

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA E INFORMÁTICA
CONCEPCIÓN - CHILE



**“Desarrollo del Frontend y Backend de la Aplicación
Móvil Vision Security: Una Solución Tecnológica para
la Seguridad Personal y Comunitaria”**

NICOLÁS ALFREDO SEPÚLVEDA BRIONES

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO EN INFORMÁTICA**

Profesor Guía: Javier Maldonado Carmona



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Desarrollo del Frontend y Backend de la Aplicación Móvil Vision Security: Una Solución Tecnológica para la Seguridad Personal y Comunitaria

Nombre del candidato(a): Nicolás Alfredo Sepúlveda Briones

Carrera / Grado: Ingeniería en informática

Campus: Concepción **Departamento:** Electrónico e informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Javier Maldonado Carmona, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 31/08/2026 **Firma:** _____

JAVIER ALEXIS
MALDONADO
CARMONA

Signed by: JAVIER ALEXIS
MALDONADO CARMONA
Date: 2026-08-31 11:50:19

Estudiante o Candidato(a):

Fecha: 14-08-2026

Firma: _____



Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Abril - 2026

DEDICATORIA

A mi querida familia,

Este trabajo es el resultado de años de esfuerzo, sacrificio, dedicación, y no podría haberlo logrado sin su incondicional apoyo y amor.

A mis padres, por enseñarme el valor del esfuerzo y la perseverancia, y por ser siempre mi guía en cada decisión importante de mi vida. Su ejemplo y sus palabras de aliento han sido mi mayor fuente de motivación.

A mi hermano Sebastián, por ser mi apoyo silencioso, por creer en mí cuando más lo necesitaba y por recordarme la importancia de mantenerme firme ante los retos, además de ser la persona que me aconsejó entrar en la carrera y estoy a pocos meses de titularme.

A toda mi familia, que con su amor y comprensión me brindó el espacio y la tranquilidad para enfocarme en este proyecto y durante toda la carrera que me propuse terminar.

Este trabajo no es solo mío, es el reflejo de los valores que ustedes me inculcaron y del amor que siempre me han dado.

Con todo mi cariño y gratitud,

Nicolás Alfredo Sepúlveda Briones

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que de una u otra forma contribuyeron al desarrollo y culminación de este trabajo.

A mis compañeros, por su colaboración y apoyo constante durante este proceso. Su disposición para compartir ideas, solucionar problemas juntos y superar los desafíos ha sido fundamental para el éxito de este proyecto. Este trabajo no habría sido posible sin ustedes.

A mis profesores, por su guía y dedicación en cada etapa de mi formación académica. En especial, quiero destacar a quienes participaron directamente en este proyecto, por su paciencia, sus consejos y por brindarme el conocimiento necesario para enfrentar los retos que encontré en el camino. Su compromiso con mi aprendizaje me inspira a buscar siempre las mejores prácticas.

A todos ustedes, mi más profundo agradecimiento. Su apoyo no solo fue clave para este logro, sino que también dejó una huella significativa en mi vida personal y profesional a futuro.

Con gratitud,
Nicolás Alfredo Sepúlveda Briones.

RESUMEN

El presente trabajo describe el desarrollo y validación de una solución tecnológica orientada a la seguridad personal y comunitaria mediante la creación de una aplicación móvil llamada Vision Security. Este proyecto incluyó la implementación de funcionalidades clave, como reportes interactivos de incidentes, alertas de seguridad en tiempo real y un sistema de gestión de datos alojados en servicios en la nube. Utilizando herramientas como Flutter para el desarrollo móvil, AWS (EC2, S3, RDS, VPC) para la infraestructura en la nube, y Mapbox para mapas interactivos, se logró una integración eficiente entre la experiencia de usuario y la tecnología.

Los resultados demuestran que la solución es funcional, logrando satisfacer las necesidades de los usuarios objetivos en términos de seguridad y facilidad de uso. Además, se validó su efectividad a través de pruebas y encuestas que reflejan un alto nivel de satisfacción con la aplicación. Este proyecto no solo representa un avance en el ámbito de la seguridad tecnológica, sino que también establece una base sólida para futuras innovaciones y mejoras en el sector.

Palabras clave: Vision Security, seguridad tecnológica, desarrollo móvil, AWS, sensores, reportes interactivos.

GLOSARIO

- **API (Application Programming Interface):** Conjunto de funciones y protocolos que permiten la comunicación entre diferentes sistemas o aplicaciones.
- **AWS (Amazon Web Services):** Plataforma de servicios en la nube que ofrece almacenamiento, bases de datos, servidores y otras herramientas para el desarrollo de aplicaciones.
- **Backend:** Parte del sistema encargada de la lógica de negocio, procesamiento de datos y comunicación con la base de datos.
- **Base de datos:** Sistema que permite almacenar, organizar y gestionar información de manera estructurada.
- **Cloud Computing (Computación en la nube):** Uso de servidores remotos a través de internet para almacenar, gestionar y procesar datos.
- **EC2 (Elastic Compute Cloud):** Servicio de AWS que permite crear y gestionar servidores virtuales en la nube.
- **Flutter:** Framework de desarrollo de aplicaciones móviles creado por Google, que permite construir interfaces de usuario utilizando el lenguaje Dart.
- **Dart:** Lenguaje de programación utilizado por Flutter para el desarrollo de aplicaciones móviles.
- **Frontend:** Parte visible de la aplicación con la que interactúa el usuario.
- **Geolocalización:** Tecnología que permite determinar la ubicación geográfica de un dispositivo en tiempo real.
- **GPS (Global Positioning System):** Sistema de posicionamiento global que permite obtener coordenadas geográficas precisas.
- **HTTP (Hypertext Transfer Protocol):** Protocolo utilizado para la comunicación entre cliente y servidor en aplicaciones web.
- **JSON (JavaScript Object Notation):** Formato ligero de intercambio de datos utilizado para transmitir información entre sistemas.
- **Mapbox:** Plataforma que proporciona herramientas para la creación de mapas interactivos y visualización de datos geográficos.
- **Flutter Map:** Librería utilizada en Flutter para integrar mapas interactivos dentro de la aplicación.
- **MapController:** Clase utilizada para controlar el comportamiento del mapa, como el movimiento o el zoom.
- **Marker:** Elemento visual que representa una ubicación específica dentro del mapa.
- **MarkerLayer:** Capa que agrupa y gestiona múltiples marcadores dentro del mapa.
- **LatLng:** Estructura de datos utilizada para representar coordenadas geográficas (latitud y longitud).
- **Geocoding (Geocodificación):** Proceso de convertir coordenadas geográficas en direcciones legibles.
- **Geolocator:** Librería utilizada para obtener la ubicación actual del dispositivo.

- **Placemark:** Representación de una ubicación geográfica con información detallada como calle, ciudad y país.
- **RDS (Relational Database Service):** Servicio de AWS para la gestión de bases de datos relacionales en la nube.
- **S3 (Simple Storage Service):** Servicio de almacenamiento en la nube de AWS utilizado para guardar archivos como imágenes.
- **VPC (Virtual Private Cloud):** Red virtual privada dentro de AWS que permite aislar y proteger los recursos en la nube.
- **Sensor (móvil):** Componente del dispositivo que permite detectar información del entorno, como movimiento o ubicación.
- **Giroscopio:** Sensor que permite detectar la orientación y rotación del dispositivo.
- **Acelerómetro:** Sensor que mide cambios de velocidad y movimientos bruscos del dispositivo.
- **Timer:** Mecanismo que permite ejecutar tareas repetitivas en intervalos de tiempo definidos.
- **SMTP (Simple Mail Transfer Protocol):** Protocolo utilizado para el envío de correos electrónicos.
- **Notificaciones:** Mensajes enviados al usuario para alertar sobre eventos o situaciones relevantes dentro de la aplicación.
- **Reporte de incidente:** Registro generado por un usuario que describe una situación de riesgo, incluyendo ubicación, tipo y detalles.
- **Modo Emergencia:** Funcionalidad que permite activar acciones automáticas ante situaciones de riesgo, como envío de alertas o detección de movimientos.
- **UI (User Interface):** Interfaz de usuario, es decir, el diseño visual con el que interactúa el usuario.
- **UX (User Experience):** Experiencia del usuario al interactuar con la aplicación.

INDICE DE CONTENIDOS

Contenido

RESUMEN.....	4
INDICE DE FIGURAS.....	10
INDICE DE TABLAS.....	11
INTRODUCCIÓN.....	12
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA.....	14
1.1 Contexto del problema.....	14
1.2 Contexto de la solución propuesta.....	16
1.3 Situación actual y actores involucrados.....	17
1.4 Justificación del problema y consecuencias de no solucionarlo.....	18
1.5 Competencia Existente.....	19
1.6 Objetivos.....	20
CAPÍTULO 2: MARCO CONCEPTUAL.....	21
2.1 Desarrollo de Aplicaciones Móviles.....	22
2.2 Arquitectura Backend para Aplicaciones Móviles.....	23
2.3 Servicios en la Nube (Cloud Computing).....	24
2.4 Geolocalización y Mapas Interactivos.....	25
2.5 Sensores de Dispositivos Móviles.....	26
2.6 Seguridad Ciudadana y Tecnología.....	27
CAPÍTULO 3: PROPUESTA DE SOLUCION.....	30
3.1 Metodología de Desarrollo.....	30
3.2 Arquitectura de la Solución.....	32
3.3 Proceso de Generación de Reportes de Seguridad.....	34
3.4 Creación y Gestión de Comunidades.....	34
3.5 Integración con Firebase.....	34
3.6 Tabla de Funcionalidades Implementadas.....	35
3.7 Desarrollo e Implementación del Sistema.....	36
3.7.1 Módulo de Autenticación de Usuarios.....	38
3.7.2 Módulo de Geolocalización y Mapa Interactivo.....	39
3.7.3 Módulo de Generación de Reportes.....	40
.....	41
3.7.4 Módulo de Notificaciones.....	42

3.7.5 Módulo de Emergencia.....	42
3.7.6 Módulo de Gestión de Cuenta.....	44
3.7.7 Módulo de Configuración.....	44
3.7.8 Integración con Backend y Servicios Externos.....	44
3.8 Diagrama de Casos de Uso del Sistema.....	44
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN.....	49
4.1 Metodología de Validación.....	49
4.2 Encuesta de Validación.....	52
4.3 Resultados.....	53
4.3.1 Facilidad de uso.....	53
4.3.2 Uso del sistema de reportes.....	54
4.3.3 Funcionalidades necesarias.....	55
4.3.4 Satisfacción con comunidades.....	56
4.3.5 Recomendación de la app.....	56
4.4 Validación funcional de los módulos principales.....	57
CAPÍTULO 5: CONCLUSIONES.....	60
5.1 Cumplimiento de los objetivos.....	60
5.2 Validación de la solución desarrollada.....	61
5.3 Limitaciones identificadas.....	61
5.4 Trabajo futuro.....	62
REFERENCIAS BIBLIOGRÁFICAS.....	63

INDICE DE FIGURAS

Imagen 1 .Arquitectura Inicial del sistema Vision Security.Fuente: Elaboración propia.....	33
Imagen 2 .Login y creación de cuenta. Fuente: Elaboración propia.....	39
Imagen 3. Mapa y Reportes Fuente: Elaboración propia.....	40
Imagen 4. Tipos y generación de reportes Fuente: Elaboración propia.....	41
Imagen 5. Apartado de emergencias Fuente: Elaboración propia.....	43
Imagen 6. Diagrama casos de uso Fuente: Elaboración propia.....	47

INDICE DE TABLAS

Tabla 1. Enfoques de desarrollo móvil y selección tecnológica del proyecto.....	21
Tabla 2. Endpoints backend utilizados en Vision Security.....	22
Tabla 3. Servicios AWS utilizados en el proyecto.....	23
Tabla 4. Componentes de geolocalización implementados.....	24
Tabla 5. Sensores y funcionalidades asociadas.....	25
Tabla 6. Relación código – funcionalidad en Vision Security.....	28
Tabla 7. Etapas de implementación y validación de la solución propuesta.....	30
Tabla 8. Funcionalidades. Fuente: Elaboración Propia.....	35
Tabla 9. Características generales de la actividad de validación.....	49
Tabla 10. Tareas ejecutadas por los participantes durante la validación.....	50
Tabla 11. Resultados de la validación funcional de los módulos implementados.....	57

INTRODUCCIÓN

En la actualidad, la seguridad ciudadana se ha convertido en una preocupación prioritaria, impulsada por el aumento de incidentes que generan una sensación de inseguridad entre las comunidades. Paralelamente, los avances tecnológicos en dispositivos móviles y servicios en la nube han abierto nuevas posibilidades para desarrollar herramientas innovadoras que respondan a estas problemáticas de manera eficiente y accesible. Este trabajo se centra en el desarrollo de una solución tecnológica que aborda la gestión de situaciones de emergencia y fomenta la colaboración comunitaria mediante una aplicación móvil.

