

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



**“BACKEND SISTEMA HAWKEYE: SEGURIDAD
PERSONAL EN TIEMPO REAL”**

MOISE FELIPE CONTRERAS RUBILAR

MIGUEL EDUARDO BELMAR CARRILLO

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

Profesor Guía: Cristian Antonio Lara Valenzuela

Marzo - 2024

DEDICATORIA

Este trabajo está dedicado a todas las personas que hacen posible que Hawkeye sea una realidad. A los incansables miembros del equipo que han dedicado su tiempo y esfuerzo para desarrollar esta innovadora solución, a los usuarios que confían en nuestra tecnología para cuidar de sus seres queridos y a todos aquellos que, de una forma u otra, han contribuido a hacer de Hawkeye una herramienta valiosa para nuestra sociedad.

Dedico este logro a mi familia, amigos y seres queridos, quienes han brindado su apoyo constante y comprensión en cada etapa de este emocionante viaje. También a mis profesores, mentores y a la Universidad, por brindarme los conocimientos y la inspiración necesarios para llevar a cabo este proyecto.

Hawkeye es el resultado de un esfuerzo colectivo, y esta memoria de título es un tributo a todos los que han formado parte de esta increíble aventura. Vuestra dedicación y pasión han hecho posible que Hawkeye marque la diferencia en nuestras vidas.

¡Gracias por creer en este sueño y hacerlo realidad!

AGRADECIMIENTOS

Lleno de felicidad, de amor y esperanza, dedico este proyecto a cada uno de mis seres queridos, quienes han sido mis pilares para seguir adelante. Es para mí una gran alegría poder dedicarles a ellos, que con mucho esfuerzo, esmero y trabajo me lo he ganado. A mis padres Juan Belmar y María Carrillo, porque gracias a ellos soy yo y he llegado aquí por ellos. A mi hermana Karolayn Olivero que gracias a ella logre salir adelante con mis estudios en la media y lograr entrar a esta grandiosa carrera universitaria, también agradecer sus hijos, mi sobrina y sobrino debido que ellos son la motivación de mi vida y mi orgullo de ser lo que seré. Y sin dejar atrás a toda mi familia por confiar en mí, a mis abuelitos, tíos y primos, gracias por ser parte de mi vida y por permitirme ser parte de su orgullo. Finalmente, haré una dedicatoria con todo mi agradecimiento a los profesores de la Universidad para entregarnos los conocimientos para desarrollarnos como profesionales, agradecer los consejos y el apoyo que nos han dado en estos años en la universidad.

Miguel Eduardo Belmar Carrillo.

Con gran alegría y emoción, dedico este trabajo a mi familia, quienes han sido mis pilares para seguir adelante en esta disciplina que tanto amo. Es para mí un honor poder expresarles mi gratitud por su amor, apoyo y aliento incondicional.

A mis padres, Carlos Magdiel Contreras Gutiérrez y Jacqueline del Carmen Rubilar Villarroel, les agradezco por su sacrificio y esfuerzo para que yo pudiera tener una educación superior. Gracias a ustedes, soy la persona que soy hoy.

A mi pareja, Yulisa Macarena Mella Ibarra, le agradezco por el compromiso, el amor, el apoyo incondicional, sus palabras de aliento, su apoyo en todo momento de la carrea y por ser un pilar fundamental durante estos 4 años que me ha inspirado a ser una mejor persona y ser el mejor en lo que hago. Yulisa, Eres una persona muy importante para mí

A mis hermanos, Carlos, Josué, Daniela y Magdiel, les agradezco por su apoyo incondicional y preocupación por mí. Gracias a ustedes, sé que siempre puedo contar con ustedes.

A mis amigos que he conocido en la universidad, les agradezco por su amistad y compañía. Gracias a ustedes, la universidad ha sido una experiencia mucho más enriquecedora.

Finalmente, quiero agradecer a mis profesores y tutores, quienes me han brindado sus conocimientos y experiencias. Gracias a ustedes, pude desarrollar las competencias necesarias para ser un profesional exitoso.

Moisés Felipe Contreras Rubilar

RESUMEN

Resumen—

Este proyecto se enfoca en la implementación y optimización de las respuestas de API REST en el sistema HawkEye, una solución de seguridad personal altamente personalizable y efectiva. El objetivo principal de esta memoria es proporcionar una visión detallada de cómo se logró esta integración para mejorar la eficiencia y eficacia del monitoreo de seguridad personal.

Para lograrlo, se describen en detalle tanto el diseño de la API REST como su interacción con la Base de Datos MongoDB subyacente. Se destaca la importancia de esta arquitectura para el funcionamiento de *HawkEye*, ya que permite la comunicación fluida entre la aplicación móvil, el sitio web y el dispositivo de rastreo GPS (Pulsera GPS).

Los resultados principales de este proyecto incluyen un sistema robusto y escalable que garantiza una respuesta rápida a las solicitudes de los usuarios, así como un almacenamiento y recuperación eficientes de datos de seguridad personal. Además, se explora la posible conexión directa de la Pulsera GPS con la base de datos, evaluando sus implicaciones en términos de seguridad y rendimiento.

La relevancia de estos resultados radica en la mejora significativa de la experiencia del usuario y la capacidad de respuesta en situaciones de emergencia. La arquitectura implementada fortalece la posición de *HawkEye* como una solución líder en seguridad personal, brindando a los usuarios la tranquilidad y la confianza necesarias para enfrentar los desafíos de seguridad modernos.

Palabras Clave—API REST; Integración; Base de Datos; Seguridad Personal; *HawkEye*

ABSTRACT

Abstract—

This project focuses on the implementation and optimization of REST API responses in the HawkEye system, a highly customizable and effective personal security solution. The main objective of this report is to provide a detailed overview of how this integration was achieved to improve the efficiency and effectiveness of personal security monitoring.

To achieve this, both the design of the REST API and its interaction with the underlying MongoDB Database are described in detail. The importance of this architecture for the operation of HawkEye is highlighted, as it allows smooth communication between the mobile application, the website, and the GPS tracking device (GPS Bracelet).

The main results of this project include a robust and scalable system that guarantees a quick response to user requests, as well as efficient storage and retrieval of personal security data. Additionally, the possible direct connection of the GPS Bracelet to the database is explored, evaluating its implications in terms of security and performance.

The relevance of these results lies in the significant improvement of the user experience and the responsiveness in emergency situations. The implemented architecture strengthens the position of HawkEye as a leading personal security solution, providing users with the peace of mind and confidence they need to face modern security challenges.

Keywords— REST API; Integration; Database; Personal security; HawkEye.

GLOSARIO

API (*Application Programming Interface*): Interfaz de programación de aplicaciones

CRUD (*Create, Read, Update, Delete*): Operaciones básicas de base de datos

GPS (*Global Positioning System*): Sistema de posicionamiento global que utiliza satélites para determinar la ubicación de un objeto en la Tierra.

HTTP (*Hypertext Transfer Protocol*): Protocolo de transferencia de hipertexto. Es el protocolo que permite a los navegadores web solicitar y recibir datos de los servidores web. Es un protocolo simple y eficiente que es la base de la World Wide Web.

HTTPS (*Hypertext Transfer Protocol Secure*): Es una versión segura de HTTP que cifra todo el tráfico entre el cliente y el servidor. Esto evita que los fisgones intercepten y lean el tráfico. HTTPS se utiliza para todos los datos sensibles, como contraseñas y números de tarjetas de crédito.

IDLE (*Integrated Development Environment*): Entorno de desarrollo integrado que proporciona herramientas y utilidades para facilitar el desarrollo de software.

JSON (*JavaScript Object Notation*): Formato de intercambio de datos

MVC (*Model-View-Controller*): Patrón de diseño de software que separa el contenido de una aplicación (el modelo) de su presentación (la vista) y su lógica de control (el controlador).

ORM (*Object-Relational Mapping*): Mapeo objeto-relacional

REST (*Representational State Transfer*): Es un conjunto de principios para diseñar y desarrollar interfaces de programación de aplicaciones (API). REST se basa en el uso de recursos, que son entidades que se pueden identificar y acceder a través de una URL.

SQL (*Structured Query Language*): Lenguaje de consulta estructurado que se utiliza para acceder y manipular datos en bases de datos relacionales.

UML (*Unified Modeling Language*): Es un lenguaje de modelado visual para el análisis, el diseño y la documentación de sistemas de software.

URI (*Uniform Resource Identifier*): Es un identificador de recurso uniforme. Una URI es una cadena de caracteres que se puede utilizar para identificar un recurso en Internet.

URL (*Uniform Resource Locator*): Es una dirección web que identifica un recurso en Internet.

