictions of Crime?" *ISPRS International Journal of Geo-Information*, vol. 10, no. 6, p. 369, May 2021.

[2] E. Ahishakiye, R. Wario, E. O. Omulo, y I. Niyonzima, "A Performance Analysis of Business Intelligence Techniques on Crime Prediction," vol. 06, no. 02.

[3] I. D. Reid, S. Appleby-Arnold, N. Brockdorff, I. Jakovljev, y S. Zdravković, "Developing a model of perceptions of security and insecurity in the context of crime," *Psychiatry, Psychology and Law*, vol. 27, no. 4, pp. 620–636, Jul. 2020.

[4] N. Tasnim, I. T. Imam, y M. M. A. Hashem, "A Novel Multi-Module Approach to Predict Crime Based on Multivariate Spatio-Temporal Data Using Attention and Sequential Fusion Model," *IEEE Access*, vol. 10, pp. 48 009–48 030, 2022.

[5] T. Vieira, T. Paulino, J. M. S. Souza, y E. Lima, "Crime prediction and prevention using police patrolling data: Challenges and prospects," en *Anais Estendidos Da Conference on Graphics, Patterns and Images (SIBGRAPI)*. SBC, Oct. 2022, pp. 183–186.

[6] ECLAC. United Nations, "Reference framework for security and criminal justice statistics in Latin America and the Caribbean," p. 71, 2022.

[7] E. D. Arias, "Crime and Punishment in Latin America," *Latin American Research Review*, pp. 1–11, Oct. 2023.

[8] X. Tayupanta, "BOLETÍN SEMESTRAL DE HOMICIDIOS INTENCIONALES EN ECUADOR," Fundación Panamericana para el Desarrollo (PADF), Rep. Tec., 2023.

[9] D. A. Narváez, "LA GESTIÓN DE INFORMACIÓN CRIMINOLÓGICA POLICIAL," *INNOVACIÓN & SABER*, no. 7, pp. 40–48, 2018.

[10] D. T. V. Álvarez, “Compendio económico y social OBEST - INFORMA,” vol. 1, p. 84, 2022.

[11] J. X. I. Quintana, C. B. C. Guamán, J. L. D. Paredes, y L. P. M. Quitio, “Análisis sistemático del aumento de la inseguridad en el Ecuador.” *Dilemas contemporáneos: Educación, Política y Valores*, Ene. 2023.

[12] F. Herrera, R. Sosa, y T. Delgado, “GeoBI and Big VGI for Crime Analysis and Report,” en *2015 3rd International Conference on Future Internet of Things and Cloud*. Rome, Italy: IEEE, Ago. 2015, pp. 481–488.

[13] Y.-L. Lin, T.-Y. Chen, y L.-C. Yu, “Using Machine Learning to Assist Crime Prevention,” en *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. Hamamatsu: IEEE, Jul. 2017, pp. 1029–1030.

[14] “Crimes Prediction Using Spatio-Temporal Data and Kernel Density Estimation,” en *2019 Asia Pacific Conference on Research in Industrial and Systems Engineering (APCoRISE)*. Depok, Indonesia: IEEE, Abr. 2019, pp. 1–6.

[15] K. N. Win, J. Chen, Y. Chen, y P. Fournier-Viger, “PCPD: A Parallel Crime Pattern Discovery System for Large-Scale Spatiotemporal Data Based on Fuzzy Clustering,” *International Journal of Fuzzy Systems*, vol. 21, no. 6, pp. 1961–1974, Sep. 2019.

[16] A. Amirkhanyan, “Methods and Frameworks for GeoSpatioTemporal Data Analytics,” Ph.D. dissertation, Universität Potsdam, 2020.

[17] A. Kumar, A. Verma, G. Shinde, Y. Sukhdeve, y N. Lal, “Crime Prediction Using K-Nearest Neighboring Algorithm,” en *2020 International Conference on Emerging Trends in Information Technology and Engineering (Ic-ETITE)*. Vellore, India: IEEE, Feb. 2020, pp. 1–4.

[18] S. Hossain, A. Abtahee, I. Kashem, M. M. Hoque, y I. H. Sarker, “Crime Prediction Using Spatio-Temporal Data,” en *Computing Science, Communication and Security*, N. Chaubey, S. Parikh, y K. Amin, Eds. Singapore: Springer Singapore, 2020, vol. 1235, pp. 277–289.

[19] M. Botto-Tobar, S. Montes León, O. Camacho, D. Chávez, P. Torres-Carrión, y M. Zambrano Vizuete, Eds., *Applied Technologies: Second International Conference, ICAT 2020, Quito, Ecuador, December 2–4, 2020, Proceedings*, ser. Communications in Computer and Information Science. Cham: Springer International Publishing, 2021, vol. 1388.

[20] C. P. Chasi Chango, "Aplicación móvil de apoyo a la seguridad barrial para envío y localización de alertas de auxilio mediante notificaciones push en la parroquia santa rosa de la ciudad de Ambato," Ph.D. dissertation, 2022.

[21] F. V. Lesano Pérez, "Aplicativo móvil de geo-ubicación en tiempo real, para el transporte privado de Plasticaucho Industrial S.A de la ciudad de Ambato," Ph.D. dissertation, Universidad Técnica de Ambato, 2022.

[22] J. A. Chicaiza Villegas, "Aplicación web para la gestión de servicios, notificaciones de pago y geolocalización de viviendas de los clientes en la empresa Optynet del cantón Quero," Ph.D. dissertation, Universidad Técnica de Ambato, 2023.

[23] K. G. Cantillo y I. E. Villalobos Mercado, "Aplicativo Móvil para la Predicción de Delitos en la Ciudad de Valledupar – VIGIS," p. 7.

[24] F. S. Remache Gañay, "APLICATIVO WEB UTILIZANDO FRAMEWORK OPEN SOURCE PARA LA GESTIÓN DE ACTIVIDADES DE LA FUNDACIÓN DE RESCATE ANIMAL PAQUITO DE LA CIUDAD DE AMBATO," Ph.D. dissertation, UNIVERSIDAD TÉCNICA DE AMBATO.

[25] E. Haro, T. Guarda, A. O. Z. Peñaherrera, y N. Quiña, "Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot."

[26] A. D. Pham, "DEVELOPING BACK-END OF A WEB APPLICATION WITH NESTJS FRAMEWORK," p. 46, 2020.