El problema principal identificado es la carencia de herramientas accesibles, intuitivas y efectivas que permitan a los ciudadanos reportar incidentes de seguridad, organizarse en comunidades y tomar acciones inmediatas en situaciones críticas. Este problema afecta tanto a la prevención como a la respuesta ante emergencias, limitando el alcance de las iniciativas individuales y colectivas.

En respuesta a esta necesidad, se diseñó y desarrolló una aplicación móvil que integra diversas funcionalidades enfocadas en mejorar la seguridad personal y comunitaria. La propuesta incluye el desarrollo de un sistema que permite la generación y visualización de reportes en tiempo real, la creación de comunidades para fortalecer el apoyo mutuo, y la incorporación de sensores del dispositivo móvil para detectar movimientos bruscos y activar alertas automáticas. Además, se utilizaron servicios en la nube para garantizar la escalabilidad y seguridad de los datos.

El presente trabajo tiene como objetivo el diseño y desarrollo de una aplicación móvil orientada a la seguridad personal, denominada Vision Security. Para ello, se aborda la implementación de un sistema que integra un frontend desarrollado en Flutter, el cual permite ofrecer una interfaz accesible y fácil de utilizar para los usuarios.

En cuanto al soporte tecnológico, se emplean servicios en la nube mediante Amazon Web Services (AWS), utilizados para el almacenamiento y gestión de la información generada por la aplicación. Adicionalmente, se incorpora el uso de Mapbox para la visualización de datos georreferenciados, permitiendo representar los reportes de incidentes en un entorno interactivo. Por otro lado, Firebase es utilizado para la gestión de imágenes asociadas a dichos reportes.

De esta forma, la memoria se centra en describir el proceso de desarrollo e integración de estos componentes, dando como resultado una solución funcional que busca apoyar a los usuarios en situaciones de riesgo mediante herramientas tecnológicas accesibles.

La metodología empleada para este proyecto se basa en un enfoque ágil, específicamente en el marco de trabajo Scrum, el cual permite el desarrollo incremental e iterativo de software, facilitando la adaptación continua a los requerimientos y la mejora progresiva del sistema.

Este enfoque se caracteriza por dividir el desarrollo en ciclos cortos de trabajo (iteraciones), permitiendo validar constantemente las funcionalidades implementadas y realizar ajustes en función de los resultados obtenidos.

En este contexto, el desarrollo del proyecto se estructuró en las siguientes etapas:

1. **Análisis de necesidades:** Identificación de los requerimientos del sistema y del contexto en el que se implementaría.
2. **Diseño de la solución:** Creación de la arquitectura de la aplicación, seleccionando las tecnologías más adecuadas para cada funcionalidad.
3. **Desarrollo iterativo:** Implementación de las funcionalidades clave en ciclos de prueba y ajuste.
4. **Validación funcional:** Pruebas de la aplicación, evaluando su rendimiento, usabilidad y cumplimiento de los objetivos planteados.

La estructura de este documento está diseñada para guiar al lector a través de cada etapa del desarrollo del proyecto:

- **Capítulo 1:** Presenta el contexto general, el problema identificado y los objetivos específicos que guiaron el trabajo.
- **Capítulo 2:** Aborda el marco conceptual, proporcionando una base teórica y técnica sobre las tecnologías y metodologías utilizadas.
- **Capítulo 3:** Detalla el proceso de diseño e implementación, describiendo las decisiones clave y las herramientas empleadas.
- **Capítulo 4:** Expone los resultados obtenidos y la validación de la solución, evaluando su impacto y efectividad.
- **Capítulo 5:** Ofrece las conclusiones del trabajo, reflexionando sobre las lecciones aprendidas y proponiendo mejoras o extensiones futuras.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

La seguridad ciudadana representa una de las principales preocupaciones en la actualidad debido al aumento progresivo de hechos delictivos en contextos urbanos y rurales. Esta situación no solo afecta la calidad de vida de las personas, sino que también limita su movilidad, genera percepción de inseguridad y disminuye la confianza en las instituciones encargadas de garantizar el orden público. La problemática es global y transversal, manifestándose con mayor gravedad en países que presentan desigualdades sociales y recursos limitados para enfrentarla.

En Chile, los indicadores actuales reflejan un escenario preocupante. De acuerdo con la **Encuesta Nacional Urbana de Seguridad Ciudadana (ENUSC, 2023)**, el 60% de la población percibe que la delincuencia ha aumentado en el último año, mientras que un 25% declara haber sido víctima directa de algún delito, principalmente robos o asaltos en la vía pública. Este escenario es agravado por la falta de herramientas tecnológicas efectivas que permitan una comunicación rápida entre ciudadanos, autoridades y la comunidad.

1.1 Contexto del problema

La inseguridad ciudadana es un fenómeno complejo que afecta la calidad de vida de las personas y genera un ambiente de desconfianza en la sociedad. En la actualidad, este problema se manifiesta de forma especialmente visible en los espacios públicos, donde los ciudadanos perciben una mayor exposición a situaciones como robos, asaltos, agresiones o episodios de acoso. Diversos estudios han señalado que la vía pública constituye uno de los principales escenarios donde se concentran este tipo de delitos, lo que incrementa la percepción de riesgo en la población (Subsecretaría de Prevención del Delito, 2023). Estas experiencias generan un impacto significativo tanto en la seguridad física como emocional de las personas, debilitando la cohesión social y fomentando un sentimiento generalizado de vulnerabilidad.

Uno de los principales desafíos para abordar esta problemática radica en la falta de mecanismos efectivos y accesibles para denunciar y prevenir incidentes. Diversos estudios han evidenciado que las herramientas disponibles presentan limitaciones en términos de accesibilidad, integración tecnológica y participación ciudadana (Banco Interamericano de Desarrollo, 2020). A continuación, se detallan algunos aspectos clave que ilustran las limitaciones de las soluciones actuales:

1. Sistemas Tradicionales de Denuncia:

El proceso de denuncia en muchos países sigue siendo lento, burocrático y poco accesible para amplios sectores de la población. La mayoría de las denuncias deben realizarse en dependencias policiales o judiciales, lo que implica un gasto significativo de tiempo y recursos para las víctimas. Esta situación no solo desincentiva a las personas a denunciar delitos menores, sino que también contribuye a la subestimación estadística de los índices de criminalidad, dificultando el diseño de políticas públicas basadas en datos confiables (Oficina de las Naciones Unidas contra la Droga y el Delito, 2019).

2. Falta de Tecnología Preventiva:

Aunque existen aplicaciones y plataformas digitales que permiten reportar incidentes, como SoSafe o bSafe, muchas de estas presentan limitaciones técnicas que afectan su efectividad. En varios casos, estas soluciones no integran de manera adecuada funcionalidades como geolocalización en tiempo real, generación automática de alertas o análisis de datos a mayor escala. Asimismo, su diseño suele ser generalizado y no siempre considera las particularidades de contextos locales, lo que reduce su utilidad en países como Chile. Estas limitaciones han sido identificadas en estudios sobre el uso de tecnologías digitales en seguridad ciudadana, donde se señala la necesidad de soluciones más integradas y adaptadas al entorno (United Nations, 2021).

3. Escasa Participación Ciudadana:

La percepción de inseguridad y la limitada disponibilidad de herramientas de colaboración comunitaria han contribuido a una menor vinculación entre los ciudadanos y las instituciones encargadas de la seguridad pública. Diversos estudios han evidenciado una disminución en los niveles de confianza hacia estas instituciones, lo que impacta directamente en la disposición de las personas a participar en procesos de denuncia o prevención del delito (Latinobarómetro, 2023).

En muchas comunidades, las personas no cuentan con los recursos necesarios para trabajar de manera conjunta en la prevención de delitos o en la generación de estrategias de respuesta ante emergencias. Diversos estudios han señalado que la falta de participación comunitaria y de herramientas de coordinación limita la efectividad de las estrategias de prevención del delito a nivel local (Programa de las Naciones Unidas para el Desarrollo, 2022). Esto no solo perpetúa el sentimiento de indefensión, sino que también reduce el impacto de las iniciativas orientadas a fortalecer la seguridad en los territorios.

4. Impacto en Zonas Vulnerables:

La problemática se agrava en zonas de alta vulnerabilidad social, donde la presencia institucional suele ser más limitada y los recursos tecnológicos son menos accesibles. Diversos estudios han señalado que estas condiciones incrementan la exposición al delito y dificultan la implementación de estrategias de prevención efectivas, afectando especialmente a grupos como mujeres, niños y adultos mayores (Comisión Económica para América Latina y el Caribe, 2021). En estos contextos, la falta de acceso a herramientas de seguridad modernas y de comunicación eficiente contribuye a aumentar la sensación de riesgo y desprotección.

La carencia de soluciones integrales y accesibles para abordar estas problemáticas ha generado la necesidad de desarrollar herramientas tecnológicas más adaptadas al contexto local y orientadas a fomentar la participación de las comunidades. En este sentido, diversos estudios han destacado el potencial de las tecnologías digitales para mejorar la prevención del delito y facilitar la comunicación entre ciudadanos e instituciones, especialmente mediante el uso de geolocalización, sensores y sistemas de monitoreo en tiempo real (United Nations Office on Drugs and Crime, 2021). En este contexto, el desarrollo de aplicaciones móviles orientadas a la seguridad representa una alternativa viable para complementar los mecanismos tradicionales de denuncia y prevención.

1.2 Contexto de la solución propuesta

La presente memoria se desarrolla en el contexto del diseño e implementación de una solución tecnológica orientada a la seguridad personal, enmarcada dentro del área de la Informática Aplicada. En particular, el trabajo se centra en el desarrollo de una aplicación móvil que busca apoyar a los usuarios en la prevención y reporte de incidentes, utilizando herramientas digitales accesibles y de uso cotidiano.

El proyecto, denominado Vision Security, surge como una iniciativa enfocada en integrar distintas tecnologías con el fin de ofrecer una solución práctica ante problemáticas relacionadas con la inseguridad. Para ello, se propone el desarrollo de una aplicación móvil multiplataforma, apoyada en servicios en la nube mediante AWS (Amazon Web Services), que permite gestionar información en tiempo real.