INDICE DE CONTENIDOS

Contenido

RESUMEN	4
ABSTRACT	5
INDICE DE FIGURAS	15
INDICE DE TABLAS	17
INTRODUCCIÓN	18
1. CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	19
1.1. ORGANIZACIÓN	19
1.1.1. NOMBRE DE LA ORGANIZACIÓN	19
1.1.2. INTEGRANTES	19
1.1.3. LOGO DE LA ORGANIZACIÓN	19
1.2. PROBLEMÁTICA	19
1.2.1. ACTORES INVOLUCRADOS	20
1.2.2. Segmentación Geográfica	21
1.2.3. Segmentación Demográfica	21
1.2.4. Segmentación por Comportamiento	21
1.3. COMPETENCIA ACTUAL	22
1.4. MOTIVACIÓN	22
1.5. OBJETIVOS	23
2. CAPÍTULO 2: MARCO CONCEPTUAL	24
2.1. CONCEPTOS TÉCNICOS	24
2.1.1. API REST	24
2.1.2. Base de datos	24
2.1.3. Bluetooth	24

HawkEye Backend: Seguridad Personal en Tiempo Real

2.1.4.	Casos de uso	24
2.1.5.	Cobertura celular	24
2.1.6.	Cobertura GPS	24
2.1.7.	Desarrollo ágil.....	24
2.1.8.	Dispositivo GPS	25
2.1.9.	Escalamiento vertical	25
2.1.10.	Escalamiento horizontal.....	25
2.1.11.	Geolocalización	25
2.1.12.	Hardware de servidor	25
2.1.13.	Historias de usuario	25
2.1.14.	Hosting.....	26
2.1.15.	HTTP.....	26
2.1.16.	HTTP method.....	26
2.1.17.	HTTPS.....	26
2.1.18.	IP.....	26
2.1.19.	Java	26
2.1.20.	Lenguaje de programación.....	26
2.1.21.	Modelo de datos	26
2.1.22.	Modelo de servicio	27
2.1.23.	MVC (Modelo Vista Controlador)	27
2.1.24.	Navegación GPS.....	27
2.1.25.	NoSQL	27
2.1.26.	Operador celular	27
2.1.27.	ORM (Mapeo de Objetos a Relacionales)	27

2.1.28.	Patrones de diseño	27
2.1.29.	Precisión GPS.....	28
2.1.30.	Puerto	28
2.1.31.	Receptor GPS.....	28
2.1.32.	Rendimiento iterativo.....	28
2.1.33.	REST	28
2.1.34.	Scrum	28
2.1.35.	Servicios de hosting	28
2.1.36.	TLS	29
2.1.37.	UML	29
2.1.38.	URI	29
2.1.39.	URL	29
2.1.40.	Wearable.....	29
2.2.	METODOLOGÍAS	30
2.2.1.	METODOLOGIA SCRUM.....	30
2.3.	HERRAMIENTAS.....	30
2.4.	MODELOS	37
2.4.1.	MODELO DE DATOS	37
2.4.1.1.	Descripción de Documentos/ Tablas	38
2.4.1.1.1.	User.	38
2.4.1.1.2.	PersonSafe.	39
2.4.1.1.3.	Invitation.	39
2.4.1.1.4.	Wristband.	40
2.4.1.1.5.	SafeZones.....	40
2.4.1.1.6.	Contract.	41
2.4.1.1.7.	Plan.	41

2.4.1.2.	Conclusión Modelo de Datos	41
2.4.2.	MODELO DE ARQUITECTURA.....	43
2.4.2.1.	Explicación Arquitectura de Servicio.	43
2.4.2.1.1.	Capa de hardware	43
2.4.2.1.2.	Capa de software.....	44
2.4.2.1.3.	Zona amarilla: Dispositivo GPS	44
2.4.2.1.4.	Zona morada o rosada: Usuario	44
2.4.2.1.5.	Zona verde: Servidor	44
2.4.2.1.6.	Zona azul o celeste: Base de datos	45
2.4.2.1.7.	Conclusión	45
2.4.3.	MODELO DE RELACIÓN DE OBJETOS (ORM) EN HAWKEYE.....	46
2.4.3.1.	Ventajas de ORM	46
2.4.3.2.	Elección de ORM en HawkEye.....	46
2.4.3.3.	Modelo Monolítico y Limitaciones	47
3.	CAPÍTULO 3: PROPUESTA DE SOLUCION.....	48
3.1.	TABLAS DE HISTORIAS DE USUARIO.....	49
3.2.	CASOS DE USO EXTENDIDOS.....	62
3.2.1.	Perfil - Individuo.....	62
3.2.1.1.	Crear perfil.....	62
3.2.1.2.	Editar perfil	63
3.2.1.3.	Eliminar perfil	63
3.2.2.	Mapa.....	64
3.2.2.1.	Ver Mapa	64
3.2.2.2.	Designar Vallado Virtual	64
3.2.2.3.	Ver Historial de Recorrido.....	65
3.2.3.	Pulsera.....	66
3.2.3.1.	Registrar pulsera	66
3.2.4.	Alertas	66

3.2.4.1.	<i>Ver Historial de Alertas</i>	66
3.2.4.2.	<i>Ejecutar una Alerta</i>	67
3.2.4.3.	<i>Recibir Notificación</i>	67
3.2.4.4.	<i>Simular Alerta</i>	68
3.3.	INFRAESTRUCTURA DE SERVICIO HAWKEYE	69
3.3.1.	¿Qué es la infraestructura?	69
3.3.2.	¿Cuál utilizamos?	69
3.3.4.	Conclusiones	70
3.4.	OPTIMIZACIÓN DE RENDIMIENTO Y ESCALABILIDAD	72
3.4.1.	Escalabilidad y Rendimiento	72
3.4.1.1.	Escalamiento Vertical y Horizontal	72
3.4.1.2.	Ventajas y Desventajas:	72
3.4.1.3.	Uso en HawkEye	73
3.4.1.4.	Impacto en el Proyecto	73
3.4.2.	VENTAJAS DE MONGODB	74
3.4.2.1.	Modelo de Datos Flexible	74
3.4.2.2.	Escalamiento Horizontal Eficiente	74
3.4.2.3.	Alta Disponibilidad y Tolerancia a Fallos	74
3.4.2.4.	Índices y Consultas Eficientes	74
3.4.2.5.	Manejo de Grandes Volúmenes de Datos en Tiempo Real:	75
3.4.2.6.	Desarrollo Ágil y Esquema Dinámico	75
3.4.2.7.	Ecosistema y Comunidad Activa:	75
3.4.2.8.	Conclusión	75
3.4.3.	Rendimiento Iterativo	77
3.4.3.1.	Desarrollo Incremental	77
3.4.3.2.	Optimización de Funcionalidades Existentes	77
3.4.3.3.	Adopción de Nuevas Tecnologías	77
4.	CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	79

4.1.	ENCUESTA DE VIABILIDAD DE PROYECTO HAWKEYE	79
4.2.	RESUMEN DE ENCUESTA	79
4.3.	ANÁLISIS ENCUESTA.....	80
4.3.1.	DISPOSICIÓN DE SERVICIO.....	80
4.3.1.1.	Pregunta	80
4.3.1.2.	Contexto y Objetivo de la Pregunta	80
4.3.1.3.	Resultados	80
4.3.1.4.	Implicaciones y Conclusiones.....	80
4.3.2.	SEGURIDAD DE TUS SERES QUERIDOS	81
4.3.2.1.	Pregunta	81
4.3.2.2.	Contexto y Objetivo de la Pregunta	81
4.3.2.3.	Resultados	81
4.3.2.4.	Implicaciones y Conclusiones.....	81
4.3.3.	COMPROMISO FINANCIERO	82
4.3.3.1.	Pregunta	82
4.3.3.2.	Contexto y Objetivo de la Pregunta	82
4.3.3.3.	Resultados	82
4.3.3.4.	Implicaciones y Conclusiones.....	82
4.3.4.	RANGOS DE PRECIOS.....	83
4.3.4.1.	Pregunta	83
4.3.4.2.	Contexto y Objetivo de la Pregunta	83
4.3.4.3.	Resultados	83
4.3.4.4.	Implicaciones y Conclusiones.....	83
4.3.5.	NECESIDAD EN EL MERCADO	84
4.3.5.1.	<i>Pregunta</i>	84
4.3.5.2.	<i>Contexto y Objetivo de la Pregunta</i>	84
4.3.5.3.	<i>Resultados</i>	84
4.3.5.4.	<i>Implicaciones y Conclusiones</i>	84

4.3.6.	ASPECTO MÁS IMPORTANTE	85
4.3.6.1.	<i>Pregunta</i>	85
4.3.6.2.	<i>Contexto y Objetivo de la Pregunta</i>	85
4.3.6.3.	<i>Resultados</i>	85
4.3.6.4.	<i>Implicaciones y Conclusiones</i>	85
4.3.7.	PRIVACIDAD Y PROTECCIÓN DE DATOS	86
4.3.7.1.	<i>Pregunta</i>	86
4.3.7.2.	<i>Contexto y Objetivo de la Pregunta</i>	86
4.3.7.3.	<i>Resultados</i>	86
4.3.7.4.	<i>Implicaciones y Conclusiones</i>	86
4.3.8.	EXPERIENCIAS PREVIAS	87
4.3.8.1.	<i>Pregunta</i>	87
4.3.8.2.	<i>Contexto y Objetivo de la Pregunta</i>	87
4.3.8.3.	<i>Respuestas Relevantes:</i>	87
4.3.8.4.	<i>Implicaciones y Conclusiones</i>	87
4.3.9.	ALCANCE DEL SERVICIO	88
4.3.9.1.	<i>Pregunta</i>	88
4.3.9.2.	<i>Contexto y Objetivo de la Pregunta</i>	88
4.3.9.3.	<i>Resultados</i>	88
4.3.9.4.	<i>Implicaciones y Conclusiones</i>	88
4.3.10.	ALCANCE DEL SERVICIO.....	89
4.3.10.1.	<i>Pregunta</i>	89
4.3.10.2.	<i>Contexto y Objetivo de la Pregunta</i>	89
4.3.10.3.	<i>Resultados</i>	89
4.3.10.4.	<i>Implicaciones y Conclusiones</i>	89
4.4.	CONCLUSIÓN DE LA ENCUESTA Y VALIDACIÓN DE LA SOLUCIÓN.....	90
5.	CAPÍTULO 5: CONCLUSIONES	92
5.1.	ALCANCES Y LIMITACIONES DE LA PROPUESTA DE SOLUCIÓN	92

5.2.	RESULTADOS Y VALIDEZ DE LOS OBJETIVOS FORMULADOS	93
5.3.	CONTRIBUCIONES Y APLICACIONES CLAVE DEL TRABAJO.....	94
5.4.	IMPACTO Y APOORTE A LA COMUNIDAD	95
5.5.	DESARROLLO DE HAWKEYE	97
5.6.	CONCLUSIÓN SOBRE EL DESARROLLO DEL BACKEND	98
5.6.1.	IMPORTANCIA DE LA ARQUITECTURA.....	98
5.6.2.	USO ESTRATÉGICO DE HERRAMIENTAS.....	98
5.6.3.	APROVECHAMIENTO DE TECNOLOGÍAS EMERGENTES	98
5.6.4.	ESCALABILIDAD PARA EL FUTURO	98
5.6.5.	ADAPTABILIDAD A CAMBIOS EN LA RECEPCIÓN DE DATOS	98
6.	REFERENCIAS BIBLIOGRÁFICAS	100