[27] "Express - Infraestructura de aplicaciones web Node.js," https://expressjs.com/es/.

[28] "Fast and low overhead web framework, for Node.js | Fastify," https://fastify.dev/.

[29] “Stack Overflow Developer Survey 2023,” https://survey.stackoverflow.co/2023/?utm_source=social-share&utm_medium=social&utm_ca

[30] J. Cincović y M. Punt, “Comparison: Angular vs. React vs. Vue. Which framework is the best choice?” 2020.

[31] Y. Xing, J. Huang, y Y. Lai, “Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development,” en *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering*. Perth WN Australia: ACM, Feb. 2019, pp. 68–72.

[32] A. J. Alférez Zamora, “Estudio comparativo de frameworks multiplataforma para desarrollo de aplicaciones móviles,” Ph.D. dissertation, Universitãt Stuttgart, 2018.

[33] F. O. C. Lazo, “ANÁLISIS Y DISEÑO DE UN APLICATIVO MOVIL, CON LA UTILIZACION DE UN FRAMEWORK MULTIPLAFORMA, PARA REGISTRO DE INCIDENTES PARA LA EMPRESA FIBERMEDIA TELECOMUNICACIONES.”

[34] M. Basyir, M. Nasir, S. Suryati, y W. Mellyssa, “Determination of Nearest Emergency Service Office using Haversine Formula Based on Android Platform,” *EMITTER International Journal of Engineering Technology*, vol. 5, no. 2, pp. 270–278, 2017.

[35] B. R. Dario y G. M. Mariano, “DATA WAREHOUSING: Marco Conceptual HEFESTO: Metodología Data Warehouse,” p. 182, 2018.

ANEXOS

Anexo A. Guía de entrevista aplicada a experto en Business Intelligence

N.	Pregunta	Respuesta	Observación
1	¿Qué características considera usted que debe tener un indicador clave de rendimiento (KPI) para ser considerado relevante?		
2	¿Qué frecuencia considera adecuada para la actualización de los informes de BI? ¿Diaria, semanal, mensual u otra?		
3	¿Qué formatos de presentación de datos prefieren para facilitar la comprensión y análisis de la información? (por ejemplo, gráficos, tablas, dashboards interactivos, entre otros.)		
4	¿Cuáles son los principales desafíos que enfrenta al interpretar los datos y convertirlos en acciones concretas?		
5	¿Qué características considera esenciales en un sistema de BI para garantizar su efectividad en la toma de decisiones?		
6	¿Qué herramientas o tecnologías de BI considera más útiles y eficaces en la actualidad?		

Continua en la siguiente página

Tabla 1 – continuación de la página anterior

N.	Pregunta	Respuesta	Observación
7	¿Cómo se pueden abordar eficazmente los problemas de escalabilidad en sistemas de Business Intelligence a medida que la cantidad de datos y usuarios aumenta?		
8	¿Qué estrategias o enfoques considera más efectivos para garantizar la calidad y confiabilidad de los datos utilizados en los informes de Business Intelligence?		
9	¿Cuál es su enfoque para identificar y priorizar los requisitos de los usuarios al diseñar informes de BI?		

Anexo B. Guía de entrevista aplicada al coronel de la policía nacional

N.	Pregunta	Respuesta	Observación
1	¿Qué tipo de información o datos específicos sobre incidentes delictivos considera más críticos para la gestión efectiva de la seguridad pública?		
Continua en la siguiente página			

Tabla 1 – continuación de la página anterior

N.	Pregunta	Respuesta	Observación
2	¿Cuál es la disponibilidad y calidad de los datos actuales sobre incidentes delictivos? ¿Existen desafíos específicos en la recolección o integridad de los datos?		
3	¿Cuáles son los principales desafíos que enfrenta actualmente el departamento de policía en la recopilación, análisis y utilización de datos relacionados con delitos?		
4	¿Con qué frecuencia y en qué formato se recopilan actualmente los datos sobre incidentes delictivos? ¿Hay algún requisito específico en términos de temporalidad y presentación de la información?		
5	¿Qué características o funcionalidades le gustaría ver en un sistema de análisis de incidentes delictivos diseñado para el departamento de policía en términos de visualización de datos, generación de informes, capacidad de análisis, entre otros?		

Continua en la siguiente página

Tabla 1 – continuación de la página anterior

N.	Pregunta	Respuesta	Observación
7	¿Qué medidas de seguridad y privacidad de datos considera esenciales para garantizar el manejo adecuado de la información sensible relacionada con incidentes delictivos?		
8	¿Cómo cree que un sistema de análisis de incidentes delictivos podría mejorar la colaboración y coordinación entre diferentes unidades o departamentos dentro de la policía?		
9	¿Qué nivel de capacitación considera necesario para que el personal del departamento de policía pueda utilizar eficazmente un sistema de análisis de incidentes delictivos?		

Anexo C. Script en Node.js para la comparación de los datos georreferenciados



```
import Exceljs from "exceljs";
import distances from "./distances.json" with { type: "json" };
import { haversineDistance } from "./haversineDistance.js";
import { roundDecimal } from "./lib/roundDecimal.js";
const distancesArray = [];
distances.forEach((d) => {
    const center = { lat: d.realLatitude, lng: d.realLongitude };
    const location = { lat: d.latitude, lng: d.longitude };
    const distance = haversineDistance(center, location);
    distancesArray.push({ distance: roundDecimal(distance) });
});
const newWorkbook = new Exceljs.Workbook();
const sheet = newWorkbook.addWorksheet("geo-distances");
sheet.addRow(["#", "Latitude", "Longitude", "Haversine distance from center (m)"]);
distancesArray.forEach((distance, index) => {
    sheet.addRow([index + 1, distances[index].latitude, distances[index].longitude, distance.distance]);
});
newWorkbook.xlsx.writeFile("./geo-locations.xlsx");
```