La solución incorpora funcionalidades como geolocalización de incidentes, generación de alertas, contacto con redes de apoyo y registro de reportes comunitarios, con el objetivo de facilitar la comunicación y mejorar la respuesta ante situaciones de riesgo. De esta

manera, el trabajo se orienta a describir el proceso de desarrollo e integración de estas tecnologías en una solución funcional.

1.3 Situación actual y actores involucrados

Actualmente, la situación de la seguridad en Chile presenta diversos desafíos relacionados tanto con la ocurrencia de delitos como con la percepción de inseguridad en la población. De acuerdo con la Encuesta Nacional Urbana de Seguridad Ciudadana (ENUSC, 2023), una proporción significativa de víctimas no realiza denuncias formales, lo que se asocia principalmente a factores como desconfianza en las instituciones o percepción de baja efectividad del proceso.

Por otra parte, distintos estudios han señalado que, si bien existen avances en el uso de tecnologías aplicadas a la seguridad, aún persisten limitaciones en la integración de herramientas digitales que permitan una comunicación eficiente entre ciudadanos y autoridades (Banco Interamericano de Desarrollo, 2020). Asimismo, la participación ciudadana en plataformas digitales de seguridad continúa siendo limitada, lo que dificulta el desarrollo de estrategias colaborativas de prevención.

En este contexto, se identifican los siguientes actores involucrados:

Ciudadanos: Principales afectados por la inseguridad, especialmente en entornos urbanos, donde la percepción de riesgo tiende a ser mayor (ENUSC, 2023).

Autoridades: Instituciones encargadas del orden y la seguridad pública, como Carabineros de Chile y la Policía de Investigaciones de Chile, responsables de la prevención, control y persecución del delito.

Redes cercanas: Familiares y amigos que cumplen un rol relevante como apoyo en situaciones de emergencia, especialmente en contextos donde se utilizan herramientas digitales de comunicación inmediata.

En términos de percepción y acceso a tecnología, se observa que una parte importante de la población considera que la inseguridad ha aumentado en el último tiempo (ENUSC, 2023). Asimismo, el acceso a dispositivos móviles es alto en el país, alcanzando una amplia penetración de teléfonos inteligentes, lo que representa una oportunidad para el desarrollo de soluciones tecnológicas orientadas a la seguridad (Statista, 2023).

1.4 Justificación del problema y consecuencias de no solucionarlo

De no implementarse una solución tecnológica eficaz, las consecuencias a mediano plazo serían:

- **Aumento en la percepción de inseguridad:** Los ciudadanos continuarían sintiéndose vulnerables en la vía pública.
- **Mayor subreporte de delitos:** La falta de herramientas accesibles desmotiva a las víctimas a denunciar.
- **Deterioro de la confianza:** Las autoridades enfrentarían mayores dificultades para responder oportunamente.

El desarrollo de esta memoria permite aprovechar la tecnología para crear una aplicación innovadora, que combina funcionalidades avanzadas y adaptadas a las necesidades específicas del contexto chileno. A diferencia de otras soluciones existentes, esta aplicación integrará geolocalización precisa, alertas en tiempo real y un sistema de contacto directo con familiares y autoridades, optimizando la respuesta a emergencias. Su diseño se centra en la colaboración comunitaria, haciendo uso de redes sociales y herramientas de análisis predictivo para prevenir incidentes. Además, incorpora una interfaz amigable y accesible, asegurando que sea utilizable por personas de todas las edades. que aborde estas problemáticas de manera eficaz y accesible.

1.5 Competencia Existente

En el mercado existen aplicaciones de seguridad que abordan parcialmente la problemática de la inseguridad ciudadana. Por ejemplo, "SafeCity" en India permite reportar incidentes relacionados con acoso y violencia de género en un mapa público, mientras que "Citizen" en Estados Unidos ofrece notificaciones sobre delitos y emergencias cercanas en tiempo real. Ambas aplicaciones han demostrado ser efectivas en sus respectivos contextos, destacándose por facilitar el acceso a información relevante y promover la acción comunitaria. Sin embargo, presentan limitaciones importantes al ser generalistas y no completamente adaptadas a la realidad chilena.

- **Adaptabilidad:** No están diseñadas para las necesidades y cultura local.
- **Funcionalidades limitadas:** Carecen de integración con redes familiares y servicios de alerta en tiempo real.

La solución propuesta en esta memoria se diferencia por ser una herramienta adaptada al contexto chileno, con funcionalidades más completas y enfocadas en la colaboración comunitaria.

Competencia Nacional: SoSafe

A nivel nacional, el principal competidor es **SoSafe**, una aplicación ampliamente utilizada en Chile que permite reportar incidentes de seguridad, emergencias médicas y otros problemas comunitarios directamente a las autoridades locales y a vecinos cercanos. Su popularidad se debe a su integración con servicios municipales y su enfoque en mejorar la comunicación entre los ciudadanos y las entidades gubernamentales.

Sin embargo, **SoSafe** también presenta limitaciones que Vision Security busca abordar:

- **Enfoque centralizado:** Aunque SoSafe ofrece comunicación directa con las autoridades, su estructura depende en gran medida de la participación municipal, lo que puede limitar su efectividad en zonas con menor cobertura de servicios.
- **Falta de personalización:** La aplicación carece de funcionalidades específicas como la integración de redes familiares, sensores para detectar situaciones de emergencia personal y mapas globales interactivos, aspectos que hacen de Vision Security una alternativa más versátil y adecuada para la seguridad tanto individual como comunitaria.

Lo que diferencia nuestro proyecto de la competencia, como SOSAFE, es nuestra continua preocupación por los usuarios. Al probar la aplicación móvil de SOSAFE, nos dimos cuenta de que su atención hacia los usuarios no era la óptima. Cuando surgían bugs y problemas de uso, el equipo de SOSAFE no proporcionaba una respuesta inmediata y satisfactoria. Además, algunos botones dentro de su app estaban desactivados, lo que perjudicaba a muchos usuarios que querían realizar reportes. Con nosotros, esos errores, aunque pequeños pero importantes, no se presentarán al utilizar nuestros servicios.

Nuestra propuesta de valor ofrece una aplicación móvil que proporciona información en tiempo real sobre áreas inseguras, alertas de seguridad y comunicación rápida con contactos de emergencia y autoridades locales. A diferencia de SOSAFE, que se enfoca más en reportes generales de cualquier suceso que el usuario prefiera reportar, nuestro enfoque está dirigido específicamente a la seguridad de nuestros clientes y usuarios. Por ende, todo lo que ofrecemos en nuestros servicios tiene un carácter prioritario de seguridad.

1.6 Objetivos

Objetivo General

Desarrollar una aplicación móvil para el reporte de incidentes de seguridad, mediante la implementación de un frontend en Flutter y un backend soportado en servicios en la nube, integrando funcionalidades de geolocalización, almacenamiento de datos y comunicación en tiempo real.

Objetivos Específicos

1. Desarrollar una aplicación móvil multiplataforma utilizando Flutter, incorporando funcionalidades de geolocalización, generación de alertas y contacto con redes de apoyo.
2. Implementar un backend basado en servicios en la nube (AWS) para la gestión, almacenamiento y disponibilidad de la información.
3. Diseñar e implementar un sistema de reporte de incidentes dentro de la aplicación, permitiendo a los usuarios registrar y visualizar eventos de seguridad.
4. Realizar pruebas funcionales de la aplicación desarrollada, verificando el correcto funcionamiento de sus principales características.

CAPÍTULO 2: MARCO CONCEPTUAL

El presente capítulo describe los fundamentos teóricos, metodológicos y tecnológicos que sustentan el desarrollo de la aplicación móvil Vision Security. Se abordan los conceptos clave relacionados con el desarrollo de aplicaciones móviles, arquitectura backend, servicios en la nube, geolocalización, sensores móviles y seguridad ciudadana, estableciendo un marco de referencia que permite comprender la solución propuesta y su relación con problemáticas reales.

Además, se incorporan referencias a trabajos y tecnologías existentes, con el fin de evitar duplicidad de enfoques y asegurar que la solución desarrollada se basa en prácticas consolidadas dentro del área.

2.1 Desarrollo de Aplicaciones Móviles

El desarrollo de aplicaciones móviles corresponde al proceso de creación de software diseñado para ejecutarse en dispositivos como smartphones y tablets. Estas aplicaciones pueden ser nativas, híbridas o multiplataforma, dependiendo de las tecnologías utilizadas.

En este proyecto se utilizó Flutter, un framework de código abierto desarrollado por Google que permite crear aplicaciones multiplataforma utilizando el lenguaje Dart [Google21]. Flutter destaca por su capacidad de generar interfaces nativas de alto rendimiento a partir de un único código base, lo que reduce tiempos de desarrollo y facilita el mantenimiento.

En el contexto de Vision Security y propio, Flutter permitió implementar una interfaz interactiva que integra mapas, formularios de reportes y funcionalidades en tiempo real, asegurando una experiencia de usuario fluida y consistente.

Enfoque	Características principales	Ventajas	Desventajas	Uso en el proyecto
Nativo (Android)	Desarrollo específico para Android (Kotlin/Java)	Alto rendimiento y acceso completo al hardware	Mayor tiempo de desarrollo	No utilizado
Híbrido	Basado en tecnologías web (HTML, CSS, JS)	Desarrollo rápido	Menor rendimiento	No utilizado
Flutter (Framework multiplataforma)	Desarrollo con un solo código base en Dart	Buen rendimiento y rapidez de desarrollo	Requiere adaptación inicial	✓ Utilizado (despliegue en Android)

Tabla 1. Enfoques de desarrollo móvil y selección tecnológica del proyecto.

2.2 Arquitectura Backend para Aplicaciones Móviles

El backend es el componente encargado de gestionar la lógica de negocio, almacenamiento de datos y comunicación con el frontend. Actúa como intermediario entre la aplicación móvil y los servicios externos, permitiendo operaciones como autenticación, registro de usuarios y almacenamiento de información [Fielding00].

En este proyecto, el backend fue implementado mediante servicios web desarrollados en PHP, los cuales se encargan de:

- Registrar reportes de incidentes.
- Gestionar solicitudes entre usuarios.
- Almacenar información de ubicación.
- Proveer datos al frontend mediante respuestas en formato JSON.

Esta arquitectura sigue un modelo cliente-servidor, donde la aplicación Flutter consume servicios mediante solicitudes HTTP, permitiendo una separación clara entre la lógica de presentación y la lógica de negocio.

Endpoint	Método	Función	Uso en la app
registrar_ubicacion.php	POST	Guarda ubicación del usuario	MapScreen
obtener_reportes.php	GET	Obtiene reportes para el mapa	MapScreen
card_reportes.php	GET	Lista reportes tipo feed	ReportScreen
obtener_solicitudes.php	POST	Obtiene solicitudes de amistad	NotificationScreen
aceptar_solicitud.php	POST	Acepta solicitud	NotificationScreen
eliminar_solicitud.php	POST	Elimina solicitud	NotificationScreen

Tabla 2. Endpoints backend utilizados en Vision Security

2.3 Servicios en la Nube (Cloud Computing)

El Cloud Computing permite el acceso a recursos tecnológicos a través de internet, evitando la necesidad de infraestructura local. Según su modelo de servicio, se clasifica en IaaS, PaaS y SaaS [Mell10].

Para el desarrollo de Vision Security, se utilizó Amazon Web Services (AWS), una de las plataformas más utilizadas a nivel mundial, integrando los siguientes servicios:

- Amazon EC2: Para alojar el backend de la aplicación.
- Amazon RDS: Para la gestión de bases de datos relacionales.
- Amazon S3: Para el almacenamiento de imágenes asociadas a reportes.