INDICE DE FIGURAS

Figura 1 Logo de la organización – “Hawkeye Security”	19
Figura 2 Órdenes de Investigar por Presunta Desgracia, año 2020 y primer semestre enero a junio 2021.....	20
Figura 3 Interfaz de IntelliJ IDEA Community	31
Figura 4 Interfaz de MongoDB Compass	32
Figura 5 Interfaz de Postman	33
Figura 6 Interfaz de RailWay	34
Figura 7 Interfaz de GitHub Repositorio Backend HawkEye	35
Figura 8 Interfaz de Notion	36
Figura 9 Esquema No Relacional Base de Datos HawkEye	37
Figura 10 Arquitectura de Servicio HawkEye	43
Figura 11 Relación Objeto, ORM y Base de Datos	46
Figura 12 Arquitectura Monolítica Vs ORM	47
Figura 13 Caso de uso crear perfil	62
Figura 14 Caso de uso editar perfil	63
Figura 15 Caso de uso eliminar perfil.....	63
Figura 16 Caso de uso ver mapa	64
Figura 17 Caso de uso designar vallado virtual	64
Figura 18 Caso de uso zonas seguras	65
Figura 19 Caso de uso ver historial de recorrido	65
Figura 20 Caso de uso registrar pulsera	66
Figura 21 Caso de uso ver historial de alertas	66
Figura 22 Caso de uso ejecutar una alerta	67

Figura 23 Caso de uso recibir notificación	67
Figura 24 Caso de uso simular alerta	68
Figura 25 Modelo Vista Controlador HawkEye	70
Figura 26 Modelo Vista Controlador Usuario.....	71
Figura 27 Escalamiento Vertical vs Horizontal	72
Figura 28 Arquitectura Mongo DB.....	73

INDICE DE TABLAS

Tabla 1 Historia de Usuario Asignación de Monitoreo	49
Tabla 2 Historia de Usuario Monitoreo de Familiar	50
Tabla 3 Historia de Usuario Fijado de Perímetro.....	50
Tabla 4 Historia de Usuario Duración de la Batería	51
Tabla 5 Historia de Usuario Botón SOS	51
Tabla 6 Historia de Usuario Vinculación de Contactos de Emergencia	52
Tabla 7 Historia de Usuario Ver Ubicación y estado de la Persona desde Cualquier Dispositivo	52
Tabla 8 Historia de Usuario Servicio al Cliente.....	53
Tabla 9 Historia de Usuario notificación Salida y Entrada de Áreas Especificas	53
Tabla 10 10 Historia de Usuario Sonidos de Alertas Personalizados	54
Tabla 11 Historia de Usuario Interfaz Intuitiva y Sencilla	55
Tabla 12 Ajuste de Velocidad Máxima.....	55
Tabla 13 Historia de Usuario Programación del Horario de Funcionamiento	56
Tabla 14 Historia de Usuario Resistencia a Líquidos.....	56
Tabla 15 Historia de Usuario Batería Recargable	58
Tabla 16 Historia de Usuario Autorización de Ingreso.....	58
Tabla 17 Historia de Usuario Variedad de Productos	59
Tabla 18 Historia de Usuario Procedimiento SOS.....	59
Tabla 19 Historia de Usuario Autorización de Cambio en Configuración.....	60
Tabla 20 Historia de Usuario Configuración de Notificaciones de Estado	61

INTRODUCCIÓN

La seguridad personal es una preocupación fundamental en la sociedad actual, donde la creciente complejidad de los entornos urbanos y las cambiantes dinámicas sociales han dado lugar a desafíos significativos en la protección de individuos vulnerables. En este contexto, el proyecto HawkEye surge como una respuesta innovadora y efectiva para abordar estas preocupaciones y proporcionar tranquilidad a las personas que necesitan supervisar la ubicación y el bienestar de sus seres queridos.

Naturaleza del Problema y Propuesta de Solución

El aumento alarmante en el número de secuestros y desapariciones en los últimos años en Chile y en otras partes del mundo ha generado una necesidad apremiante de soluciones de seguridad personal más efectivas y accesibles. En 2022, los secuestros aumentaron en un 79% en Chile, afectando a familias y comunidades en todo el país. Además, la preocupación por la seguridad de niños, adultos mayores y personas con discapacidad ha llevado a una demanda creciente de sistemas de monitoreo y seguimiento personalizados. HawkEye se posiciona como una solución integral y altamente personalizable que utiliza tecnología GPS y API REST para proporcionar un monitoreo en tiempo real y alertas precisas. Este proyecto se basa en un servicio de Spring Boot conectado a una Base de Datos MongoDB que atiende a las aplicaciones móviles, el sitio web y el módulo ESP32 (Pulsera GPS) que constituyen el sistema HawkEye.

Metodología y Estructura del Documento

A lo largo de esta memoria, se detallará en profundidad la arquitectura técnica de HawkEye, centrándose en la implementación del Backend para la comunicación entre las diversas partes del sistema y la Base de Datos. Se explicará cómo se diseñaron y desarrollaron los microservicios *GET/POST* que gestionan las solicitudes desde la aplicación móvil, el sitio web y el módulo ESP32.

El documento se estructura de manera coherente para proporcionar una visión completa del proyecto. Comenzaremos con una revisión de la arquitectura del sistema, seguida de una descripción detallada de la implementación de la API REST y su conexión con la Base de Datos. Luego, se presentarán los resultados clave de este enfoque técnico y su relevancia en la creación de una solución efectiva para la seguridad personal.

1. CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1. ORGANIZACIÓN

1.1.1. NOMBRE DE LA ORGANIZACIÓN

HAWKEYE SECURITY

1.1.2. INTEGRANTES

Este título de memoria es uno de los cinco títulos que se están realizando a la par para la conformación del Software que lleva por título “HAWKEYE”, de los cuales se desprenden las siguientes áreas:

1.- Front-WEB 2.- Front-APPS 3.- Backend Apis 4.-Backend Datos 5.- Ciberseguridad Hawkeye.

1.1.3. LOGO DE LA ORGANIZACIÓN



Figura 1 Logo de la organización – “Hawkeye Security”

1.2. PROBLEMÁTICA

El secuestro es un delito que ha aumentado en los últimos años en Chile, dejando a las familias y comunidades afectadas por la pérdida de un ser querido. De acuerdo con los datos que maneja Carabineros, en 2022 hubo 508 secuestros, lo que comparado a 2021 mostró un aumento del 79% a nivel nacional¹. Desde 2017 a 2022, se contabilizaron 1.925 secuestros en todo el país, lo que muestra una tendencia al alza en la cantidad de casos reportados. Además, durante 2020 se recibieron 8.824 órdenes de investigar el paradero de personas que se consideran extraviadas por sus familiares o personas cercanas.

¹ https://www.cnnchile.com/pais/carabineros-secuestros-chile-aumentaron-79-2022_20230117/

HawkEye Backend: Seguridad Personal en Tiempo Real

Región	AÑO 2020 ENE-DIC		Total 2020	AÑO 2021 ENE-JUN		Total 2021	Total general
	Ubicado	Sin ubicar		Ubicado	Sin ubicar		
XV	136	31	167	44	17	61	228
I	178	20	198	148	23	171	369
II	307	36	343	138	18	156	499
III	110	7	117	70	4	74	191
IV	326	37	363	112	12	124	487
V	969	157	1126	435	39	474	1600
RM	2271	200	2471	871	73	944	3415
VI	317	10	327	275	15	290	617
VII	343	29	372	196	6	202	574
XVI	293	17	310	143	3	146	456
VIII	1217	146	1363	620	74	694	2057
IX	645	95	740	299	44	343	1083
XIV	189	10	199	67	9	76	275
X	569	42	611	214	7	221	832
XI	73	3	76	23	1	24	100
XII	34	7	41	16	1	17	58
Total general	7977	847	8824	3671	346	4017	12841

Figura 2 Órdenes de Investigar por Presunta Desgracia, año 2020 y primer semestre enero a junio 2021

Esta situación se agrava cuando los afectados son niños, adultos mayores o personas con discapacidad, quienes son más vulnerables ante estas situaciones y requieren de una respuesta rápida y efectiva. La falta de herramientas y tecnologías especializadas para la búsqueda y localización de personas desaparecidas aumenta el tiempo y la dificultad para encontrar a estas personas con vida.

No solo observamos la problemática de los secuestros, sino también la frecuente ocurrencia de casos de niños menores de cinco años extraviados. Además, el aumento de los casos de Alzheimer en adultos mayores² ha generado un gran temor entre sus familiares, creando una sensación de incertidumbre al regresar a casa. Por lo cual, hoy es indispensable utilizar una herramienta para reducir los tiempos de búsqueda.

1.2.1. ACTORES INVOLUCRADOS

Como ya se mencionó en el punto 1.2 (Problemática), nuestros actores involucrados y objetivo a resguardos son niños, personas con discapacidad y tercera edad, no obstante, nuestro mercado objetivo se expande más allá. Este software tiene las bases para extenderse a un cuidado de personas masivo, eso quiere decir, que está adaptado para que en futuras versiones se implemente en establecimientos, instituciones o actividades masivas.

² <https://www.bbc.com/mundo/noticias-59917913>

HawkEye Backend: Seguridad Personal en Tiempo Real

Realizamos el siguiente estudio de mercado en el cual segmentamos a nuestros futuros y posibles clientes en diferentes parámetros que vamos a mencionar a continuación:

1.2.2. Segmentación Geográfica

Objetivo: Habitantes de Chile³.

Tamaño: 19.212.363 personas.

Descripción: Se ha identificado a la nación de Chile como el mercado inicial para la venta y distribución del servicio de "HAWKEYE". El producto será promocionado y distribuido a lo largo del país.

1.2.3. Segmentación Demográfica

Objetivo: Personas de diferentes edades, géneros, niveles socioeconómicos y educativos.

Tamaño: 11.983.846 personas.

Descripción: Dentro de este segmento, se encuentran: Niños, adolescentes, adultos y adultos mayores. Cada grupo etario tiene necesidades diferentes, por lo que HAWKEYE puede ser adaptado según el usuario.

Género: No se discrimina por género; el producto es útil para todos por igual.