Anexo D. Scrip en Node.js para el calculo de la distancia entre dos puntos georreferenciados

```javascript
function haversineDistance(point1, point2) {
  const R = 6378137; // Radius of the Earth in meters
  const rlat1 = point1.lat * (Math.PI / 180); // Convert degrees to radians
  const rlat2 = point2.lat * (Math.PI / 180); // Convert degrees to radians
  const difflat = rlat2 - rlat1; // Radian difference (latitudes)
  const difflon = (point2.lng - point1.lng) * (Math.PI / 180); // Radian difference (longitudes)
  const d = 2 * R * Math.asin( Math.sqrt(
       Math.sin(difflat / 2) * Math.sin(difflat / 2) +
       Math.cos(rlat1) *
       Math.cos(rlat2) *
       Math.sin(difflon / 2) *
       Math.sin(difflon / 2)
     )
    );
  return d;
}
```

Anexo E. Script en Python para la creación de clusters de puntos georreferenciados

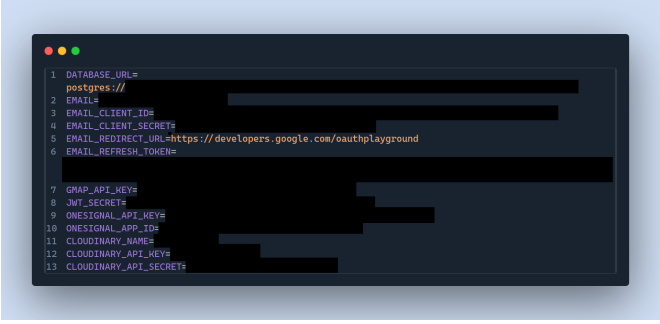
```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import pygments as pg

df = pd.read_excel('geo-locations.xlsx', usecols='B:D', skiprows=0)
df.columns = ['latitude', 'longitude', 'distance']

# Elbow Method
def elbow_method(data):
    wcss = []
    for i in range(1, 11):
        kmeans = KMeans(n_clusters=i, max_iter=300)
        kmeans.fit(df[['latitude', 'longitude']])
        wcss.append(kmeans.inertia_)
    plt.plot(range(1, 11), wcss)
    plt.title('Método del Codo')
    plt.xlabel('Número de clusters')
    plt.ylabel('WCSS (Within Cluster Sum of Squares)')
    plt.show()

elbow_method(df)

# Cluster optimal number
numero_optimo_clusters = 2

# apply K-Means
kmeans = KMeans(n_clusters=numero_optimo_clusters, max_iter=300, random_state=42)
kmeans.fit(df[['latitude', 'longitude']])
df['cluster'] = kmeans.labels_

# Calc the mean distance of each cluster
distancias_medias_clusters = df.groupby('cluster')['distance'].mean()
print("Distancias medias de cada cluster:")
print(distancias_medias_clusters)

# Calc the mean of the mean distances of the clusters
media_de_las_medias = round(distancias_medias_clusters.mean(), 2)
print("Media de las distancias medias de los clusters:")
print(media_de_las_medias)

# Graph
plt.scatter(df['latitude'], df['longitude'], c=df['cluster'], cmap='viridis')
plt.title('Clusters de Puntos GPS')
plt.xlabel('Latitud')
plt.ylabel('Longitud')
plt.colorbar(label='Cluster')
plt.show()
```

Anexo F. Archivo package.json con las dependencias de la API REST



```
"dependencies": {
  "@fastify/multipart": "^8.2.0",
  "@fastify/static": "^7.0.4",
  "@fastify/swagger": "^8.14.0",
  "@googlemaps/google-maps-services-js": "^3.4.0",
  "@nestjs-cls/transactional": "^2.3.1",
  "@nestjs-cls/transactional-adapter-prisma": "^1.2.1",
  "@nestjs-modules/mailer": "^2.0.2",
  "@nestjs/cache-manager": "^2.2.2",
  "@nestjs/common": "^10.3.9",
  "@nestjs/config": "^3.2.2",
  "@nestjs/core": "^10.3.9",
  "@nestjs/event-emitter": "^2.0.4",
  "@nestjs/jwt": "^10.2.0",
  "@nestjs/mapped-types": "*",
  "@nestjs/platform-express": "^10.3.9",
  "@nestjs/platform-fastify": "^10.3.9",
  "@nestjs/platform-socket.io": "^10.3.9",
  "@nestjs/serve-static": "^4.0.2",
  "@nestjs/swagger": "^7.3.1",
  "@nestjs/websockets": "^10.3.9",
  "@onesignal/node-onesignal": "^5.0.0-alpha-01",
  "@prisma/client": "^5.15.0",
  "bcrypt": "^5.1.1",
  "buffer-to-stream": "^1.0.0",
  "class-transformer": "^0.5.1",
  "class-validator": "^0.14.1",
  "cloudinary": "^2.2.0",
  "css-inline": "^0.11.2",
  "fastify": "^4.26.2",
  "fastify-file-interceptor": "^1.0.9",
  "geometric": "^2.5.4",
  "googleapis": "^139.0.0",
  "handlebars": "^4.7.8",
  "morgan": "^1.10.0",
  "nestjs-cls": "^4.3.0",
  "nodemailer": "^6.9.13",
  "reflect-metadata": "^0.2.2",
  "rxjs": "^7.8.1",
  "socket.io": "^4.7.5"
}
```

Anexo G. Archivo de configuración .env de la API REST



```
DATABASE_URL=
postgres://
EMAIL=
EMAIL_CLIENT_ID=
EMAIL_CLIENT_SECRET=
EMAIL_REDIRECT_URL=https://developers.google.com/oauthplayground
EMAIL_REFRESH_TOKEN=
GMAP_API_KEY=
JWT_SECRET=
ONESIGNAL_API_KEY=
ONESIGNAL_APP_ID=
CLOUDINARY_NAME=
CLOUDINARY_API_KEY=
CLOUDINARY_API_SECRET=
```