El uso de la nube permitió garantizar:

- Escalabilidad del sistema.
- Alta disponibilidad de los servicios.
- Seguridad en el almacenamiento de datos.

Esto resulta fundamental en aplicaciones de seguridad, donde la disponibilidad en tiempo real es un requisito crítico.

Servicio	Tipo	Función	Aplicación en Vision Security
EC2	IaaS	Servidor backend	Hosting de servicios PHP
RDS	PaaS	Base de datos relacional	Almacenamiento de usuarios y reportes
S3	PaaS	Almacenamiento de archivos	Imágenes de reportes

Tabla 3. Servicios AWS utilizados en el proyecto

2.4 Geolocalización y Mapas Interactivos

La geolocalización es la capacidad de determinar la ubicación de un dispositivo en tiempo real mediante tecnologías como GPS, redes móviles o Wi-Fi [Zandbergen20].

En este proyecto, la geolocalización cumple un rol central, permitiendo:

- Mostrar la ubicación actual del usuario.
- Registrar la ubicación exacta de incidentes.
- Visualizar reportes en un mapa interactivo.

Para ello, se integró la API de Mapbox, la cual permite la visualización de mapas personalizados y la gestión de datos geoespaciales. Esta herramienta facilita la representación visual de zonas de riesgo, mejorando la toma de decisiones por parte del usuario.

Además, se utilizaron librerías como Geolocator y Geocoding para obtener coordenadas y traducirlas en direcciones comprensibles.

Componente	Librería/API	Función	Uso en el código
GPS	Geolocator	Obtener ubicación actual	<code>determinePosition()</code>
Geocoding	Geocoding	Convertir coordenadas a dirección	<code>getStreetName()</code>
Mapas	Flutter Map + Mapbox	Visualización de mapa	<code>FlutterMap</code>
Marcadores	MarkerLayer	Mostrar reportes	<code>Lista markers</code>

Tabla 4. Componentes de geolocalización implementados

2.5 Sensores de Dispositivos Móviles

Los dispositivos móviles modernos incorporan múltiples sensores que permiten capturar información del entorno y del propio dispositivo, como acelerómetro, giroscopio y GPS [Kumar21].

En Vision Security, el uso de sensores está orientado principalmente a:

- Detectar movimientos bruscos mediante el giroscopio.
- Activar el modo de emergencia ante posibles situaciones de riesgo.
- Obtener la ubicación en tiempo real del usuario.

Estas funcionalidades permiten transformar el dispositivo móvil en una herramienta activa de seguridad, capaz de reaccionar ante eventos críticos sin intervención manual directa.

Sensor	Función	Aplicación en el proyecto
GPS	Ubicación en tiempo real	Reportes y mapa
Giroscopio	Detectar movimiento	Modo emergencia
Acelerómetro	Movimiento brusco	Modo emergencia/Notificación

Tabla 5. Sensores y funcionalidades asociadas

2.6 Seguridad Ciudadana y Tecnología

Los conceptos presentados en este capítulo no solo definen el contexto teórico del proyecto, sino que también se aplican directamente en la implementación de *Vision Security*.

- Flutter permitió construir la interfaz móvil.
- AWS proporcionó la infraestructura backend.
- Mapbox y GPS habilitaron la geolocalización.
- Los sensores móviles permitieron funciones de emergencia.
- La metodología ágil facilitó el desarrollo iterativo.

De esta manera, el marco conceptual establece una base sólida que justifica las decisiones tecnológicas adoptadas y demuestra que la solución desarrollada se encuentra alineada con prácticas actuales del desarrollo de software y la seguridad tecnológica.

Con el fin de evidenciar la correspondencia entre el desarrollo técnico y las funcionalidades implementadas en la aplicación *Vision Security*, se presenta la siguiente tabla que vincula componentes del código con su propósito dentro del sistema.

Componente / Archivo	Fragmento o Función	Funcionalidad implementada	Descripción
MapScreen.dart	determinePosition()	Obtención de ubicación	Permite obtener la ubicación actual del usuario mediante GPS
MapScreen.dart	getAndSendLocation()	Envío de ubicación	Captura la ubicación y la envía al servidor
MapScreen.dart	sendLocation()	Persistencia de ubicación	Envía latitud y longitud a la base de datos mediante HTTP POST

Componente / Archivo	Fragmento o Función	Funcionalidad implementada	Descripción
MapScreen.dart	FlutterMap	Visualización de mapa	Muestra el mapa interactivo con Mapbox
MapScreen.dart	MarkerLayer	Visualización de reportes	Representa los reportes mediante marcadores en el mapa
MapScreen.dart	onTap()	Creación de reportes	Permite seleccionar ubicación y generar un reporte
MapScreen.dart	getStreetName()	Geocodificación	Convierte coordenadas en dirección legible
MapScreen.dart	emergencyAction()	Menú de emergencia	Despliega opciones de acción rápida (policía, bomberos, etc.)
MapScreen.dart	sendEmergencyEmail()	Alerta de emergencia	Envía correo con ubicación del usuario
ReportScreen.dart	obtenerReportes()	Consulta de reportes	Obtiene reportes desde el backend
ReportScreen.dart	ListView.builder	Visualización tipo feed	Muestra reportes en formato similar a red

Componente / Archivo	Fragmento o Función	Funcionalidad implementada	Descripción
			social
NotificationScreen.dart	retrieveSolicitudes()	Gestión de solicitudes	Obtiene solicitudes de amistad desde el servidor
NotificationScreen.dart	aceptarSolicitud()	Aceptar solicitudes	Permite aceptar conexiones entre usuarios
NotificationScreen.dart	eliminarSolicitud()	Rechazar solicitudes	Permite eliminar solicitudes pendientes
SettingsScreen.dart	Switch (Notificaciones)	Configuración de notificaciones	Permite activar o desactivar notificaciones
SettingsScreen.dart	DropDownButton (Tema)	Cambio de tema	Permite seleccionar modo claro u oscuro
Global (main.dart)	idUsuarios	Identificación de usuario	Mantiene el ID del usuario activo en la sesión

Tabla 6. Relación código – funcionalidad en Vision Security

CAPÍTULO 3: PROPUESTA DE SOLUCION

3.1 Metodología de Desarrollo

El desarrollo de la aplicación móvil Vision Security se realizó mediante una estrategia de trabajo colaborativa, donde distintos módulos del sistema fueron desarrollados en paralelo por los integrantes del proyecto. En mi caso, el trabajo se centró en el desarrollo del frontend de la aplicación móvil utilizando Flutter, implementando las interfaces de usuario y las funcionalidades asociadas a la interacción del usuario con los distintos servicios del sistema.

Durante el desarrollo, las funcionalidades implementadas fueron integrándose progresivamente con los servicios backend disponibles, realizando pruebas funcionales cada vez que un módulo alcanzaba un estado estable. Este proceso permitió detectar oportunamente errores de integración, realizar ajustes en la interfaz de usuario y validar el correcto funcionamiento antes de incorporar nuevas funcionalidades.

La metodología seguida durante la implementación se resume en la siguiente tabla

Etapas	Objetivo	Actividades realizadas	Producto obtenido	Forma de validación
Diseño de interfaces	Diseñar la navegación de la aplicación	Construcción de pantallas principales, navegación y componentes visuales en Flutter	Interfaces funcionales de la aplicación	Pruebas de navegación entre pantallas
Integración del mapa	Incorporar la geolocalización y visualización de incidentes	Integración de Mapbox, GPS y visualización de marcadores	Mapa interactivo con ubicación del usuario y reportes	Comparación de coordenadas reales y visualización correcta en pantalla
Implementación de reportes	Permitir registrar incidentes desde la aplicación	Desarrollo del formulario de reportes, selección de ubicación e integración con el backend	Registro de reportes con ubicación e imágenes	Verificación del almacenamiento y recuperación de la información
Implementación del modo emergencia	Incorporar funciones de asistencia al	Desarrollo del botón de emergencia, envío	Funcionalidad de asistencia operativa	Pruebas funcionales verificando el

Etapas	Objetivo	Actividades realizadas	Producto obtenido	Forma de validación
	usuario	de ubicación y notificación al contacto de emergencia		envío correcto de la información
Integración de notificaciones	Gestionar solicitudes e interacción entre usuarios	Implementación de la pantalla de notificaciones y aceptación de solicitudes	Gestión de solicitudes desde la aplicación	Verificación mediante consultas al servidor y actualización de la interfaz
Configuración de la aplicación	Incorporar opciones básicas de personalización	Desarrollo del módulo de configuración y preferencias	Pantalla de configuración funcional	Pruebas de funcionamiento de cada opción disponible
Integración final	Verificar el funcionamiento conjunto de todos los módulos	Corrección de errores de integración, optimización de navegación y pruebas generales	Aplicación completamente funcional	Pruebas integrales ejecutadas por los desarrolladores y posteriormente por usuarios durante la etapa de validación

Tabla 7 .Etapas de implementación y validación de la solución propuesta.

3.2 Arquitectura de la Solución

La solución propuesta se diseñó siguiendo un modelo cliente-servidor, integrando tecnologías modernas que permiten facilitar la gestión de la información, la comunicación entre componentes y la posibilidad de escalar la solución en el futuro. Para ello, se utilizaron herramientas como Flutter para el desarrollo del cliente móvil y servicios en la

nube mediante AWS para el backend, lo que proporciona una base tecnológica adecuada para el desarrollo de aplicaciones distribuidas.

1. **Frontend de la aplicación móvil:**

- **Tecnología utilizada:** Flutter.
- **Características:** Diseño responsivo, integración con sensores del dispositivo, interacción con Mapbox para la visualización de mapas y reportes.
- **Interfaces principales:**

Vista de mapa con geolocalización en tiempo real.

Formulario para generar reportes de seguridad.

Vista de comunidades y acceso a imágenes en Firebase.

2. **Backend de la aplicación móvil:**

- **Tecnología utilizada:** PHP para la lógica del servidor, MySQL como base de datos y AWS como infraestructura en la nube.
- **Características:**

Gestión de usuarios y autenticación.

Almacenamiento y consulta de reportes de seguridad.

Integración con Mapbox para datos de ubicación.

3. **Uso de sensores del teléfono:**

- **Función principal:** Botón de emergencia que detecta movimientos bruscos utilizando el acelerómetro y giroscopio del dispositivo móvil.
- **Tecnología utilizada:** Sensores nativos de Flutter y bibliotecas complementarias.

4. **Almacenamiento en la nube:**

- **Proveedor:** Amazon Web Services (AWS).

- **Servicios utilizados:**

RDS para la base de datos relacional.

S3 para almacenamiento de imágenes y archivos relacionados.

EC2 para el hosting del backend.