Nivel Socioeconómico: Aunque el producto puede ser útil para todas las clases sociales, se espera un interés particular de las clases media y alta, debido a la capacidad económica y la disposición a invertir en seguridad.

Nivel Educacional: Personas con educación básica hasta nivel universitario y más. La aplicación está diseñada para ser intuitiva y fácil de usar para todos los niveles educativos.

1.2.4. Segmentación por Comportamiento

Objetivo: Personas dispuestas a invertir en seguridad personal⁴.

Tamaño: 3.374.554 personas.

Descripción: Este segmento se refiere a individuos que valoran la seguridad y están dispuestos a pagar una tarifa mensual por ella. En este caso, son aquellos dispuestos

³ <https://www.populationpyramid.net/es/chile/2021/>

⁴ <https://percepcioneseconomicas.cl/>

HawkEye Backend: Seguridad Personal en Tiempo Real

a invertir más de \$10.000 pesos chilenos al mes para garantizar su seguridad o la de sus seres queridos.

Motivación: Preocupación por la seguridad personal y de seres queridos.

Patrón de Compra: Estos consumidores buscan soluciones efectivas y confiables. Poder recibir notificaciones en tiempo real y el botón SOS son características muy atractivas para ellos.

En conclusión, tenemos un mercado atractivo de alrededor de: **3.374.554 personas** dispuestas a obtener un producto el cual les garantice “seguridad”.

1.3. COMPETENCIA ACTUAL

En el mercado actual chileno encontramos ideas similares a lo que “*Hawkeye*” ha planteado como solución, pero en base a diferentes análisis como lo han sido encuestas y porcentajes de uso de aquellos productos, hemos observado que aquellas han sido deficientes ante el problema observado.

Una de estas soluciones mencionadas es “*SoyMomo*”, este es una tecnología desarrollada en base a un prototipo de *SmartWatch* en el cual el tutor a cargo tiene un control parental en todo momento, adicionalmente cuenta con la opción de llamadas, videollamadas, mensajes, entre otras aplicaciones que contiene el mismo.

Las complicaciones o deficiencias de este producto han sido la robustez de su dispositivo GPS y lo complicado que es el uso del software en base al tutor.

Otra de las soluciones que encontramos en el mercado es “*Localizatech*”, el cual es un dispositivo para personas con mayor riesgo de extravío como lo son personas con discapacidad o personas con Alzheimer, este es un dispositivo GPS que va incluido dentro de las vestimentas o accesorios de la persona cuidado, como lo puede ser tipo lanyard, pulsera o clip (Enganche en el cinturón).

Las complicaciones o deficiencias que se encontraron en este producto fue nuevamente la robustez del dispositivo GPS, la incomodidad al utilizarlo y lo “poco moderno” de su software. Todo esto ha hecho que el producto no tenga un amplio comercio ni crezca en el mercado.

1.4. MOTIVACIÓN

La motivación de la realización de este software nace ante la preocupación social de las altas pérdidas en el segmento de mercado mencionado. Los principales actores que conforman la elaboración de este sistema han estado preocupados por la desaparición de unos seres queridos. Es por ello que nace “*Hawkeye*”, como propuesta de solución, adicionalmente se ha llegado a estudios de prototipo que han dado validación a la propuesta actual.

1.5. OBJETIVOS

El objetivo principal de *Hawkeye* es reducir los tiempos de respuesta en casos de secuestros, desapariciones o pérdidas de personas vulnerables, logrando esto mediante el uso de un dispositivo GPS de vanguardia. Nuestra misión es proporcionar a los tutores y cuidadores la tranquilidad y confianza de que las personas a su cargo, en particular niños, adultos mayores y personas con discapacidad, están seguras y protegidas.

Como objetivos secundarios, *Hawkeye* se enfoca en la creación de una experiencia de usuario excepcional. Esto se logra mediante el diseño de una interfaz intuitiva y cómoda, que asegure la facilidad de uso para una amplia gama de usuarios. Además, Hawkeye se compromete a garantizar la protección de datos, cumpliendo con todas las regulaciones y estándares de seguridad de datos, para preservar la privacidad de los usuarios.

2. CAPÍTULO 2: MARCO CONCEPTUAL

2.1. CONCEPTOS TÉCNICOS

2.1.1. API REST

Una API REST es una interfaz de programación de aplicaciones (API) que utiliza el protocolo de transferencia de hipertexto (HTTP) para transferir datos. Las APIs REST se utilizan en una variedad de aplicaciones, como la web, la movilidad y la nube.

2.1.2. Base de datos

Una base de datos es una colección organizada de datos. Las bases de datos se utilizan para almacenar información de manera eficiente y accesible.

2.1.3. Bluetooth

Bluetooth es una tecnología inalámbrica que permite la conexión entre dispositivos a corta distancia. El Bluetooth se utiliza en una variedad de dispositivos, como los teléfonos inteligentes, los ordenadores portátiles y los altavoces.

2.1.4. Casos de uso

Los casos de uso son una herramienta de análisis y diseño que se utiliza para describir la funcionalidad de un sistema de software desde la perspectiva de los usuarios. Los casos de uso se utilizan para identificar los requisitos funcionales de un sistema, así como para comunicar esos requisitos a los interesados.

2.1.5. Cobertura celular

La cobertura celular es el área en la que un operador celular puede proporcionar servicios de telecomunicaciones.

2.1.6. Cobertura GPS

La cobertura GPS es el área de la Tierra en la que un receptor GPS puede determinar su ubicación geográfica.

2.1.7. Desarrollo ágil

El desarrollo ágil es un enfoque de desarrollo de software que se basa en la entrega de software a intervalos regulares, llamados sprints. El desarrollo ágil se centra en la colaboración entre los equipos de desarrollo y los usuarios, así como en la flexibilidad y la capacidad de respuesta a los cambios.

2.1.8. Dispositivo GPS

Un dispositivo GPS es un dispositivo que utiliza el sistema de posicionamiento global (GPS) para determinar su ubicación geográfica.

2.1.9. Escalamiento vertical

El escalamiento vertical es un método de escalado de un sistema informático que consiste en aumentar el rendimiento de los componentes individuales del sistema. El escalamiento vertical se suele realizar añadiendo más potencia a los componentes existentes, como más memoria, más CPU o más espacio de almacenamiento.

2.1.10. Escalamiento horizontal

El escalamiento horizontal es un método de escalado de un sistema informático que consiste en añadir más componentes al sistema. El escalamiento horizontal se suele realizar añadiendo más servidores, más bases de datos o más instancias de aplicaciones.

2.1.11. Geolocalización

La geolocalización es la capacidad de determinar la ubicación geográfica de un objeto, persona o evento.

2.1.12. Hardware de servidor

El hardware de servidor es el conjunto de componentes físicos que componen el servidor. El hardware de servidor suele ser más potente que el hardware de un ordenador personal.

2.1.13. Historias de usuario

Las historias de usuario son una herramienta de desarrollo de software que se utiliza para describir los requisitos de un sistema desde la perspectiva de los usuarios. Las historias de usuario se utilizan para comunicar los requisitos a los desarrolladores y para proporcionar una base para la planificación y el desarrollo del software.

2.1.14. Hosting

El hosting es el servicio que ofrece un proveedor de servicios de Internet (ISP) para alojar un sitio web o una aplicación web. El hosting se puede realizar en un servidor físico o en una plataforma en la nube.

2.1.15. HTTP

HTTP es el protocolo de transferencia de hipertexto, que es el protocolo estándar para la comunicación en la World Wide Web.

2.1.16. HTTP method

Un método HTTP es una acción que se puede realizar en un recurso. Los métodos HTTP más comunes son GET, POST, PUT y DELETE.

2.1.17. HTTPS

HTTPS es una versión segura de HTTP que utiliza el protocolo de capa de sockets seguros (SSL) para cifrar la comunicación entre el cliente y el servidor.

2.1.18. IP

Una dirección IP es una dirección única que se utiliza para identificar un dispositivo en una red.

2.1.19. Java

Java es un lenguaje de programación general-propósito que se utiliza para crear una variedad de aplicaciones, como aplicaciones web, aplicaciones móviles y aplicaciones de escritorio.

2.1.20. Lenguaje de programación

Un lenguaje de programación es un conjunto de reglas que se utilizan para crear programas informáticos.

2.1.21. Modelo de datos

Un modelo de datos es una representación abstracta de los datos que se utilizarán en un sistema de software. El modelo de datos se utiliza para describir la estructura de los datos, los tipos de datos y las relaciones entre los datos.

2.1.22. Modelo de servicio

Un modelo de servicio es una representación de un sistema de software que se centra en los servicios que proporciona el sistema. El modelo de servicio se utiliza para describir la funcionalidad del sistema, así como las interfaces que se utilizan para acceder a la funcionalidad.

2.1.23. MVC (Modelo Vista Controlador)

MVC es un patrón de diseño de software que separa la lógica de la presentación de la lógica de la aplicación. MVC se utiliza en una variedad de aplicaciones, como las aplicaciones web, las aplicaciones móviles y las aplicaciones de escritorio.

2.1.24. Navegación GPS

La navegación GPS es el proceso de utilizar un dispositivo GPS para determinar la ruta a un destino.

2.1.25. NoSQL

NoSQL es un término genérico que se refiere a las bases de datos que no se adhieren al modelo relacional. Las bases de datos NoSQL son más flexibles y escalables que las bases de datos relacionales, y son especialmente adecuadas para almacenar grandes cantidades de datos no estructurados.

2.1.26. Operador celular

Un operador celular es una empresa que proporciona servicios de telecomunicaciones a través de una red celular.

2.1.27. ORM (Mapeo de Objetos a Relacionales)

El mapeo de objetos a relacionales (ORM) es una técnica de programación que permite a los desarrolladores mapear objetos en un lenguaje de programación orientado a objetos a filas en una base de datos relacional. Los ORMs proporcionan una capa de abstracción entre el código orientado a objetos y la base de datos relacional, lo que facilita a los desarrolladores trabajar con datos.

2.1.28. Patrones de diseño

Los patrones de diseño son soluciones generales a problemas comunes en el diseño de software. Los patrones de diseño se utilizan para mejorar la calidad del software, haciéndolo más flexible, escalable y fácil de mantener.