Anexo H. Guard de seguridad de la API REST



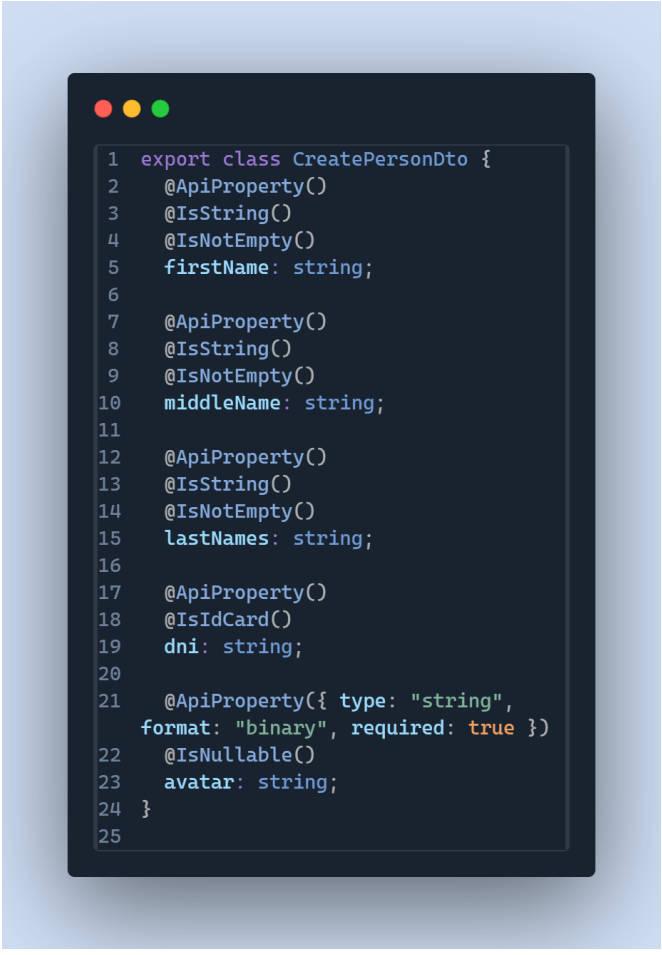
```
export class AuthGuard implements CanActivate {
  constructor(
    private jwtService: JwtService,
    private readonly configService: ConfigService,
    private reflector: Reflector
  ) {}

  async canActivate(context: ExecutionContext): Promise<boolean> {
    const isPublic = this.reflector.getAllAndOverride<boolean>(
IS_PUBLIC_KEY, [
      context.getHandler(),
      context.getClass()
    ]);

    if (isPublic) return true;

    const request = context.switchToHttp().getRequest();
    const token = this.extractTokenFromHeader(request);

    if (!token) {
      throw new HttpErrorException(GLOBAL_ERRORS_MESSAGES.
UnauthorizedException, HttpStatus.UNAUTHORIZED);
    }

    const payload = await this.validateToken(token);
    const admittedRoles = this.reflector.getAllAndOverride<AppRoles
[]>(ROLES_KEY, [
      context.getHandler(),
      context.getClass()
    ]);

    if (this.isAllowedForAllLoggedUsers(admittedRoles)) {
      return true;
    }

    if (!this.hasAdmittedRole(payload.roles, admittedRoles)) {
      throw new HttpErrorException(GLOBAL_ERRORS_MESSAGES.
ForbiddenException, HttpStatus.FORBIDDEN);
    }

    return true;
  }

  private isAllowedForAllLoggedUsers(admittedRoles: AppRoles[]):
boolean {
    return admittedRoles.includes("ALL_LOGGED_USERS");
  }

  private hasAdmittedRole(userRoles: Roles[], admittedRoles:
AppRoles[]): boolean {
    return userRoles.some((role) => admittedRoles.includes(role.
roleName));
  }

  private extractTokenFromHeader(request: Request): string |
undefined {
    const [type, token] = request.headers.authorization?.split(" ")
?? [];
    return type === "Bearer" ? token : undefined;
  }

  private async validateToken(token: string): Promise<LoginPayload>
{
    try {
      const payload = await this.jwtService.verifyAsync<LoginPayload
>(token, {
        secret: this.configService.get("JWT_SECRET")
      });
      return payload;
    } catch {
      throw new HttpErrorException(GLOBAL_ERRORS_MESSAGES.
UnauthorizedException, HttpStatus.UNAUTHORIZED);
    }
  }
}
```

Anexo I. Parte del modelo de datos de la API REST



```
generator client {
  provider        = "prisma-client-js"
  previewFeatures = ["postgresqlExtensions"]
}

datasource db {
  provider   = "postgresql"
  url        = env("DATABASE_URL")
  extensions = [postgis]
}

model Person {
  personID          Int               @id @default(autoincrement())
  firstName         String
  middleName        String
  lastNames         String
  dni               String            @unique
  avatar            String
  personDisability PersonDisability[]
  personInfo        PersonInfo?
  user              User?
}

model Disability {
  disabilityID     Int               @id @default(autoincrement())
  disabilityName   String            @unique
  disabilityDesc   String
  state            Int               @default(1)
  Incident         Incident[]
  personDisability PersonDisability[]
}
```

Anexo J. Validación de datos de la API REST



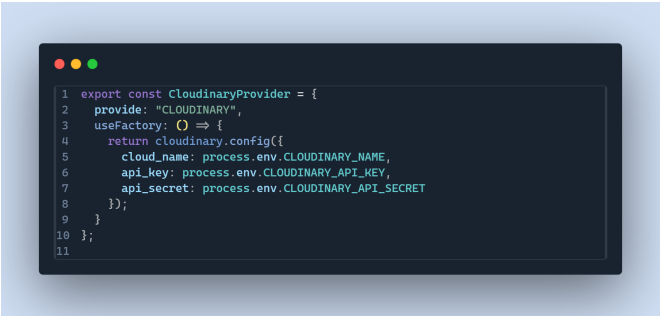
```
export class CreatePersonDto {
  @ApiProperty()
  @IsString()
  @IsNotEmpty()
  firstName: string;

  @ApiProperty()
  @IsString()
  @IsNotEmpty()
  middleName: string;

  @ApiProperty()
  @IsString()
  @IsNotEmpty()
  lastNames: string;

  @ApiProperty()
  @IsIdCard()
  dni: string;

  @ApiProperty({ type: "string", format: "binary", required: true })
  @IsNullable()
  avatar: string;
}
```