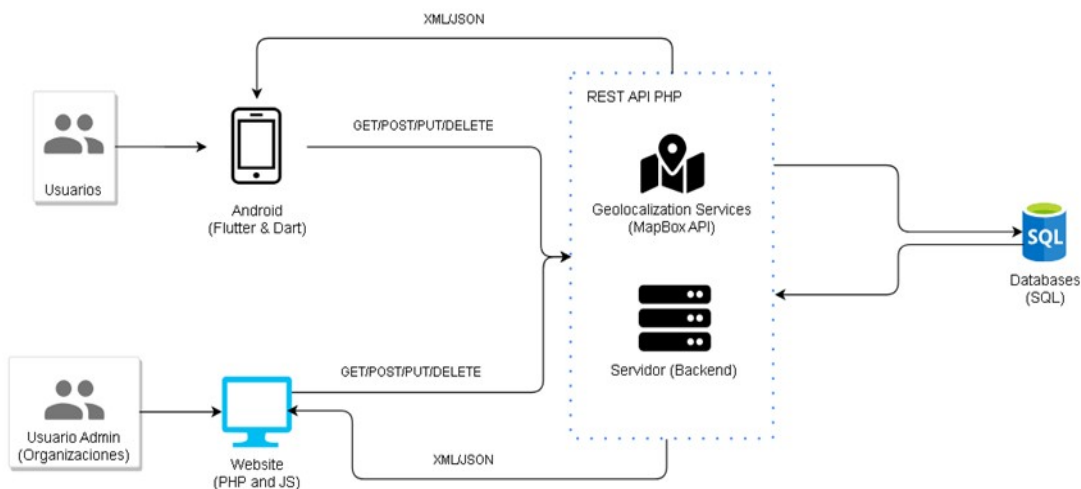


Imagen 1 .Arquitectura Inicial del sistema Vision Security.Fuente: Elaboración propia.

Para el desarrollo de la solución, se definió inicialmente una arquitectura basada en un modelo cliente-servidor, la cual permitió establecer la interacción entre los distintos componentes del sistema. En la **Imagen 1** se presenta la arquitectura inicial considerada para el proyecto.

3.3 Proceso de Generación de Reportes de Seguridad

El proceso de reportes se desarrolló con los siguientes pasos:

1. **Ingreso de datos:** El usuario introduce información relevante en un formulario en la aplicación móvil, incluyendo ubicación, imágenes y descripción del incidente.
2. **Validación y almacenamiento:** Los datos son validados localmente y enviados al backend para su almacenamiento en la base de datos alojada en AWS RDS.
3. **Visualización:** Los reportes se muestran en tiempo real en la vista de mapa, utilizando Mapbox para una experiencia visual intuitiva.

3.4 Creación y Gestión de Comunidades

La funcionalidad de comunidades permite a los usuarios organizarse y colaborar en la prevención de delitos.

- **Creación de comunidades:** Los usuarios pueden crear o unirse a comunidades basadas en su ubicación o intereses comunes.
- **Gestión de comunidades:** Los administradores pueden gestionar los miembros y reportes asociados.

3.5 Integración con Firebase

Firebase se utilizó para:

- **Almacenamiento de imágenes:** Cada reporte puede incluir imágenes relevantes, almacenadas en Firebase Storage.
- **Notificaciones push:** Los miembros de comunidades reciben alertas cuando se genera un reporte cercano.

3.6 Tabla de Funcionalidades Implementadas

Con el objetivo de resumir las principales funcionalidades implementadas en la aplicación, en la Tabla 1 se presentan los módulos desarrollados junto con las tecnologías utilizadas en cada uno de ellos. Esta tabla permite visualizar de manera integrada cómo se estructuró la solución y qué herramientas fueron empleadas en cada componente.

Como se observa, la aplicación incorpora un conjunto de funcionalidades orientadas a la seguridad del usuario. Entre ellas destaca el botón de emergencia, el cual utiliza los sensores del dispositivo móvil para detectar movimientos bruscos y generar alertas. Asimismo, la visualización de información se realiza mediante un sistema de mapas interactivos basado en Mapbox, permitiendo ubicar geográficamente los incidentes reportados.

Por otra parte, la generación de reportes constituye una de las funcionalidades centrales del sistema, ya que permite a los usuarios registrar incidentes incluyendo ubicación e imágenes, integrando el frontend desarrollado en Flutter con servicios en la nube mediante AWS. De manera complementaria, se implementó un módulo de gestión de comunidades que permite organizar a los usuarios y facilitar la interacción entre ellos.

Finalmente, el almacenamiento de la información se realiza utilizando servicios como AWS S3 y Firebase, lo que permite gestionar imágenes y datos de forma estructurada. En conjunto, estas funcionalidades reflejan el desarrollo integral de la aplicación, abarcando tanto la interfaz de usuario como los servicios backend necesarios para su funcionamiento.

Funcionalidad	Tecnología	Descripción
Botón de emergencia	Sensores del móvil	Detecta movimientos bruscos y envía alertas.
Vista de mapas	Mapbox	Muestra reportes de seguridad en tiempo real.
Generación de reportes	Flutter, AWS	Permite registrar incidentes con ubicación e imágenes.
Gestión de comunidades	Flutter, PHP, MySQL	Organización de usuarios para la prevención colectiva.
Almacenamiento en la nube	AWS S3, Firebase	Guarda imágenes y datos de manera segura.

Tabla 8. Funcionalidades.Fuente: Elaboración Propia

3.7 Desarrollo e Implementación del Sistema

En esta sección se presenta el desarrollo de la aplicación *Vision Security*, describiendo las principales funcionalidades implementadas en el frontend y su integración con los servicios backend. El sistema fue desarrollado como una aplicación móvil orientada a dispositivos Android, utilizando el framework Flutter, el cual permite construir interfaces modernas y eficientes mediante un único código base en lenguaje Dart [Google21].

A nivel de frontend, se diseñaron e implementaron diversas pantallas que permiten la interacción directa con las funcionalidades principales del sistema, tales como la visualización de mapas interactivos, la generación de reportes de incidentes, la gestión de notificaciones y la configuración de la aplicación. Estas interfaces fueron desarrolladas considerando principios de usabilidad y experiencia de usuario, con el objetivo de facilitar la interacción en contextos de uso rápido y situaciones potencialmente críticas [Nielsen93].

En cuanto a la integración con el backend, la aplicación se conecta a servicios web desarrollados en PHP, desplegados sobre infraestructura en la nube mediante Amazon Web Services (AWS). Esta arquitectura permite gestionar la información de manera centralizada, asegurando disponibilidad y escalabilidad del sistema [AWS23]. La comunicación entre el frontend y el backend se realiza a través de solicitudes HTTP utilizando el formato JSON para el intercambio de datos, lo cual constituye una práctica ampliamente utilizada en el desarrollo de aplicaciones distribuidas [Fielding00].

Además, se implementaron funcionalidades que aprovechan las capacidades nativas del dispositivo móvil, como la geolocalización en tiempo real mediante el uso del GPS, lo que permite posicionar al usuario en el mapa y asociar los reportes a coordenadas específicas [Zandbergen20]. Asimismo, se integraron servicios de geocodificación para transformar coordenadas en direcciones comprensibles, facilitando la interpretación de la información por parte del usuario.

Por otra parte, la visualización de la información geográfica se realizó mediante la integración de la plataforma Mapbox, la cual permite la implementación de mapas interactivos y la representación de datos georreferenciados en tiempo real [Mapbox23]. Esta funcionalidad resulta fundamental para el objetivo del sistema, ya que permite identificar zonas de riesgo y mejorar la toma de decisiones por parte de los usuarios.

Finalmente, se incluyen evidencias visuales del funcionamiento del sistema, basadas en las pantallas desarrolladas, las cuales permiten ilustrar de manera clara la implementación de las funcionalidades descritas y validar el comportamiento de la aplicación desde la perspectiva del usuario. Estas evidencias complementan la descripción técnica, facilitando la comprensión del sistema y su evaluación funcional.

3.7.1 Módulo de Autenticación de Usuarios

El sistema implementa un mecanismo de autenticación que permite a los usuarios registrarse e iniciar sesión mediante credenciales almacenadas en una base de datos remota.

El proceso de autenticación se realiza mediante solicitudes HTTP hacia scripts PHP alojados en el servidor, los cuales validan la información ingresada.

Entre las funcionalidades implementadas se encuentran:

- Inicio de sesión mediante nombre de usuario y contraseña.
- Registro de nuevos usuarios.
- Acceso en modo invitado con funcionalidades limitadas.

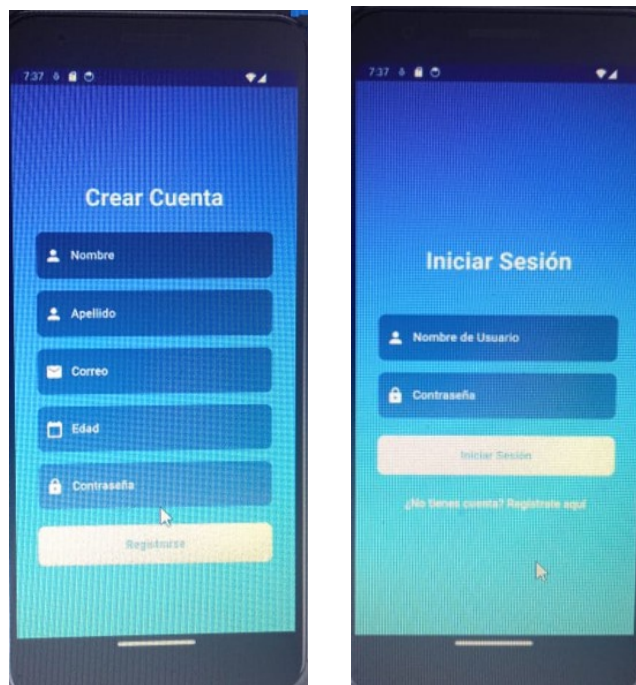


Imagen 2 .Login y creación de cuenta. Fuente: Elaboración propia.

3.7.2 Módulo de Geolocalización y Mapa Interactivo

Se desarrolló un módulo de mapa interactivo utilizando librerías de geolocalización y visualización cartográfica. Este módulo permite:

- Obtener la ubicación actual del usuario en tiempo real.
- Visualizar reportes existentes en el mapa mediante marcadores.
- Seleccionar una ubicación específica para generar un reporte.

Además, la ubicación del usuario es enviada periódicamente al servidor, permitiendo mantener información actualizada.

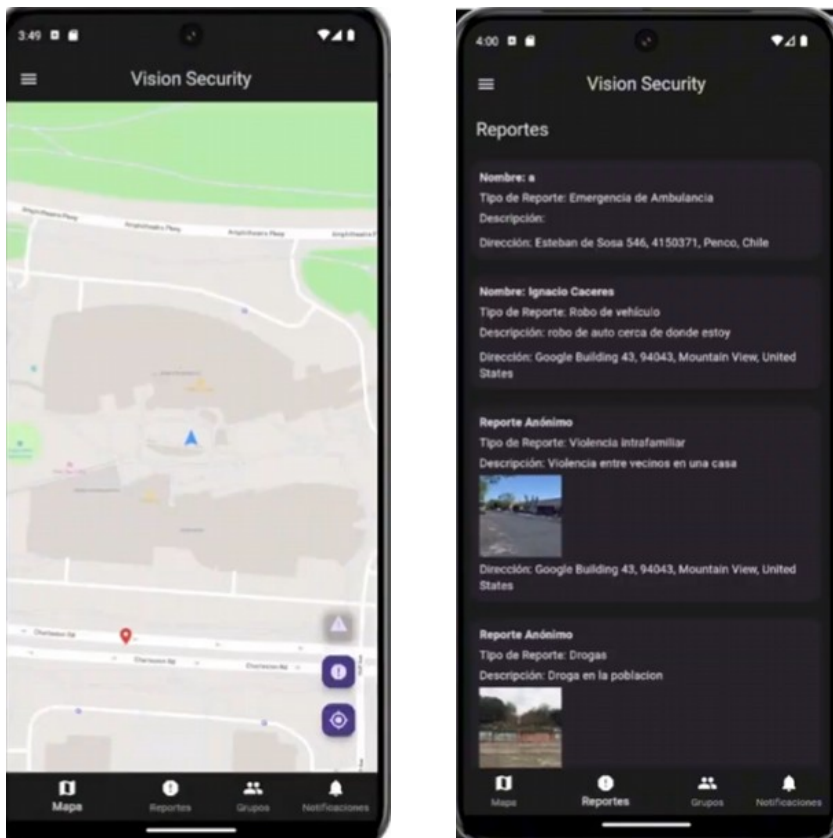


Imagen 3. Mapa y Reportes Fuente: Elaboración propia.