2.1.29. Precisión GPS

La precisión GPS es la exactitud con la que un receptor GPS puede determinar la ubicación geográfica de un objeto. La precisión GPS depende de una variedad de factores, como la cantidad de satélites GPS que están visibles, la potencia de la señal de los satélites GPS y la calidad del receptor GPS.

2.1.30. Puerto

Un puerto es un número asignado a un servicio en un servidor. Los puertos se utilizan para identificar los servicios que se ejecutan en un servidor.

2.1.31. Receptor GPS

El receptor GPS es el dispositivo que recibe las señales de los satélites GPS y calcula la ubicación geográfica del objeto. Los receptores GPS suelen estar integrados en dispositivos como los navegadores GPS, las pulseras de actividad y los teléfonos inteligentes.

2.1.32. Rendimiento iterativo

El rendimiento iterativo es una técnica de desarrollo de software que consiste en desarrollar el software de forma incremental, añadiendo nuevas características o funcionalidades en cada iteración. El rendimiento iterativo se basa en la idea de que es más fácil y eficiente desarrollar software de forma incremental que de una sola vez.

2.1.33. REST

REST es un acrónimo de Representational State Transfer, que es un estilo arquitectónico para aplicaciones web. REST se basa en un conjunto de principios que facilitan la creación de aplicaciones web escalables y fáciles de mantener.

2.1.34. Scrum

Scrum es un marco de desarrollo ágil que se utiliza para gestionar proyectos de software. Scrum se basa en un conjunto de roles, eventos y artefactos que ayudan a los equipos a planificar, desarrollar y entregar software de manera eficiente.

2.1.35. Servicios de hosting

Los servicios de hosting ofrecen una variedad de características y funcionalidades, como el espacio de almacenamiento, la transferencia de datos y el ancho de

banda. Los servicios de hosting se pueden utilizar para alojar sitios web, aplicaciones web, bases de datos y otros recursos.

2.1.36. TLS

TLS (Transport Layer Security) es un protocolo de seguridad que se utiliza para cifrar la comunicación entre el cliente y el servidor. TLS se utiliza en una variedad de aplicaciones, como las páginas web seguras (HTTPS) y las aplicaciones de correo electrónico seguras.

2.1.37. UML

UML (Unified Modeling Language) es un lenguaje de modelado visual que se utiliza para describir y documentar software. UML se utiliza para una variedad de propósitos, como el diseño de sistemas, la documentación de requisitos y la comunicación entre los equipos de desarrollo.

2.1.38. URI

Una URI (Uniform Resource Identifier) es una dirección de un recurso en Internet. Las URIs se utilizan para identificar los recursos que se exponen a través de una API REST.

2.1.39. URL

Una URL (Uniform Resource Locator) es una forma específica de URI que se utiliza para identificar recursos en la World Wide Web. Las URLs son las direcciones que se utilizan para acceder a sitios web y otros recursos web.

2.1.40. Wearable

Un wearable es un dispositivo electrónico que se lleva puesto en el cuerpo. Los wearables suelen ser pequeños, ligeros y cómodos de llevar. Pueden utilizarse para una variedad de propósitos, como el seguimiento de la actividad física, la monitorización de la salud, la comunicación o la navegación.

2.2. METODOLOGÍAS

2.2.1. METODOLOGIA SCRUM

Como equipo a comienzo de año analizamos los atributos y fortalezas de cada integrante del equipo y como metodología de trabajo encontramos la metodología SCRUM más eficiente para el equipo. La cual definimos de la siguiente manera.

Scrum Máster: Miguel Contreras

Product Owner: Miguel Belmar

Equipo de Desarrollo: Sebastián Riquelme, Moisés Contreras y Tomás Mella.

El formato que utilizamos ante esta metodología fue realizar reuniones de planificación de Sprint, reuniones de revisión de Sprint, y de retrospectivas. Aquellas nos sirvieron para ir avanzando por etapas y distintos enfoques en lo que amerito el proyecto de forma general.

2.3. HERRAMIENTAS

Las herramientas que estamos utilizando para el desarrollo del software de “Hawkeye” son las siguientes:

- **Spring Boot:** Proporciona un enfoque rápido y sencillo para crear aplicaciones Java con el marco de trabajo Spring. Nos permite crear aplicaciones web y móviles de forma rápida y eficiente, con un mínimo de esfuerzo.
- **Spring Data MongoDB:** Es parte del ecosistema de Spring y proporciona una abstracción para trabajar con MongoDB. Permite el uso de repositorios y consultas para interactuar con la base de datos MongoDB.
- **RESTful API:** Un estilo de arquitectura para sistemas distribuidos, donde los recursos son accesibles a través de URLs y se manipulan utilizando métodos HTTP como GET, POST, PUT y DELETE.
- **IntelliJ IDEA Community:** Un entorno de desarrollo integrado (IDE) proporcionado por JetBrains. Ofrece características robustas para el desarrollo de software en Java y otros lenguajes. Sus funcionalidades incluyen un potente editor de código, refactorización inteligente, herramientas de depuración, integración con sistemas de control de versiones y soporte para una amplia gama de tecnologías y marcos de trabajo.

HawkEye Backend: Seguridad Personal en Tiempo Real

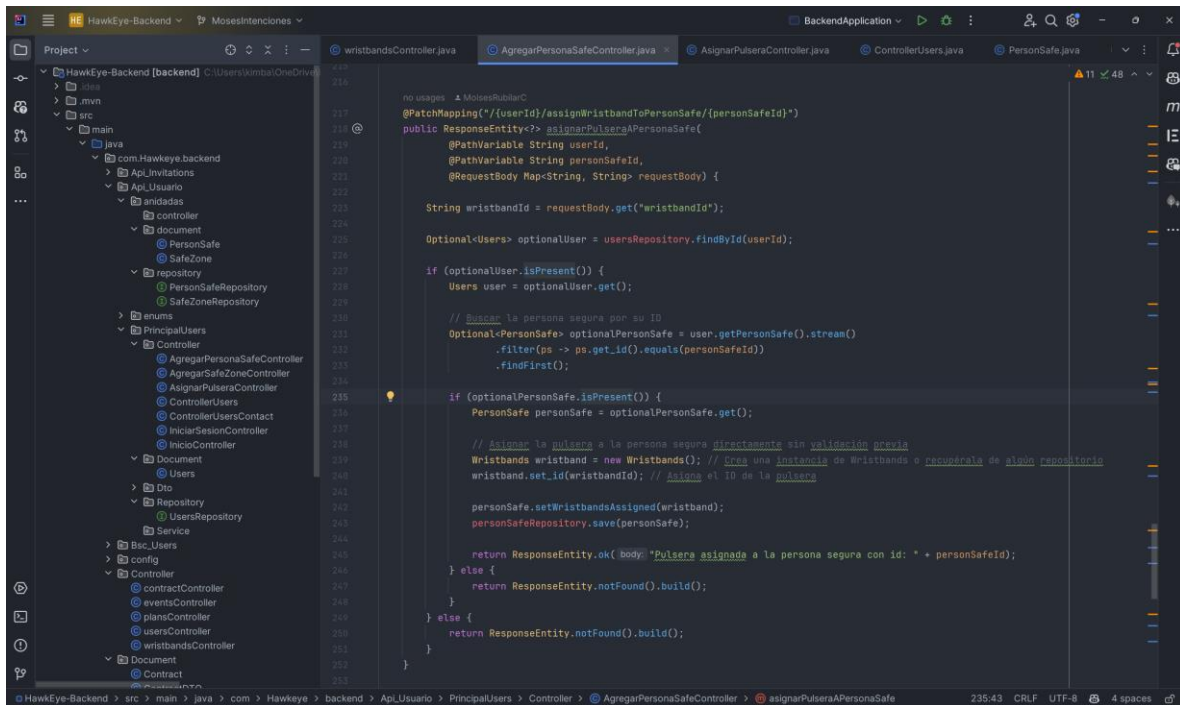


Figura 3 Interfaz de IntelliJ IDEA Community

- **Mapeo Objeto-Documento:** Se refiere a la forma en que los objetos Java se representan y almacenan en una base de datos NoSQL como MongoDB. En este caso, se utiliza la anotación **@Document** para mapear una clase Java a un documento MongoDB.
- **Controladores en Spring (Spring MVC):** Utilizados para manejar las solicitudes HTTP y enviar respuestas de vuelta al cliente. Los controladores están anotados con **@GetMapping**, **@PostMapping**, **@PatchMapping**, etc., para definir qué método manejará cada tipo de solicitud.
- **Inyección de Dependencias:** Es un patrón de diseño en el que se suministran objetos a una clase en lugar de que la clase misma cree esos objetos. Spring utiliza la inyección de dependencias para administrar los componentes y sus dependencias.
- **API (Interfaz de Programación de Aplicaciones):** Un conjunto de reglas, protocolos y herramientas para construir software y permitir la comunicación entre diferentes sistemas.
- **MongoDB Compass :** Es una potente GUI para consultar, agregar y analizar sus datos MongoDB en un entorno visual. Contiene una interfaz intuitiva para poder

HawKeye Backend: Seguridad Personal en Tiempo Real

simplificar la exploración y análisis de los datos almacenados en MongoDB, Consultas y Análisis Eficientes sus herramientas de consulta permiten realizar análisis profundos de los datos de manera ágil y eficiente.