Anexo K. Envío de correos electrónicos de la API REST



```
export class MailingService {
  constructor(
    private readonly configService: ConfigService,
    private readonly mailerService: MailerService
  ) {}

  private async setTransport() {
    const oauth2Client = new Auth.OAuth2Client(
      this.configService.get("EMAIL_CLIENT_ID"),
      this.configService.get("EMAIL_CLIENT_SECRET"),
      this.configService.get("EMAIL_REDIRECT_URL")
    );

    oauth2Client.setCredentials({
      refresh_token: this.configService.get("EMAIL_REFRESH_TOKEN")
    });

    const accessToken: string = await new Promise((resolve, reject) => {
      oauth2Client.getAccessToken((err, token) => {
        if (err) {
          reject("Failed to create access token");
        }
        resolve(token);
      });
    });

    const config: SMTPTransportOptions = {
      service: "gmail",
      auth: {
        type: "OAuth2",
        user: this.configService.get("EMAIL"),
        clientId: this.configService.get("EMAIL_CLIENT_ID"),
        clientSecret: this.configService.get("EMAIL_CLIENT_SECRET"),
        accessToken
      }
    };

    this.mailerService.addTransporter("gmail", config);
  }

  public async sendMail({
    email,
    template,
    subject,
    context
  }: {
    email: string;
    subject: string;
    template: string;
    context: { [name: string]: string };
  }) {
    await this.setTransport();
    return await this.mailerService.sendMail({
      transporterName: "gmail",
      to: email,
      from: "noreplyvivovivo@gmail.com",
      subject: subject,
      template: template,
      context: {
        host: this.configService.get("HOST"),
        ...context
      }
    });
  }
}

```

Anexo L. Configuración de Cloudinary en la API REST



```
export const CloudinaryProvider = {
  provide: "CLOUDINARY",
  useFactory: () => {
    return cloudinary.config({
      cloud_name: process.env.CLOUDINARY_NAME,
      api_key: process.env.CLOUDINARY_API_KEY,
      api_secret: process.env.CLOUDINARY_API_SECRET
    });
  }
};

```

Anexo M. Subida de imágenes a Cloudinary desde la API REST



```
export class CloudinaryService {
  uploadFile(file: Express.Multer.File, options: UploadApiOptions) {
    return new Promise<CloudinaryResponse>((resolve, reject) => {
      cloudinary.uploader
        .upload_stream(options, (error, result) => {
          if (result) resolve(result);
          else reject(error);
        })
        .end(file.buffer);
    });
  }

  deleteFile(fileName: string) {
    const public_id = fileName.split(".")[0];
    return new Promise<CloudinaryResponse>((resolve, reject) => {
      cloudinary.uploader.destroy(public_id, (error, result) => {
        if (result) resolve(result);
        else reject(error);
      });
    });
  }

  findFile(fileName: string) {
    const public_id = fileName.split(".")[0];
    return new Promise<CloudinaryResponse>((resolve, reject) => {
      cloudinary.api.resource(public_id, (error, result) => {
        if (result) resolve(result);
        else reject(error);
      });
    });
  }
}

```

Anexo N. Subida de imágenes de usuario a Cloudinary desde la API REST



```

  async create(createPersonDto: CreatePersonDto, avatarImg: Express.Multer.File) {
    const { dni } = createPersonDto;
    const foundPerson = await this.findByDNI(dni);
    if (foundPerson)
      throw new HttpErrorException(PERSON_ERROR_RESPONSE.PERSON_ALREADY_EXISTS, HttpStatus.BAD_REQUEST);
    const uploadAvatar = await this.cloudinaryService.uploadFile(avatarImg, {
      folder: "avatars",
      format: "jpg"
    });
    const avatar = `${uploadAvatar.public_id}.${uploadAvatar.format}`;
    const newPerson = await this.prismaHost.tx.person.create({
      data: { ...createPersonDto, avatar }
    });
    if (!newPerson)
      throw new HttpErrorException(PERSON_ERROR_RESPONSE.PERSON_NOT_CREATED, HttpStatus.BAD_REQUEST);
    return newPerson;
  }
```

Anexo O. Plantilla de correo electrónico de la API REST



```
<html lang="es">

  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  </head>

  <body>
    <div class="header">
      <p>Recuperar contraseña</p>
    </div>
    <main class="main">
      <p class="message">Su contraseña provisional es: <span class="password">{{password}}</span> </p>
      <p class="message">Tenga en cuenta que esta es solo una contraseña provisional generada, por favor
        asegúrese de cambiarla después</p>
    </main>
  </body>

</html>

<style>
  .header { background-color: #70ADA7; padding: 1rem; display: flex; border-radius: 1rem; font-size: medium;
  color: white; font-weight: bold; align-items: center; } .header p { margin-right: auto; } .header img {
  margin-left: auto; height: 4rem; } .main { height: 3rem; width: 100%; } .message { text-align: center;
  font-size: 1rem; } .password { font-weight: bold; color: white; background-color: #167537; padding: .3rem; }
</style>
```

Anexo P. Configuración de OneSignal en la API REST



```
export const createOneSignalConfiguration = (restApiKey: string) => {
  const app_key_provider = {
    getToken() {
      return restApiKey;
    }
  };
  const oneSignalConfiguration = createConfiguration({
    authMethods: {
      rest_api_key: {
        tokenProvider: app_key_provider
      }
    }
  });
  return oneSignalConfiguration;
};