3.7.3 Módulo de Generación de Reportes

El sistema permite a los usuarios crear reportes de incidentes de manera estructurada.

Este proceso se implementó en múltiples etapas:

1. Selección del tipo de emergencia (seguridad, ciudad o entorno educativo).
2. Selección de la ubicación en el mapa.
3. Ingreso de descripción del incidente.
4. Opción de adjuntar evidencia fotográfica.
5. Envío del reporte al backend.

Los reportes incluyen información relevante como:

- Tipo de incidente
- Ubicación geográfica
- Descripción
- Imagen (opcional)
- Identidad del usuario o anonimato

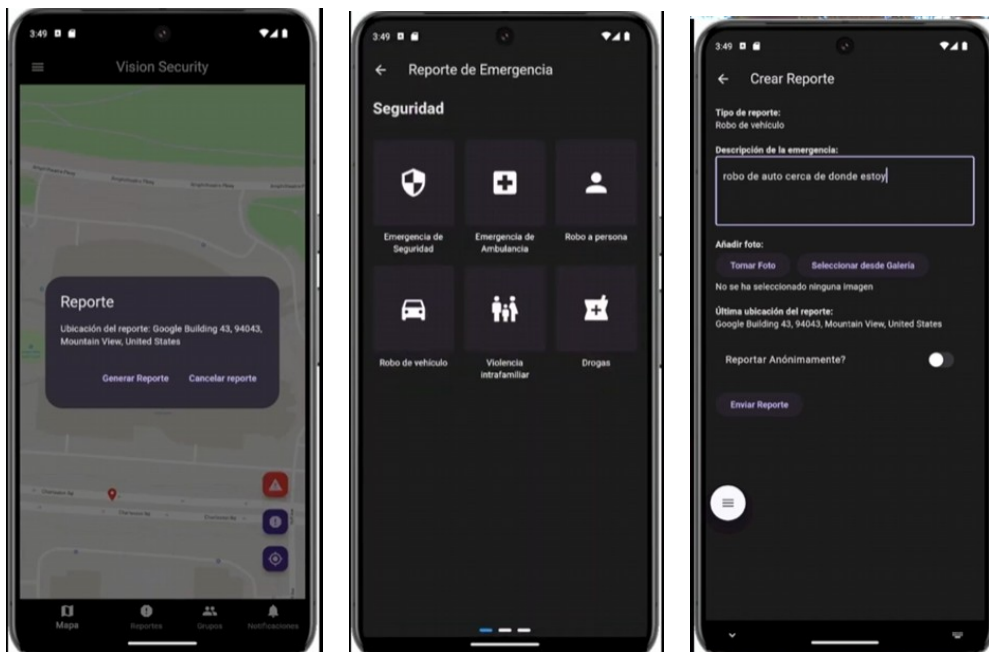


Imagen 4. Tipos y generación de reportes Fuente: Elaboración propia.

3.7.4 Módulo de Notificaciones

El sistema incorpora un módulo de notificaciones que permite gestionar solicitudes entre usuarios.

Las funcionalidades implementadas incluyen:

Visualización de solicitudes pendientes

Aceptación de solicitudes

Eliminación de solicitudes

Estas acciones se realizan mediante comunicación directa con el backend.

3.7.5 Módulo de Emergencia

Se desarrolló un módulo de emergencia accesible desde el mapa, que permite al usuario ejecutar acciones rápidas en situaciones críticas.

Entre sus funciones destacan:

- Acceso rápido a servicios de emergencia (policía, bomberos, ambulancia).
- Envío de alerta a un contacto de confianza mediante correo electrónico.

Este módulo utiliza la ubicación actual del usuario para entregar información contextual en la alerta.

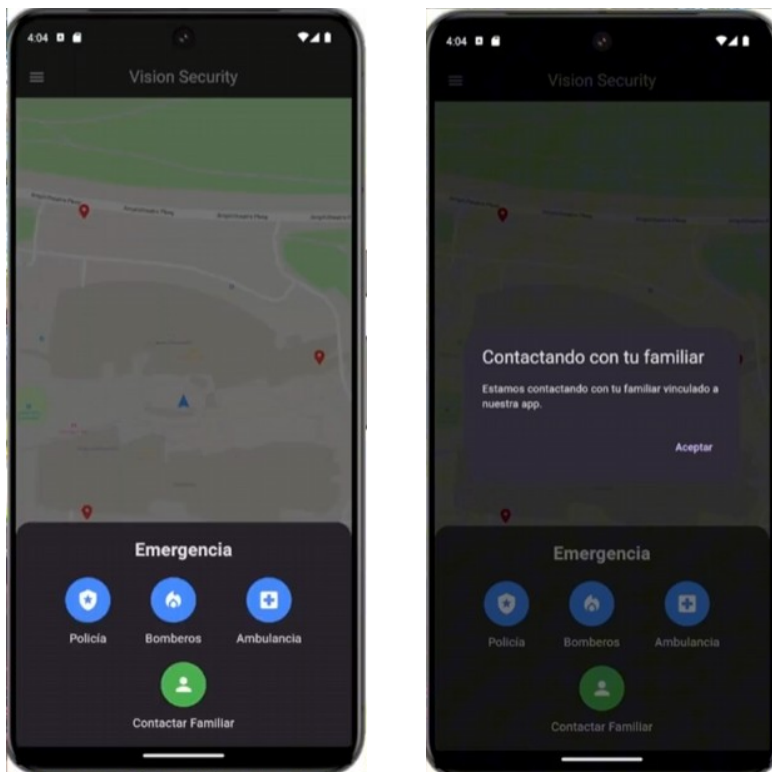


Imagen 5. Apartado de emergencias Fuente: Elaboración propia.

3.7.6 Módulo de Gestión de Cuenta

El sistema permite al usuario visualizar su información personal y actualizar datos relevantes, como el número de contacto de confianza.

Esta información es almacenada en la base de datos y utilizada en funcionalidades críticas como el envío de alertas.

3.7.7 Módulo de Configuración

Se implementó una pantalla de configuración que permite al usuario personalizar ciertos aspectos de la aplicación, tales como:

- Activación o desactivación de notificaciones
- Selección de tema visual (claro u oscuro)
- Acceso a información de la aplicación

3.7.8 Integración con Backend y Servicios Externos

El sistema se integra con un backend desarrollado en PHP y desplegado en la nube, permitiendo:

- Gestión de usuarios
- Almacenamiento de reportes
- Consulta de información en tiempo real

Adicionalmente, se integraron servicios externos como:

- APIs de geolocalización
- Servicios de almacenamiento de imágenes
- Envío de correos electrónicos para emergencias

3.8 Diagrama de Casos de Uso del Sistema

Con el objetivo de representar de manera clara las funcionalidades implementadas en la aplicación *Vision Security*, se elaboró un diagrama de casos de uso. Este modelo permite identificar las interacciones entre los distintos actores del sistema y las operaciones que estos pueden realizar dentro de la plataforma.

A diferencia de una visión centrada en el negocio, este enfoque se orienta directamente a la implementación del software, evidenciando las funcionalidades desarrolladas en el sistema.

Actores del sistema

Se identificaron los siguientes actores principales:

- **Usuario:** Persona que utiliza la aplicación y accede a todas las funcionalidades disponibles tras autenticarse.
- **Usuario invitado:** Usuario que accede sin autenticación, con acceso limitado a funcionalidades como visualización de mapas y reportes.
- **Sistema backend:** Encargado de procesar solicitudes, almacenar información y gestionar la lógica del sistema.
- **Servicios externos:** Incluyen APIs de geolocalización, servicios de almacenamiento y envío de correos electrónicos.

Descripción general del sistema

El sistema permite a los usuarios interactuar con una plataforma de seguridad ciudadana mediante funcionalidades como:

- Autenticación y gestión de cuenta
- Visualización de mapas con geolocalización en tiempo real
- Generación y consulta de reportes de incidentes
- Recepción y gestión de notificaciones
- Activación de funciones de emergencia

- Configuración de la aplicación

Estas funcionalidades se encuentran organizadas en distintos casos de uso, los cuales reflejan directamente los módulos implementados en el desarrollo de la aplicación.

Diagrama de Casos de Uso

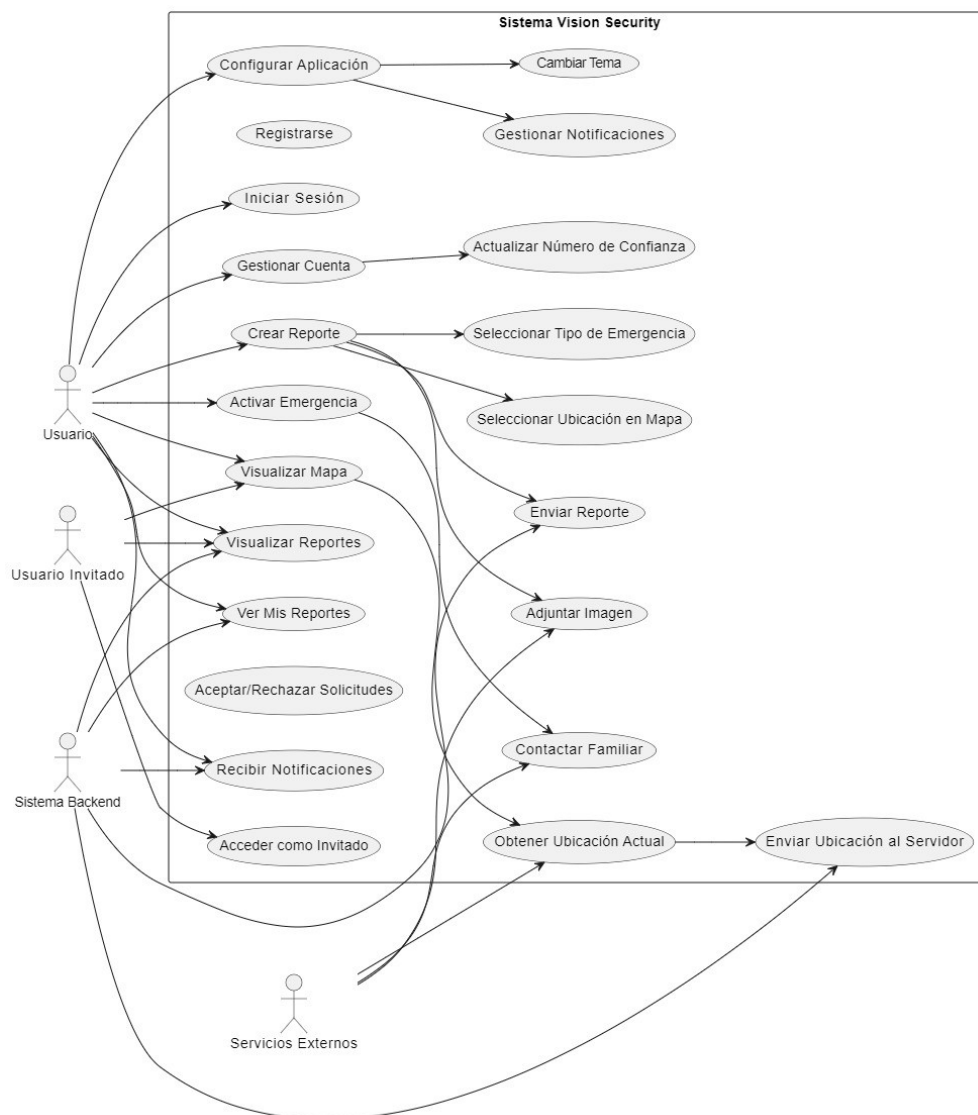


Imagen 6. Diagrama casos de uso Fuente: Elaboración propia.