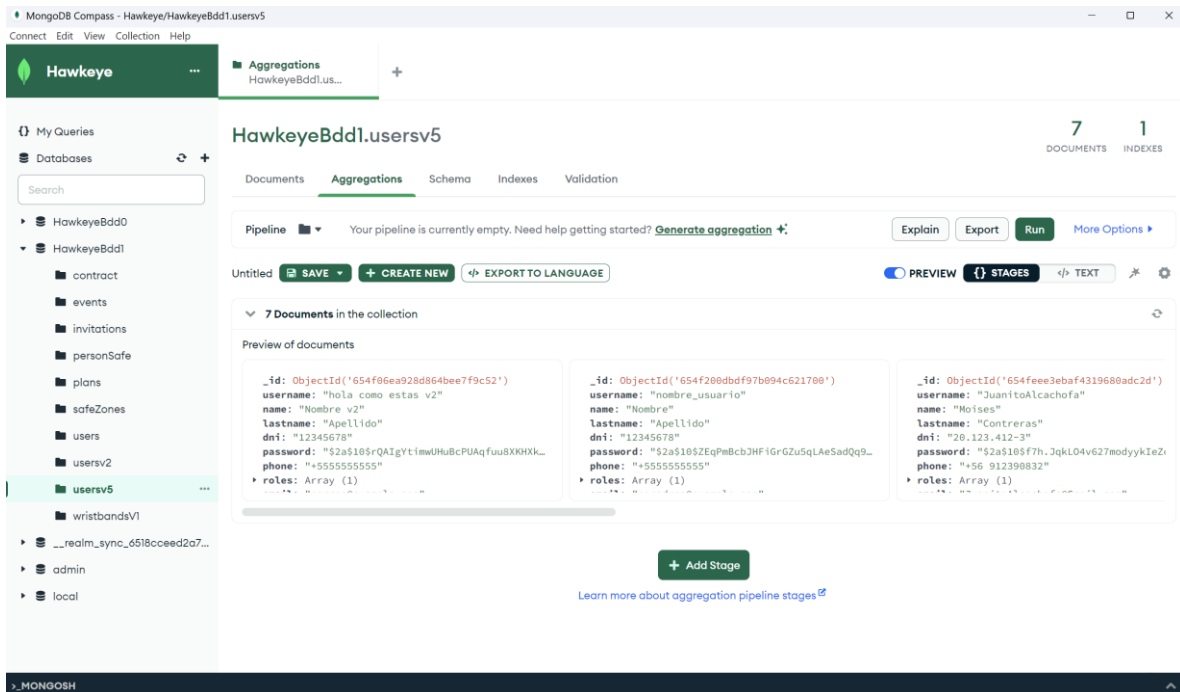


Figura 4 Interfaz de MongoDB Compass

- **Manejo de Excepciones:** Se utiliza **Optional** para manejar valores nulos y **ResponseEntity** para devolver respuestas HTTP con códigos y cuerpos personalizados.
- **Java Streams:** Se utiliza la API de streams de Java para realizar operaciones sobre colecciones de datos, como filtrar, mapear, y operaciones de reducción. En el código, se usa **stream()** y **anyMatch()** para verificar si un elemento cumple con una condición específica.
- **Anotaciones Lombok:** Utilizadas para simplificar la creación de getters, setters, constructores, y otros métodos comunes en Java. Reduce la cantidad de código boilerplate que se necesita escribir.
- **JSON Serialization/Deserialization:** Spring MVC utiliza Jackson (mediante anotaciones como **@RequestBody**) para convertir datos JSON provenientes de las solicitudes HTTP a objetos Java y viceversa.
- **MongoRepository:** Una interfaz proporcionada por Spring Data MongoDB que ofrece métodos para realizar operaciones CRUD en una base de datos MongoDB, como guardar, eliminar y buscar documentos.

HawkEye Backend: Seguridad Personal en Tiempo Real

- **Postman:** Postman es una plataforma que facilita la realización de pruebas de API al proporcionar un entorno completo y amigable para realizar solicitudes de métodos HTTPS a enlaces específicos. Sus características principales incluyen:
 - **Pruebas Simplificadas de API:** Permite ejecutar y validar solicitudes HTTP o HTTPS de manera sencilla, lo que facilita la verificación del comportamiento y la funcionalidad de las API.
 - **Entorno de Desarrollo Eficaz:** Proporciona una interfaz intuitiva que simplifica la creación, organización y ejecución de pruebas de API. Esto permite una depuración y desarrollo más ágil y eficiente.
 - **Gestión Integral:** Ofrece herramientas para organizar y guardar colecciones de solicitudes, lo que facilita la reutilización y el mantenimiento de las pruebas.

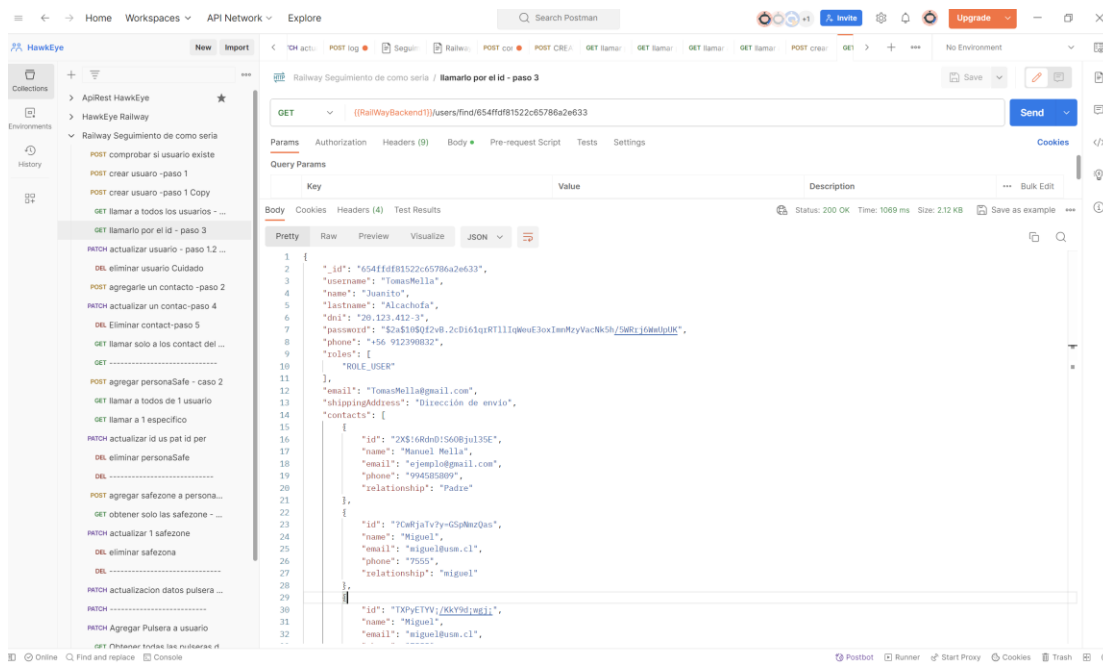


Figura 5 Interfaz de Postman

- **Uso de anotaciones:** En Java, las anotaciones son metadatos que se aplican a clases, métodos o campos para proporcionar información adicional al compilador o al tiempo de ejecución. Se utilizan anotaciones como **@DBRef**, **@Document**, **@GetMapping**, **@DeleteMapping**, **@PathVariable**, etc.
- **MongoDB:** Al ser una base de datos NoSQL, nos ofrece beneficios y ventajas, como la flexibilidad en el uso de los datos, la escalabilidad del proyecto y el alto rendimiento de este. Nos permite almacenar los datos de nuestro software de forma eficiente y flexible.

HawkEye Backend: Seguridad Personal en Tiempo Real

- **Railway:** Es una plataforma que ofrece la capacidad de crear y administrar entornos de desarrollo de manera ágil y eficiente. Sus características principales son:
 - **Infraestructura en la Nube Simplificada:** Proporciona una infraestructura basada en la nube que permite desarrollar y desplegar software de manera rápida y eficiente.
 - **Gestión Simplificada de Entornos:** Facilita la creación y administración de entornos de desarrollo, simplificando tareas como configuración, despliegue y mantenimiento.
 - **Entorno de Desarrollo Eficiente:** Ofrece una plataforma intuitiva que permite a los equipos de desarrollo trabajar de manera colaborativa en la creación y despliegue de aplicaciones

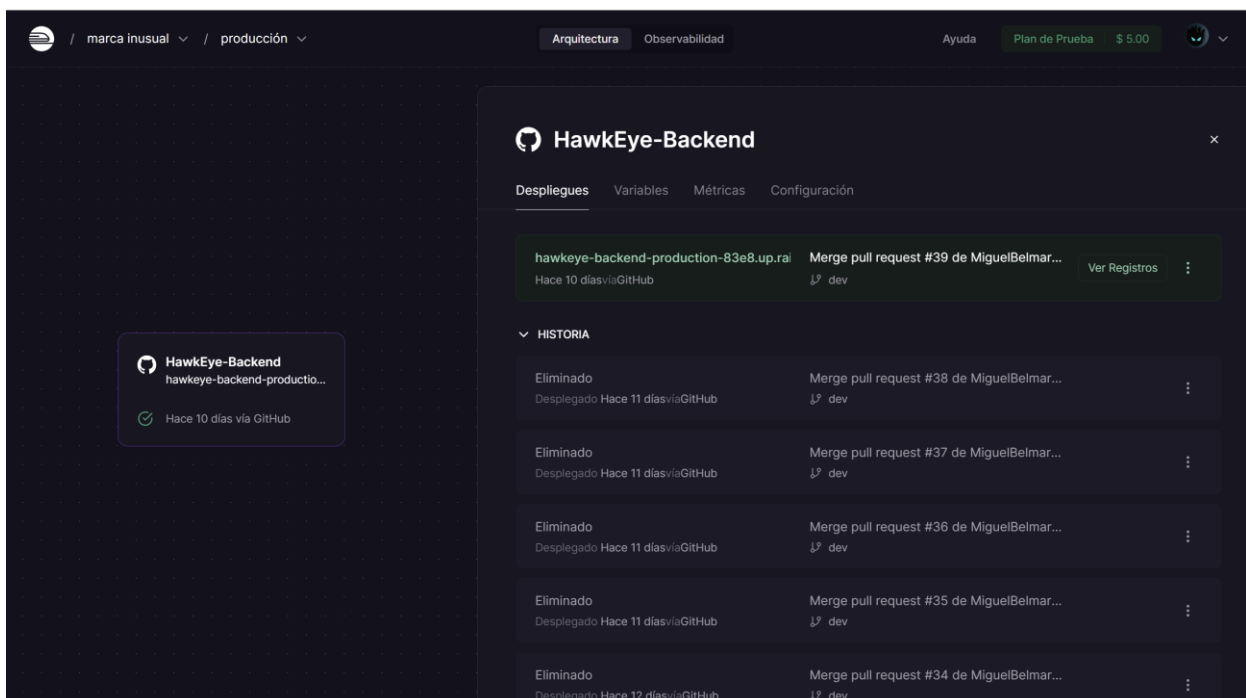


Figura 6 Interfaz de RailWay

- **GitHub:** Nos permite gestionar el código fuente de nuestro software de forma segura y eficiente. Nos proporciona una plataforma para almacenar, compartir y colaborar en el desarrollo de nuestro software.