```

Anexo Q. Envío de notificaciones a través de OneSignal desde la API REST



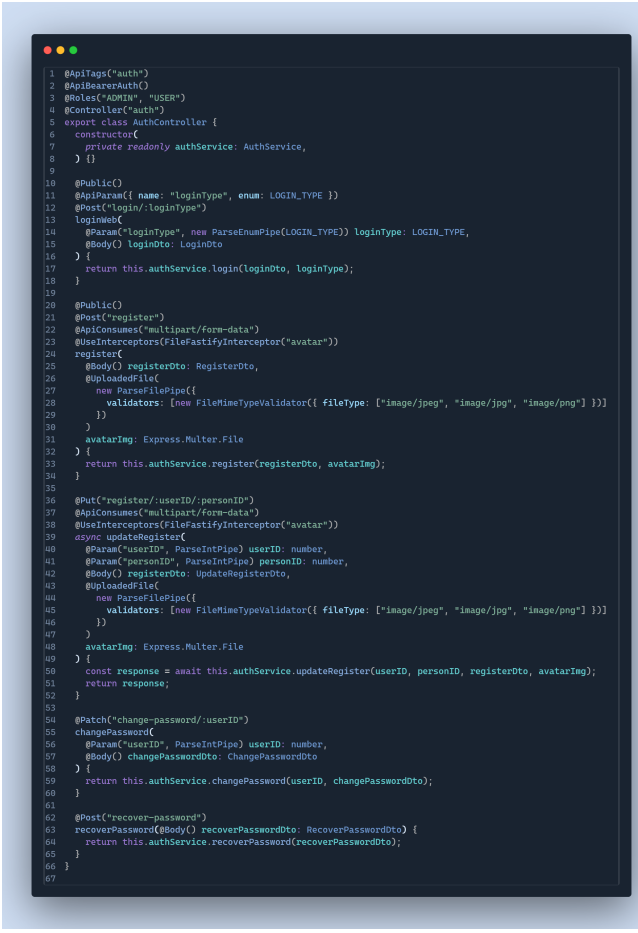
```
export class PushNotificationService {
  private imageUrl =
    "https://www.cloudways.com/blog/wp-content/uploads/MapPress-Easy-Google-Map-Plugin.jpg" as const;

  constructor(
    private readonly oneSignalService: OneSignalService,
    private readonly familyGroupService: FamilyGroupService,
    private readonly policeGroupService: PoliceGroupService,
    private readonly userRoleService: UserRoleService
  ) {}

  async sendNotificationFamilyGroup(sendNotificationFamilyGroupDto: SendNotificationFamilyGroupDto) {
    const { userID, names } = sendNotificationFamilyGroupDto;
    const { idsFamilyGroup } = await this.familyGroupService.findFamilyGroupOneSignalIDsByUserID(userID);
    const { policeOneSignalIDs } = await this.policeGroupService.findPoliceOneSignalIDsByUserMemberID(userID);
    const ids = [...idsFamilyGroup, ...policeOneSignalIDs];
    const notification = this.oneSignalService.notificationBuilder
      .setContents({
        en: `${names} se encuentra en peligro, presiona para ver su ubicación`,
        es: `${names} se encuentra en peligro, presiona para ver su ubicación`
      })
      .setIncludePlayerIds(ids)
      .setContentAvailable(true)
      .setHeadings({ en: "¡Alert de Peligro!", es: "¡Alerta de Peligro!" })
      .setBigPicture(this.imageUrl)
      .setSmallIcon("ic_notification_icon")
      .setIosAttachments({ id1: this.imageUrl })
      .build();
    const { errors } = await this.oneSignalService.client.createNotification(notification);
    if (errors) throw new HttpErrorException("NOTIFICATION_NOT_SENT", HttpStatus.BAD_REQUEST);
  }
}

```

Anexo R. Ejemplo de módulo de la API REST



```
@Module({
  imports: [
    JwtModule.registerAsync({
      inject: [ConfigService],
      useFactory: async (configService: ConfigService) => {
        return {
          secret: configService.get<string>("JWT_SECRET")
        };
      }
    }),
    PersonModule,
    PersonInfoModule,
    PersonDisabilityModule,
    UserModule,
    RoleModule,
    UserRoleModule,
    MailingModule
  ],
  controllers: [AuthController],
  providers: [AuthService, AuthGuard]
})
export class AuthModule {}

```

Anexo S. Ejemplo de controlador de la API REST



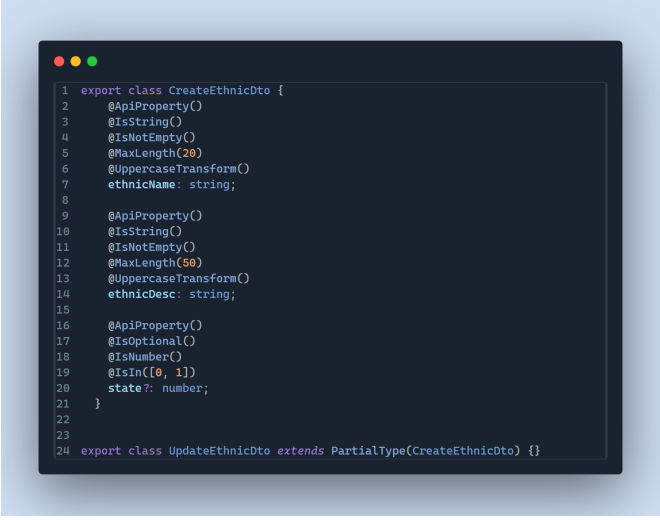
```
@ApiTags("auth")
@ApiBearerAuth()
@Roles("ADMIN", "USER")
@Controller("auth")
export class AuthController {
  constructor(
    private readonly authService: AuthService,
  ) {}

  @Public()
  @ApiParam({ name: "loginType", enum: LOGIN_TYPE })
  @Post("login/:loginType")
  loginWeb(
    @Param("loginType", new ParseEnumPipe(LOGIN_TYPE)) loginType: LOGIN_TYPE,
    @Body() loginDto: LoginDto
  ) {
    return this.authService.login(loginDto, loginType);
  }

  @Public()
  @Post("register")
  @ApiConsumes("multipart/form-data")
  @UseInterceptors(FileFastifyInterceptor("avatar"))
  register(
    @Body() registerDto: RegisterDto,
    @UploadedFile(
      new ParseFilePipe({
        validators: [new FileMimeTypeValidator({ fileType: ["image/jpeg", "image/jpg", "image/png"] })]
      })
    )
    avatarImg: Express.Multer.File
  ) {
    return this.authService.register(registerDto, avatarImg);
  }

  @Put("register/:userID/:personID")
  @ApiConsumes("multipart/form-data")
  @UseInterceptors(FileFastifyInterceptor("avatar"))
  async updateRegister(
    @Param("userID", ParseIntPipe) userID: number,
    @Param("personID", ParseIntPipe) personID: number,
    @Body() registerDto: UpdateRegisterDto,
    @UploadedFile(
      new ParseFilePipe({
        validators: [new FileMimeTypeValidator({ fileType: ["image/jpeg", "image/jpg", "image/png"] })]
      })
    )
    avatarImg: Express.Multer.File
  ) {
    const response = await this.authService.updateRegister(userID, personID, registerDto, avatarImg);
    return response;
  }

  @Patch("change-password/:userID")
  changePassword(
    @Param("userID", ParseIntPipe) userID: number,
    @Body() changePasswordDto: ChangePasswordDto
  ) {
    return this.authService.changePassword(userID, changePasswordDto);
  }

  @Post("recover-password")
  recoverPassword(@Body() recoverPasswordDto: RecoverPasswordDto) {
    return this.authService.recoverPassword(recoverPasswordDto);
  }
}
```