Interpretación del diagrama

El diagrama muestra cómo el **usuario autenticado** interactúa con el sistema, teniendo acceso completo a funcionalidades como la creación de reportes, gestión de su cuenta, recepción de notificaciones y activación de alertas de emergencia.

Por otro lado, el **usuario invitado** posee acceso restringido, pudiendo únicamente visualizar el mapa y los reportes generados por otros usuarios.

Uno de los procesos más relevantes es la **creación de reportes**, el cual se compone de múltiples subprocesos, incluyendo la selección del tipo de incidente, la ubicación geográfica, la posibilidad de adjuntar evidencia y el envío de la información al backend.

Asimismo, el sistema interactúa con un **backend**, responsable de procesar y almacenar la información, y con **servicios externos**, utilizados para funcionalidades como geolocalización y envío de alertas.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

4.1 Metodología de Validación

Con el objetivo de evaluar la usabilidad y el funcionamiento de la aplicación móvil Vision Security, se realizó una validación con usuarios en un entorno controlado. Esta actividad permitió obtener retroalimentación sobre las principales funcionalidades implementadas en el frontend de la aplicación, verificando que los usuarios pudieran interactuar correctamente con ellas y completar las tareas propuestas.

La validación se llevó a cabo con 24 participantes, pertenecientes a dos establecimientos educacionales de la Región del Biobío. Los participantes correspondían principalmente a estudiantes, quienes representan uno de los segmentos de usuarios definidos para la aplicación. Las pruebas se realizaron de manera presencial utilizando dispositivos Android con conexión a Internet, bajo la supervisión del equipo desarrollador.

Antes de responder la encuesta, cada participante utilizó la aplicación ejecutando las principales funcionalidades implementadas, tales como la visualización del mapa interactivo, la generación de reportes, la consulta de incidentes, el acceso al modo emergencia y la navegación por las distintas pantallas de la aplicación.

Una vez finalizadas las actividades, los participantes respondieron una encuesta de percepción utilizando una escala de Likert de cinco niveles, donde 1 correspondía a "Muy en desacuerdo" y 5 a "Muy de acuerdo". Las preguntas estuvieron orientadas a evaluar aspectos relacionados con la facilidad de uso, el diseño de la interfaz, la utilidad de las funcionalidades implementadas y la experiencia general de uso.

Las características generales de la actividad de validación se presentan en la Tabla 10

Aspecto	Descripción
Número de participantes	24 usuarios
Perfil de participantes	Estudiantes de dos establecimientos educacionales de la Región del Biobío
Contexto de aplicación	Validación presencial en un entorno controlado
Dispositivos utilizados	Teléfonos móviles con sistema operativo Android
Funcionalidades evaluadas	Mapa interactivo, reportes, modo emergencia, notificaciones y navegación
Instrumento de evaluación	Encuesta de percepción
Escala utilizada	Escala de Likert de 5 niveles (1 = Muy en desacuerdo; 5 = Muy de acuerdo)

Tabla 9 . Características generales de la actividad de validación.

La información presentada en la Tabla 10 permite contextualizar el proceso de validación realizado. La participación de estudiantes pertenecientes a dos establecimientos educacionales resulta pertinente, considerando que este grupo forma parte del público objetivo definido para un segmento de la aplicación.

Con el fin de asegurar que todos los participantes interactuaran con las funcionalidades principales de la aplicación antes de responder la encuesta, se definió un conjunto de tareas prácticas que debían ser ejecutadas durante la sesión de evaluación. Estas actividades se resumen en la Tabla 10

Tarea	Objetivo de la validación
Iniciar sesión en la aplicación	Verificar el acceso al sistema y la navegación inicial.
Visualizar el mapa interactivo	Comprobar el funcionamiento de la geolocalización y la carga del mapa.
Registrar un reporte	Validar el proceso de creación de incidentes con ubicación e información descriptiva.
Consultar reportes existentes	Verificar la correcta visualización de los incidentes registrados.
Acceder al modo emergencia	Comprobar el funcionamiento de las opciones de emergencia implementadas.
Revisar las notificaciones	Validar la visualización de solicitudes y notificaciones disponibles.
Navegar por el menú de configuración	Evaluar la facilidad de acceso a las opciones de la aplicación.

Tabla 10 .Tareas ejecutadas por los participantes durante la validación.

Las tareas definidas permitieron que los participantes utilizaran los módulos más relevantes de la aplicación, proporcionando una base adecuada para evaluar posteriormente aspectos relacionados con la usabilidad, utilidad y experiencia general de uso.

4.2 Encuesta de Validación

Objetivo

Recolectar opiniones de los usuarios sobre la experiencia de uso de Vision Security en aspectos como:

- Facilidad de uso.
- Eficiencia del sistema de reportes.
- Accesibilidad a las funcionalidades principales.
- Nivel de satisfacción general con la aplicación.

Preguntas de la Encuesta

1. **En una escala del 1 al 5, ¿cómo calificarías la facilidad de uso de la app?**
2. **¿Te fue fácil el uso del sistema de reportes de la app?**
3. **¿Pudiste encontrar todas las funcionalidades necesarias en la app?**
4. **En una escala del 1 al 5, ¿qué tan satisfecho estás con el apartado de comunidades en la app?**
5. **¿Recomendarías Vision Security a otros clientes y usuarios?**

4.3 Resultados

A continuación, se presentan los resultados obtenidos de las respuestas de los usuarios, acompañados de gráficas y análisis correspondientes.

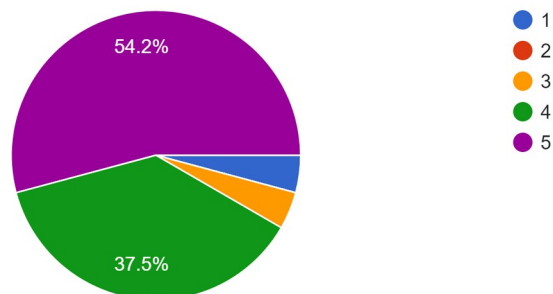
4.3.1 Facilidad de uso

- **Pregunta:** En una escala del 1 al 5, ¿cómo calificarías la facilidad de uso de la app?

- **Gráfica/Imagen:**

En una escala del 1 al 5, ¿cómo calificarías la facilidad de uso de la app? (1 = Muy difícil de usar, 5 = Muy fácil de usar)

24 respuestas

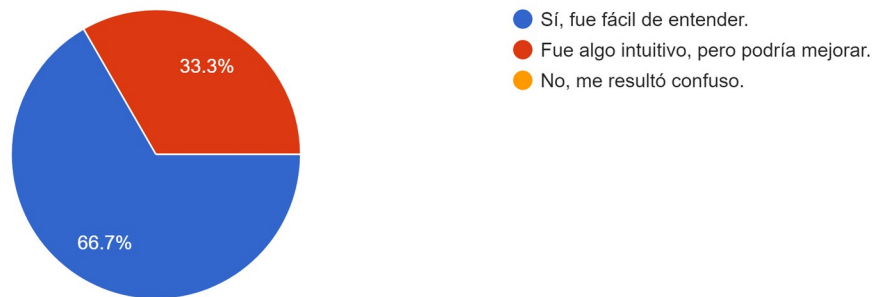


- **Análisis:** ("El 91,7% de los usuarios calificaron la app con 4 o 5, destacando la interfaz intuitiva y amigable.")

4.3.2 Uso del sistema de reportes

- **Pregunta:** ¿Te fue fácil el uso del sistema de reportes de la app?
- **Gráfica/Imagen:**

¿Te fue facil el uso del sistema de reportes de la app?
24 respuestas



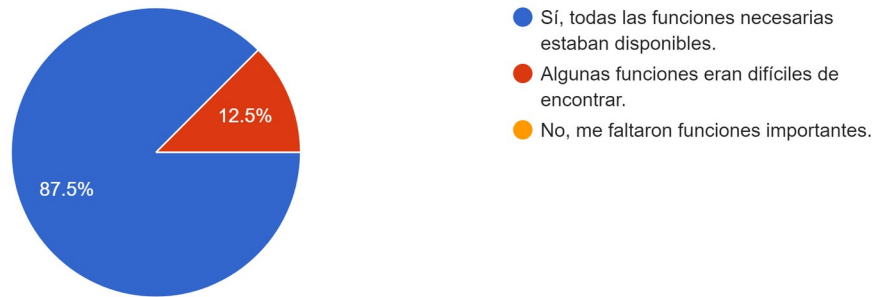
- **Análisis:** ("Un 66.7% de los encuestados indicó que el sistema de reportes es sencillo de utilizar, aunque algunos sugirieron mejoras en la claridad de las instrucciones.")

4.3.3 Funcionalidades necesarias

- **Pregunta:** ¿Pudiste encontrar todas las funcionalidades necesarias en la app?
- **Gráfica/Imagen:**

¿Pudiste encontrar todas las funcionalidades necesarias para en la app?

24 respuestas



- **Análisis:** ("La mayoría de los usuarios encontró las funcionalidades necesarias, pero un pequeño porcentaje solicitó incluir una guía rápida de uso.")

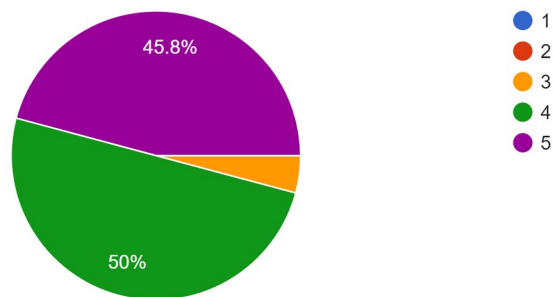
4.3.4 Satisfacción con comunidades

- **Pregunta:** En una escala del 1 al 5, ¿qué tan satisfecho estás con el apartado de comunidades en la app?

- **Gráfica/Imagen:**

En una escala del 1 al 5, ¿qué tan satisfecho estás con el apartado de comunidades en la app? (1 = Muy insatisfecho, 5 = Muy satisfecho)

24 respuestas



- **Análisis:** ("Los usuarios valoraron positivamente este apartado, escogiendo valores en su mayoría 4 y 5.")

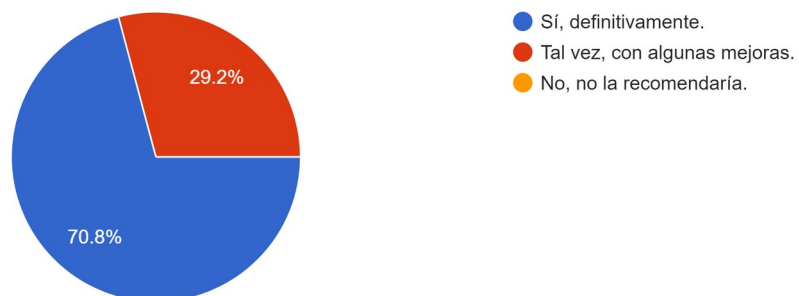
4.3.5 Recomendación de la app

- **Pregunta:** ¿Recomendarías Vision Security a otros clientes y usuarios?