HawkEye Backend: Seguridad Personal en Tiempo Real

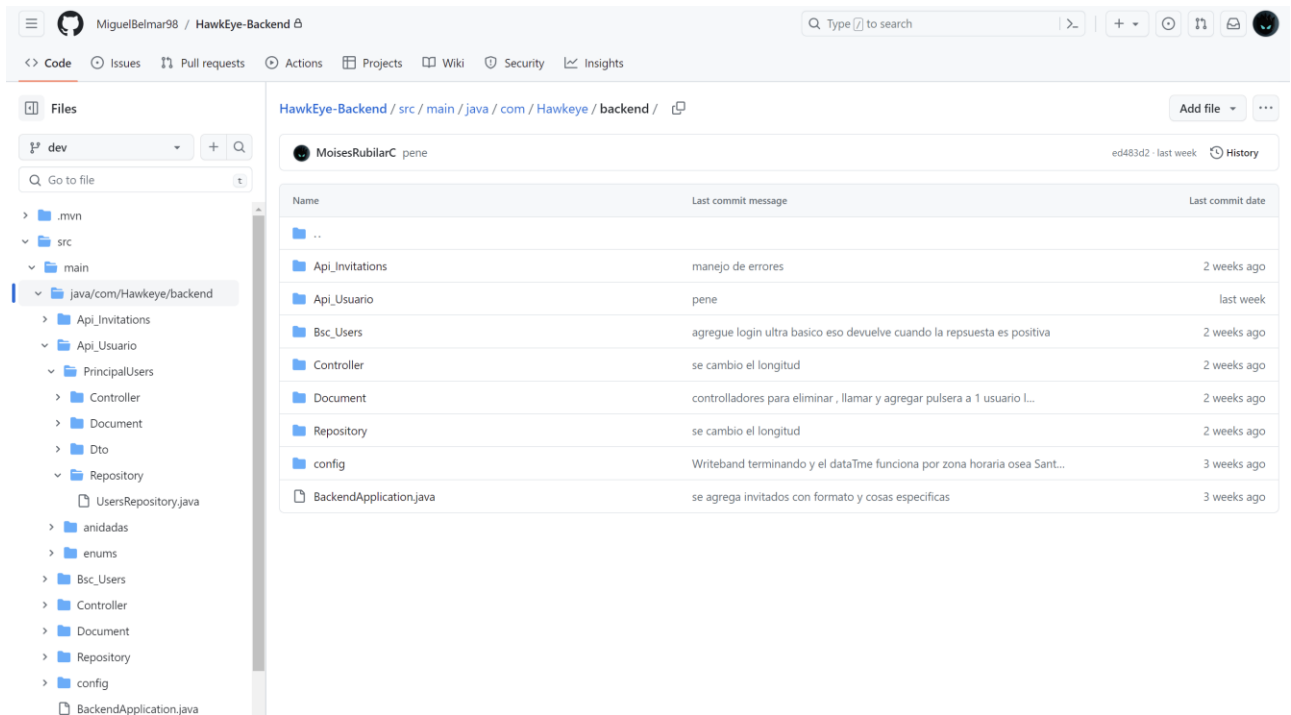


Figura 7 Interfaz de GitHub Repositorio Backend HawkEye

- **Android Studio:** Al ser un entorno de desarrollo integrado nos permite la conectividad entre las diferentes herramientas que estamos utilizando. Nos proporciona una interfaz de desarrollo integrada que nos permite desarrollar aplicaciones Android de forma rápida, eficiente y colaborativa .
- **ApiGoogleMaps:** Nos permite integrar la funcionalidad de Google Maps en nuestro software. Nos proporciona una API que nos permite acceder a los mapas de Google y utilizarlos en nuestro software.
- **Auth0:** es un servicio que nos provee todo el mecanismo para administrar la autorización y la autenticación de nuestras plataformas.
- **SSID y contraseña:** Identificador y clave de acceso de una red Wi-Fi, necesarios para que el ESP32 se conecte a la red.
- **Módulo GPS:** Dispositivo que recibe señales de satélites GPS para determinar la ubicación geográfica con precisión.
- **ESP32:** Tipo específico de placa de desarrollo de Arduino, que incluye conectividad WiFi y Bluetooth, útil para proyectos IoT (Internet de las cosas).
- **HTTPClient (Cliente HTTP):** Librería o componente utilizado en el código de Arduino para realizar solicitudes HTTP, en este caso, para enviar datos de ubicación al servidor.

HawkEye Backend: Seguridad Personal en Tiempo Real

- **Spring Data JPA:** Una extensión de Spring que simplifica el acceso a datos al integrar JPA y proporcionar funcionalidades adicionales para trabajar con bases de datos.
- **PasswordEncoder:** Interfaz de Spring Security utilizada para codificar y decodificar contraseñas
- **DTO (Data Transfer Object):** Objetos utilizados para transferir datos entre subsistemas de una aplicación.
- **Notion:** Es una herramienta integral que ofrece una amplia gama de funcionalidades para la gestión de proyectos. Permite organizar, planificar y colaborar en proyectos de manera efectiva al proporcionar un entorno versátil y flexible. Con la capacidad de crear tableros personalizables, bases de datos y listas de tareas, se adapta a distintos estilos de gestión y permite una estructuración personalizada. Además, facilita la colaboración entre equipos al ofrecer herramientas para compartir información, comentar en tiempo real y mantener un registro de cambios.