Anexo T. Ejemplo de servicio de la API REST



```
@Injectable()
export class EthnicService {
  constructor(private readonly prismaHost: TransactionHost<TransactionalAdapterPrisma>) {}

  async create(createEthnicDto: CreateEthnicDto) {
    const foundEthnic = await this.prismaHost.tx.ethnic.findFirst({
      where: { ethnicName: createEthnicDto.ethnicName }
    });
    if (foundEthnic)
      throw new HttpErrorException(ETHNIC_ERROR_RESPONSE.ETHNIC_ALREADY_EXISTS, HttpStatus.BAD_REQUEST);
    const newEthnic = await this.prismaHost.tx.ethnic.create({ data: createEthnicDto });
    if (!newEthnic)
      throw new HttpErrorException(ETHNIC_ERROR_RESPONSE.ETHNIC_NOT_CREATED, HttpStatus.BAD_REQUEST);
    return newEthnic;
  }

  async findAll() {
    const foundEthnics = await this.prismaHost.tx.ethnic.findMany({
      where: { state: 1 },
      select: {
        ethnicID: true,
        ethnicName: true
      }
    });

    return foundEthnics.map((ethnic) => ({
      id: ethnic.ethnicID,
      name: ethnic.ethnicName
    }));
  }

  async findByID(ethnicID: number) {
    const foundEthnic = await this.prismaHost.tx.ethnic.findFirst({ where: { ethnicID, state: 1 } });
    if (!foundEthnic)
      throw new HttpErrorException(ETHNIC_ERROR_RESPONSE.ETHNIC_NOT_FOUND, HttpStatus.BAD_REQUEST);
    return foundEthnic;
  }

  async update(ethnicID: number, updateEthnicDto: UpdateEthnicDto) {
    const foundEthnic = await this.prismaHost.tx.ethnic.findUnique({ where: { ethnicID } });
    if (!foundEthnic)
      throw new HttpErrorException(ETHNIC_ERROR_RESPONSE.ETHNIC_NOT_FOUND, HttpStatus.BAD_REQUEST);
    const updatedEthnic = await this.prismaHost.tx.ethnic.update({
      data: updateEthnicDto,
      where: { ethnicID }
    });
    if (!updatedEthnic)
      throw new HttpErrorException(ETHNIC_ERROR_RESPONSE.ETHNIC_NOT_UPDATED, HttpStatus.BAD_REQUEST);
    return updatedEthnic;
  }

  async remove(ethnicID: number): Promise<Ethnic> {
    const foundEthnic = await this.prismaHost.tx.ethnic.findUnique({ where: { ethnicID, state: 1 } });
    if (!foundEthnic)
      throw new HttpErrorException(ETHNIC_ERROR_RESPONSE.ETHNIC_NOT_FOUND, HttpStatus.NOT_FOUND);
    const updatedEthnic = await this.prismaHost.tx.ethnic.update({ data: { state: 0 }, where: { ethnicID } });
    return updatedEthnic;
  }
}
```

Anexo U. Ejemplo de DTO de la API REST



```
export class CreateEthnicDto {
    @ApiProperty()
    @IsString()
    @IsNotEmpty()
    @MaxLength(20)
    @UppercaseTransform()
    ethnicName: string;

    @ApiProperty()
    @IsString()
    @IsNotEmpty()
    @MaxLength(50)
    @UppercaseTransform()
    ethnicDesc: string;

    @ApiProperty()
    @IsOptional()
    @IsNumber()
    @IsIn([0, 1])
    state?: number;
  }


export class UpdateEthnicDto extends PartialType(CreateEthnicDto) {}
```

Anexo V. Dependencias de la aplicación web



```
"dependencies": {
    "@mantine/core": "^7.9.1",
    "@mantine/dates": "^7.9.1",
    "@mantine/form": "^7.9.1",
    "@mantine/hooks": "^7.9.1",
    "@mantine/modals": "^7.9.1",
    "@tabler/icons-react": "^3.5.0",
    "@tanstack/react-query": "^5.40.1",
    "@tanstack/react-query-devtools": "^5.40.1",
    "@vis.gl/react-google-maps": "^1.0.2",
    "axios": "^1.7.2",
    "clsx": "^2.1.1",
    "dayjs": "^1.11.11",
    "mantine-react-table": "^2.0.0-beta.3",
    "next": "14.2.3",
    "next-auth": "^4.24.7",
    "react": "^18",
    "react-dom": "^18",
    "socket.io-client": "^4.7.5",
    "sonner": "^1.4.41",
    "tailwind-merge": "^2.3.0",
    "zod": "^3.23.8",
    "zustand": "^4.5.2"
  }
```

Anexo W. Archivo de configuración .env.local de la aplicación web



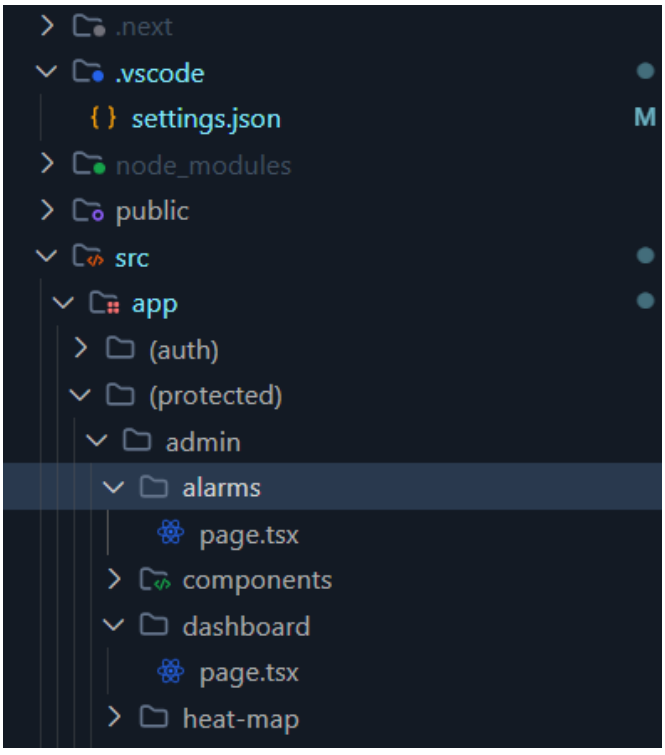
```
NEXTAUTH_SECRET=gGYzHFSSrsr2
NEXTAUTH_URL=                /api/auth
NEXT_PUBLIC_API_URL=                                    /api
NEXT_PUBLIC_SOCKET_URL=
NEXT_PUBLIC_IMG_URL=                                /upload/v1709416960
NEXT_PUBLIC_GMAP_API_KEY=
```