- **Gráfica/Imagen:**

¿Recomendarías VisionSecurity a otros clientes y usuarios?

24 respuestas



- **Análisis:** ("El 70,8% de los usuarios afirmó que recomendaría la app, destacando su utilidad y practicidad.")

4.4 Validación funcional de los módulos principales

Además de la evaluación realizada por los usuarios, se efectuó una validación funcional de los principales módulos implementados en la aplicación móvil. Esta validación tuvo como finalidad comprobar que cada funcionalidad desarrollada operara correctamente de acuerdo con los requerimientos definidos durante el desarrollo del proyecto.

La Tabla 11 resume los módulos evaluados, el escenario de prueba aplicado, el resultado obtenido y la validación correspondiente.

Módulo	Escenario de prueba	Resultado esperado	Resultado obtenido
Mapa interactivo (MapScreen)	Ejecución del método <code>getAndSendLocation()</code> al iniciar la aplicación.	Obtener la ubicación actual mediante GPS y enviarla al servidor utilizando el servicio <code>registrar_ubicacion.php</code> .	La ubicación fue obtenida correctamente mediante Geolocator y enviada al servidor utilizando una petición HTTP POST.
Visualización de reportes (MapScreen)	Ejecución del método <code>obtenerReportes()</code> .	Recuperar los reportes almacenados en el servidor y mostrarlos como marcadores sobre el mapa.	Los datos fueron recibidos en formato JSON, convertidos a coordenadas geográficas y representados correctamente mediante <code>MarkerLayer</code> de Flutter Map.
Generación de reportes (MapScreen)	Seleccionar una ubicación sobre el mapa y confirmar la opción Generar Reporte .	Registrar la ubicación seleccionada y abrir el formulario para ingresar la información del incidente.	La aplicación almacenó las coordenadas seleccionadas, obtuvo la dirección mediante geocodificación y abrió correctamente la pantalla <code>Form_ReporteScreen</code> .
Modo de emergencia (MapScreen)	Seleccionar la opción Contactar Familiar desde el menú de emergencia.	Obtener la dirección actual del usuario y enviar un correo electrónico al contacto	Se ejecutó correctamente el método <code>sendEmergencyEmail</code>

Módulo	Escenario de prueba	Resultado esperado	Resultado obtenido
		registrado.	(), enviando el mensaje con la dirección obtenida mediante geocodificación.
Notificaciones (NotificationScreen)	Ejecución del método <code>retrieveSolicitudes()</code> .	Recuperar desde el servidor las solicitudes pendientes del usuario autenticado.	Las solicitudes fueron obtenidas correctamente mediante una petición HTTP y mostradas dinámicamente en pantalla.
Aceptación de solicitudes (NotificationScreen)	Seleccionar el botón Aceptar sobre una solicitud pendiente.	Actualizar el estado de la solicitud en el servidor y eliminarla de la lista visible.	El servicio <code>aceptar_solicitud.php</code> respondió exitosamente y la solicitud fue eliminada inmediatamente de la interfaz.
Listado de reportes (ReportScreen)	Ejecución del método <code>obtenerReportes()</code> .	Obtener todos los reportes registrados y mostrarlos mediante tarjetas informativas.	Se recibieron correctamente los datos desde <code>card_reportes.php</code> , mostrando nombre, tipo de incidente, descripción, dirección e imagen cuando correspondía.
Configuración (SettingsScreen)	Acceder al módulo de configuración desde el menú principal.	Visualizar correctamente las opciones de configuración disponibles para el usuario.	La pantalla mostró las opciones de notificaciones, tema de la aplicación, política de privacidad e información del sistema sin presentar errores de navegación.

Tabla 11 . Resultados de la validación funcional de los módulos implementados.

Interpretación de la Tabla 11

Los escenarios de prueba presentados permitieron validar el funcionamiento de los principales módulos desarrollados durante la implementación de la aplicación móvil Vision Security. En todos los casos evaluados se obtuvo el comportamiento esperado, verificando la correcta integración entre la interfaz desarrollada en Flutter, los servicios backend implementados mediante peticiones HTTP y las funcionalidades propias del dispositivo móvil, como la geolocalización y el envío de correos electrónicos.

La validación evidenció que los módulos correspondientes al mapa interactivo, generación y visualización de reportes, notificaciones, modo de emergencia y configuración operan de manera consistente dentro del entorno de pruebas utilizado. Estos resultados permiten concluir que las funcionalidades implementadas cumplen los objetivos definidos para la aplicación móvil y constituyen una base funcional para futuras mejoras e incorporaciones de nuevas características.

CAPÍTULO 5: CONCLUSIONES

5.1 Cumplimiento de los objetivos

El objetivo principal de esta memoria fue desarrollar el frontend de la aplicación móvil Vision Security, implementando una interfaz que permitiera a los usuarios reportar incidentes, visualizar información georreferenciada e interactuar con los servicios proporcionados por el backend. De acuerdo con los resultados obtenidos durante el desarrollo y la validación funcional, este objetivo fue cumplido satisfactoriamente.

En relación con el primer objetivo específico, se diseñó e implementó una aplicación móvil desarrollada en Flutter, incorporando una interfaz intuitiva que facilita el acceso a las principales funcionalidades del sistema. La aplicación permite registrar usuarios, visualizar reportes, acceder a comunidades, gestionar notificaciones y utilizar herramientas de emergencia desde un entorno gráfico unificado.

Respecto al segundo objetivo específico, se integraron correctamente servicios externos necesarios para el funcionamiento de la aplicación. La comunicación mediante servicios HTTP permitió el intercambio de información con el backend desarrollado por el otro integrante del proyecto, mientras que la utilización de AWS para el almacenamiento y la base de datos posibilitó la persistencia de la información requerida por la aplicación. Asimismo, la integración de Mapbox permitió representar geográficamente los reportes e interactuar con el mapa de forma dinámica.

En cuanto al tercer objetivo específico, se implementaron funcionalidades orientadas a mejorar la seguridad del usuario, destacando la geolocalización en tiempo real, la generación de reportes sobre el mapa, el modo de emergencia y el contacto con familiares mediante correo electrónico. Estas funcionalidades fueron verificadas durante las pruebas funcionales y demostraron operar de acuerdo con los requerimientos definidos para el proyecto.

Finalmente, la validación realizada con usuarios permitió comprobar que las principales funcionalidades fueron comprendidas y utilizadas correctamente por los participantes, obteniendo resultados favorables respecto a la facilidad de uso, utilidad y funcionamiento general de la aplicación.

5.2 Validación de la solución desarrollada

La solución implementada fue validada mediante dos mecanismos complementarios. En primer lugar, se realizaron pruebas funcionales sobre los principales módulos desarrollados durante esta memoria, verificando el correcto funcionamiento de la visualización del mapa, la obtención de reportes desde el servidor, el registro de la ubicación del usuario, el sistema de notificaciones y el modo de emergencia.

En segundo lugar, se efectuó una validación con usuarios mediante una encuesta aplicada a 24 participantes pertenecientes a establecimientos educacionales de la Región del Biobío, quienes utilizaron la aplicación en un entorno controlado. Los resultados obtenidos evidenciaron una evaluación favorable respecto a la usabilidad y funcionamiento de la aplicación, permitiendo concluir que las funcionalidades implementadas cumplen adecuadamente el propósito para el cual fueron desarrolladas.

En consecuencia, la evidencia obtenida durante las pruebas técnicas y la evaluación con usuarios permite afirmar que la solución propuesta satisface los objetivos establecidos para esta memoria.

5.3 Limitaciones identificadas

Si bien los objetivos definidos fueron alcanzados, durante el desarrollo y la validación se identificaron algunas limitaciones que deben ser consideradas.

La validación se realizó en un entorno controlado y con un número acotado de participantes, por lo que los resultados obtenidos corresponden al comportamiento observado bajo dichas condiciones y no representan necesariamente el desempeño de la aplicación en un escenario de uso masivo.

Asimismo, algunas funcionalidades dependen de servicios externos, como la disponibilidad de conexión a Internet, el funcionamiento de los servicios web y la precisión proporcionada por el sistema GPS del dispositivo móvil, factores que pueden afectar el comportamiento de determinadas funciones sin depender directamente del desarrollo realizado.

Finalmente, debido al alcance definido para esta memoria, no se incorporaron mecanismos avanzados de validación automática de reportes, análisis predictivo o integración con organismos públicos de seguridad.

5.4 Trabajo futuro

A partir de la experiencia obtenida durante el desarrollo de Vision Security, se identifican diversas oportunidades de mejora que podrían incorporarse en futuras versiones de la aplicación.

Entre ellas destacan la implementación de notificaciones push mediante Firebase Cloud Messaging, la incorporación de un sistema de autenticación más robusto utilizando servicios especializados, el desarrollo de mecanismos automáticos para validar reportes mediante inteligencia artificial y la integración con plataformas institucionales que permitan compartir información con organismos de seguridad.

Asimismo, sería recomendable realizar nuevas evaluaciones con una muestra de usuarios más amplia y en distintos contextos de uso, con el objetivo de obtener información adicional sobre el comportamiento de la aplicación en escenarios reales.

Estas mejoras permitirían fortalecer las capacidades de la solución sin modificar la arquitectura general implementada durante este proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- Flutter. (s.f.). Flutter: Build apps for any screen. Recuperado de <https://flutter.dev/>
- Amazon Web Services (AWS). (s.f.). Documentación oficial de Amazon Web Services. Recuperado de <https://aws.amazon.com/documentation/>
- Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing*. Gaithersburg, MD, EE. UU.: National Institute of Standards and Technology. Recuperado de <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Kumar, P., & Lee, J. (2021). Sensors in mobile devices and their applications: A comprehensive review. *Journal of Mobile Technology Research*, 9(2), 45-60. Recuperado de <https://www.researchgate.net/>
- Zandbergen, P. A. (2020). *Geospatial Analysis and Mapping Technologies for Urban Safety*. Londres, Reino Unido: Springer.
- Fundación Paz Ciudadana. (2023). Uso de tecnología para la prevención de delitos en Chile. Recuperado de <https://www.pazciudadana.cl/>
- Google. (s.f.). Mapbox Documentation. Recuperado de <https://docs.mapbox.com/>
- Gutiérrez, J. A. (2019). La seguridad ciudadana en la era digital. *Revista de Innovación Tecnológica y Sociedad*, 14(3), 12-20.
- Subsecretaría de Prevención del Delito. (2023). *Informe anual de delitos de mayor connotación social*. Gobierno de Chile.
- Banco Interamericano de Desarrollo. (2020). *Tecnología para la seguridad ciudadana: soluciones y desafíos en América Latina y el Caribe*. BID.
- Oficina de las Naciones Unidas contra la Droga y el Delito. (2019). *Manual sobre encuestas de victimización*. UNODC.
- United Nations. (2021). *Handbook on the use of digital technologies for crime prevention and criminal justice*. United Nations.
- Programa de las Naciones Unidas para el Desarrollo. (2022). *Informe sobre desarrollo humano y seguridad ciudadana en América Latina*. PNUD.

- Comisión Económica para América Latina y el Caribe. (2021). *Panorama Social de América Latina 2021*. CEPAL.
- Amazon Web Services. (2023). *Overview of Amazon Web Services*.
<https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html>
- Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.
- [Fielding00] Fielding, R. Architectural Styles and the Design of Network-based Software Architectures. University of California, 2000.
- [SUBTEL22] Subsecretaría de Telecomunicaciones de Chile. Uso de tecnologías digitales en seguridad ciudadana. 2022.
- [Nielsen93] Nielsen, J. Usability Engineering. 1993.