Anexo X. Configuración de la aplicación web



```tsx
export default function Providers({ children }: { children: ReactNode }) {
  const [queryClient] = useState(
    () =>
      new QueryClient({
        defaultOptions: {
          queries: {
            refetchOnWindowFocus: false,
            refetchOnMount: false,
            gcTime: 1000 * 3600 // 1 hour cache invalidation
          }
        }
      })
  );

  return (
    <SessionProvider>
      <QueryClientProvider client={queryClient}>
        <ReactQueryDevtools initialIsOpen={false} />
        <MantineProvider theme={theme}>
          <ModalsProvider labels={{ cancel: "Cancelar", confirm: "Confirmar" }}>
            <Toaster richColors position="bottom-center" />
            {children}
          </ModalsProvider>
        </MantineProvider>
      </QueryClientProvider>
    </SessionProvider>
  );
}

```

Anexo Y. Rutas de la aplicación web



Anexo Z. Validación de datos de la aplicación web



```zod
import { z } from "zod";

export const loginSchema = z.object({
  email: z.string().email({ message: "Correo invalido" }),
  password: z.string().min(8, { message: "La contraseña es muy corta" })
});

export type LoginSchema = z.infer<typeof loginSchema>;

export const loginInitialValues: LoginSchema = {
  email: "",
  password: ""
};
```

Anexo AA. Guardar token de usuario en la aplicación web



```nextauth
export const authOptions: NextAuthOptions = {
  secret: process.env.AUTH_SECRET,
  session: {
    strategy: "jwt",
    maxAge: 14 * 24 * 60 * 60 // 14 days in seconds
  },
  pages: {
    signIn: "/login"
  },
  providers: [
    CredentialsProvider({
      name: "credentials",
      credentials: {
        email: { label: "Correo", type: "text", placeholder: "jsmith" },
        password: { label: "Contraseña", type: "password" }
      },
      async authorize(credentials, req) {
        if (!credentials?.email || !credentials?.password) return null;
        try {
          const url = `${process.env.NEXT_PUBLIC_API_URL}/auth/login/web`;
          const res = await fetch(url, {
            body: JSON.stringify(credentials),
            method: "POST",
            headers: {
              "Content-Type": "application/json"
            }
          });
          const data = await res.json();
          if (data.error) throw new Error(data.error);
          return data;
        } catch (error: any) {
          if (error.cause?.code === "ECONNREFUSED" || error.cause?.code === "ERR_NETWORK")
            throw new Error("Error de conexión, revise su conexión a internet");
          throw new Error(error.message);
        }
      }
    })
  ],
  callbacks: {
    jwt({ token, user }) {
      if (user) {
        token = { ...token, ...user };
      }
      return token;
    },
    session({ token, session }) {
      session.user = token.user;
      session.token = token.token;
      return session;
    }
  }
};
```

Anexo AB. Dependecias de la aplicación móvil



```yaml
dependencies:
  flutter:
    sdk: flutter
  onesignal_flutter: ^5.0.3
  shared_preferences: ^2.2.2
  provider: ^6.0.5
  awesome_snackbar_content: ^0.1.2
  flutter_svg: ^1.0.1
  vibration: ^1.8.3
  geolocator: ^10.1.0
  geocoding: ^2.1.1
  modal_bottom_sheet: ^2.1.2
  gap: ^3.0.1
  image_picker: ^1.0.4
  google_maps_flutter: ^2.5.0
  location_platform_interface: ^3.1.2
  flutter_google_places: ^0.3.0
  flutter_localization: ^0.1.14
  socket_io_client: ^2.0.3+1
  badges: ^3.1.2
  cached_network_image: ^3.3.0
  simple_ripple_animation: ^0.0.7
  location: ^5.0.3
  flutter_dotenv: ^5.1.0
  custom_marker: ^1.0.0
  lottie: ^2.7.0
  clipboard: ^0.1.3
  dio: ^5.3.3

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  flutter_localizations:
    sdk: flutter
  intl: ^0.18.1
  http_parser: ^4.0.2
```

Anexo AC. configuracion-env-movil



```
BASE_URL=[redacted]/api/
HOST=[redacted]
API_ONE_SIGNAL=[redacted]
API_KEY_GOOGLE=[redacted]
PUBLIC_IMG_URL=[redacted]
```

Anexo AD. Configuración de la aplicación móvil



```dart
Future main() async {
  await dotenv.load();
  DioSingleton.getInstance();
  await SharedPrefs().init();

  runApp(MultiProvider(
    providers: [
      ChangeNotifierProvider(create: (_) => UserProvider()),
      ChangeNotifierProvider(create: (_) => AlarmStateProvider()),
      ChangeNotifierProvider(create: (_) => GeoLocationProvider()),
      ChangeNotifierProvider(create: (_) => SocketProvider()),
    ],
    child: const MyApp(),
  ));
}

```

Anexo AE. Guardar token de usuario en la aplicación móvi



```dart
class LoginController {
  late final Function(bool state) onStateConnect;
  late ApiRepositoryUserImpl userService;
  late UserProvider userProvider;
  late BuildContext context;

  LoginController(BuildContext newContext) {
    context = newContext;
    userService = ApiRepositoryUserImpl();
    userProvider = context.read<UserProvider>();
  }

  Future<void> showHomePage(String userName, String password) async {
    var res = await userService.getUser(createCredentials(userName, password));
    if (res == null || res.error) return;
    final UserPrefProvider user = UserPrefProvider.fromJson(res.data);
    userProvider.getUser(user.getUser, user.getToken);
    saveCredentials(user.getUser, user.getToken);
    Navigator.of(context).pushReplacementNamed(HomeView.id);
  }

  Future<void> saveCredentials(UserAuth user, String token) async {
    var userString = userAuthToJson(user);
    SharedPrefs().user = userString;
    SharedPrefs().token = token;
  }
}
```