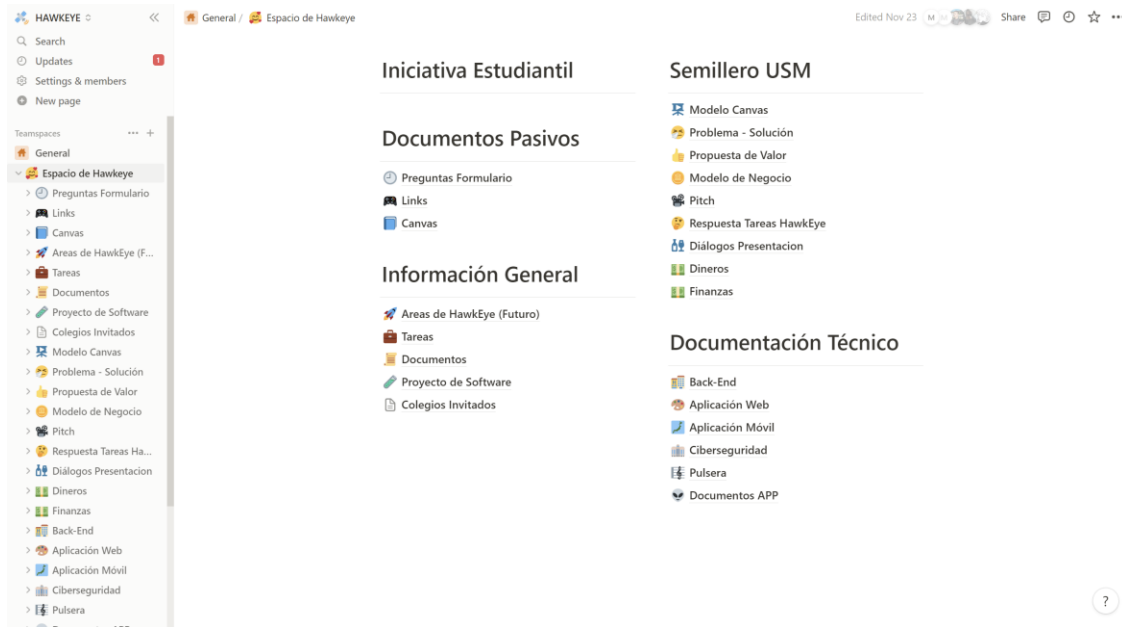


Figura 8 Interfaz de Notion

2.4. MODELOS

2.4.1. MODELO DE DATOS

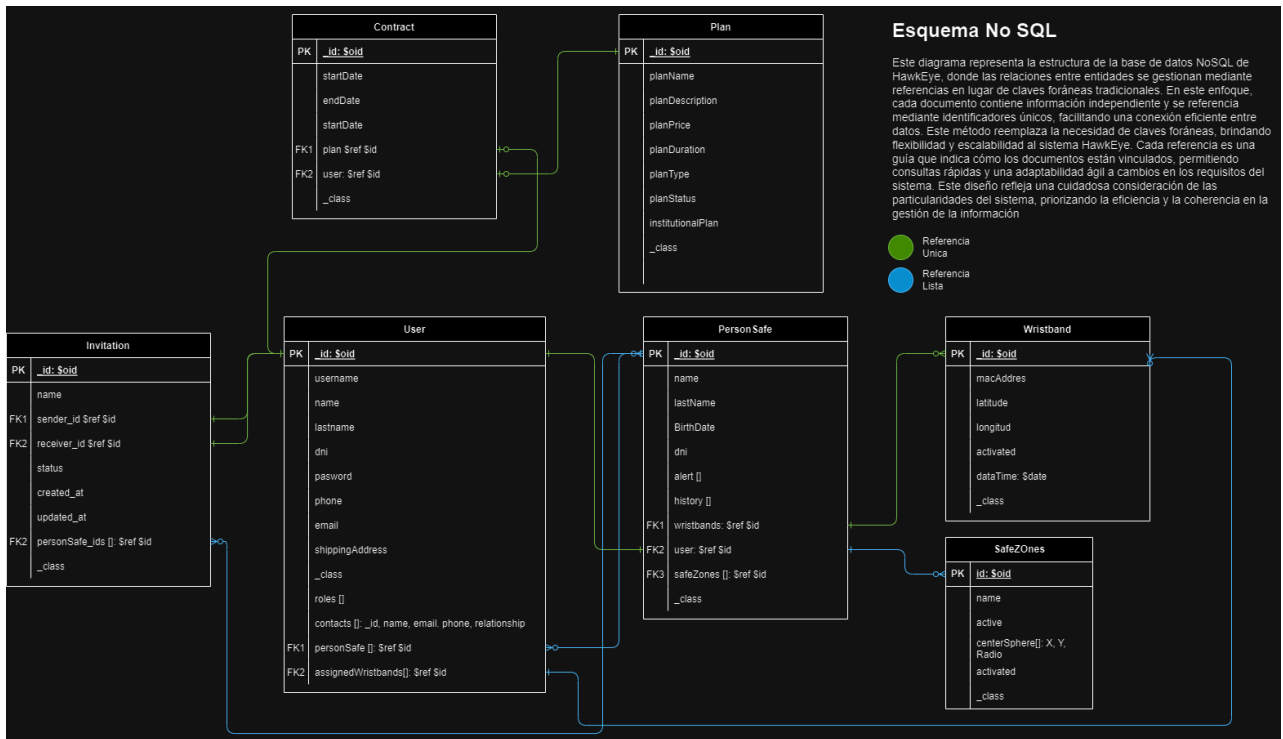


Figura 9 Esquema No Relacional Base de Datos HawkEye

El diagrama de arriba muestra un esquema no relacional para el sistema de HawkEye. Este esquema utiliza un modelo de datos documental, que almacena la información en documentos JSON. Cada documento representa una entidad en el sistema, como un usuario, una persona segura, una pulsera o una alerta.

Las relaciones entre los documentos se representan mediante campos de tipo document. Por ejemplo, el campo `personSafe` en la tabla `Wristband` hace referencia a un documento de la tabla `PersonSafe`. Esto indica que la pulsera está asociada a una persona segura.

El esquema no relacional tiene varias ventajas sobre el modelo relacional tradicional. Es más flexible y escalable, y permite almacenar datos de forma más eficiente. Además, es más fácil de aprender y usar.

En el caso específico del sistema de geolocalización, el esquema no relacional tiene las siguientes ventajas:

HawkEye Backend: Seguridad Personal en Tiempo Real

- Permite almacenar información de forma más eficiente. Por ejemplo, el campo location en la tabla History puede almacenar una lista de coordenadas de latitud y longitud, en lugar de almacenar cada coordenada como un campo separado.
- Es más fácil de escalar. El sistema puede crecer fácilmente añadiendo nuevos documentos sin necesidad de modificar el esquema.
- Es más flexible. El sistema puede adaptarse fácilmente a nuevos requisitos sin necesidad de realizar cambios importantes en el esquema.

2.4.1.1. Descripción de Documentos/ Tablas

2.4.1.1.1. User.

- username: Nombre de usuario del usuario.
- name: Nombre del usuario.
- lastName: Apellido del usuario.
- dni: Documento nacional de identidad del usuario.
- phoneNumber: Número de teléfono del usuario.
- email: Dirección de correo electrónico del usuario.
- password: Contraseña del usuario.
- class: Clase del usuario.
- status: Estado del usuario.
- personSafe: Identificador de la persona segura a la que pertenece el usuario.
- assignedWristbands: Lista de identificadores de las pulseras asignadas al usuario.

Los atributos de la tabla User proporcionan información sobre los usuarios del sistema de geolocalización. Los atributos id, username y password son necesarios para identificar y autenticar a los usuarios. Los atributos name, lastName, dni, phoneNumber, email proporcionan información básica sobre los usuarios. Los atributos status y personSafe proporcionan información sobre el estado del usuario y la persona segura a la que pertenece el usuario. Los atributos assignedWristbands proporciona información sobre las pulseras asignadas al usuario.

id: Identificador único del usuario.

2.4.1.1.2. PersonSafe.

- id: Identificador único de la persona segura.
- name: Nombre de la persona segura.
- lastName: Apellido de la persona segura.
- dni: Documento nacional de identidad de la persona segura.
- phoneNumber: Número de teléfono de la persona segura.
- email: Dirección de correo electrónico de la persona segura.
- class: Clase de la persona segura.
- status: Estado de la persona segura.
- safeZones: Lista de identificadores de las zonas seguras asociadas a la persona segura.

Los atributos de la tabla PersonSafe proporcionan información sobre las personas seguras del sistema de geolocalización. Los atributos id, name, lastName, dni, phoneNumber, email proporcionan información básica sobre las personas seguras. Los atributos status y safeZones proporcionan información sobre el estado de la persona segura y las zonas seguras asociadas a la persona segura.

2.4.1.1.3. Invitation.

- id: Identificador único de la invitación.
- sender_id: Identificador del usuario que envió la invitación.
- receiver_id: Identificador del usuario que recibió la invitación.
- name: Nombre de la invitación.
- status: Estado de la invitación.
- created_at: Fecha y hora de creación de la invitación.
- updated_at: Fecha y hora de última actualización de la invitación.
- personSafe_ids: Lista de identificadores de las personas seguras a las que está dirigida la invitación.
- class: Clase de Invitation.

Los atributos de la tabla PersonSafe proporcionan información sobre las personas seguras del sistema de geolocalización. Los atributos id, name, lastName, dni, phoneNumber, email proporcionan información básica sobre las personas seguras. Los atributos status y safeZones proporcionan información sobre el estado de la persona segura y las zonas seguras asociadas a la persona segura.

2.4.1.1.4. Wristband.

- id: Identificador único de la pulsera.
- macAddress: Dirección MAC de la pulsera.
- latitude: Latitud de la pulsera.
- longitude: Longitud de la pulsera.
- activated: Estado de activación de la pulsera.
- dateTime: Fecha y hora de activación de la pulsera.
- class: Clase de la pulsera.
- user_id: Identificador del usuario al que está asignada la pulsera.
- personSafe_id: Identificador de la persona segura a la que está asignada la pulsera.
- class: Clase de Invitation.

Los atributos de la tabla Wristband proporcionan información sobre las pulseras del sistema de geolocalización. Los atributos id, macAddress, latitude, longitude y activated proporcionan información básica sobre la pulsera. El atributo dateTime proporciona información sobre la fecha y hora en que se activó la pulsera. El atributo user_id proporciona información sobre el usuario al que está asignada la pulsera. El atributo personSafe_id proporciona información sobre la persona segura a la que está asignada la pulsera.

2.4.1.1.5. SafeZones.

- id: Identificador único de la pulsera.
- macAddress: Dirección MAC de la pulsera.
- latitude: Latitud de la pulsera.
- longitude: Longitud de la pulsera.
- activated: Estado de activación de la pulsera.
- dateTime: Fecha y hora de activación de la pulsera.
- class: Clase de la pulsera.
- user_id: Identificador del usuario al que está asignada la pulsera.
- personSafe_id: Identificador de la persona segura a la que está asignada la pulsera.

Los atributos de la tabla SafeZones proporcionan información sobre las zonas seguras del sistema de geolocalización. Los atributos id, name, active, centerSphere[], activated proporcionan información básica sobre la zona segura. El atributo personSafes proporciona información sobre las personas seguras asociadas a la zona segura.

2.4.1.1.6. Contract.

- id: Identificador único del contrato.
- startDate: Fecha de inicio del contrato.
- endDate: Fecha de finalización del contrato.
- plan_id: Identificador del plan asociado al contrato.
- user_id: Identificador del usuario asociado al contrato.
- class: Clase del contrato.

Los atributos de la tabla Contract proporcionan información sobre los contratos del sistema de geolocalización. Los atributos id, startDate, endDate, plan_id y user_id proporcionan información básica sobre el contrato.

2.4.1.1.7. Plan.

- id: Identificador único del plan.
- planName: Nombre del plan.
- planDescription: Descripción del plan.
- planPrice: Precio del plan.
- planDuration: Duración del plan.
- planType: Tipo de plan.
- planStatus: Estado del plan.
- institutionalPlan: Plan institucional.
- class: Clase del plan.

Los atributos de la tabla Plan proporcionan información sobre los planes del sistema de geolocalización. Los atributos id, planName, planDescription, planPrice, planDuration, planType, planStatus y institutionalPlan proporcionan información básica sobre el plan.

2.4.1.2. Conclusión Modelo de Datos

Las tablas presentadas en este documento proporcionan un primer prototipo para el sistema de geolocalización HawkEye. Este prototipo da respuesta al primer prototipo de la aplicación y la web de HawkEye, y proporciona una base sólida para el desarrollo futuro del sistema.

A futuro, el sistema podría mejorarse de la siguiente manera:

- Añadir atributos adicionales a las tablas para proporcionar más información y funcionalidad. Por ejemplo:
 - **Tabla PersonSafe:
 - Fecha de nacimiento
 - Género

HawkEye Backend: Seguridad Personal en Tiempo Real

- Dirección
 - Imagen
- Tabla SafeZone:
 - Descripción
 - Tipo
- Tabla SafeZone:
 - Descripción
 - Tipo
 - Ubicación
 - Límites de la zona segura
- Tabla Plan
 - features: Características del plan
 - benefits: Beneficios del plan.
- Tablas en General
 - Fecha de Creación
 - Última Actualización
- Revisar las relaciones entre las tablas para garantizar que sean correctas y eficientes. Por ejemplo, la tabla Invitation podría tener una relación de uno a muchos con la tabla PersonSafe, en lugar de una relación de muchos a muchos.
- Implementar mecanismos de seguridad para proteger los datos almacenados en las tablas. Por ejemplo, las tablas podrían cifrarse para evitar el acceso no autorizado a los datos.

Estas recomendaciones proporcionarían información adicional que podría ser útil para el sistema de geolocalización, como la fecha de nacimiento, el género, la dirección y la imagen de la persona segura, o la descripción, el tipo y la ubicación de la zona segura.

En general, las tablas presentadas en este documento proporcionan una base sólida para el desarrollo futuro del sistema de geolocalización HawkEye. Con las mejoras adecuadas, el sistema podría proporcionar una herramienta valiosa para la seguridad y el bienestar de las personas.

2.4.2. MODELO DE ARQUITECTURA

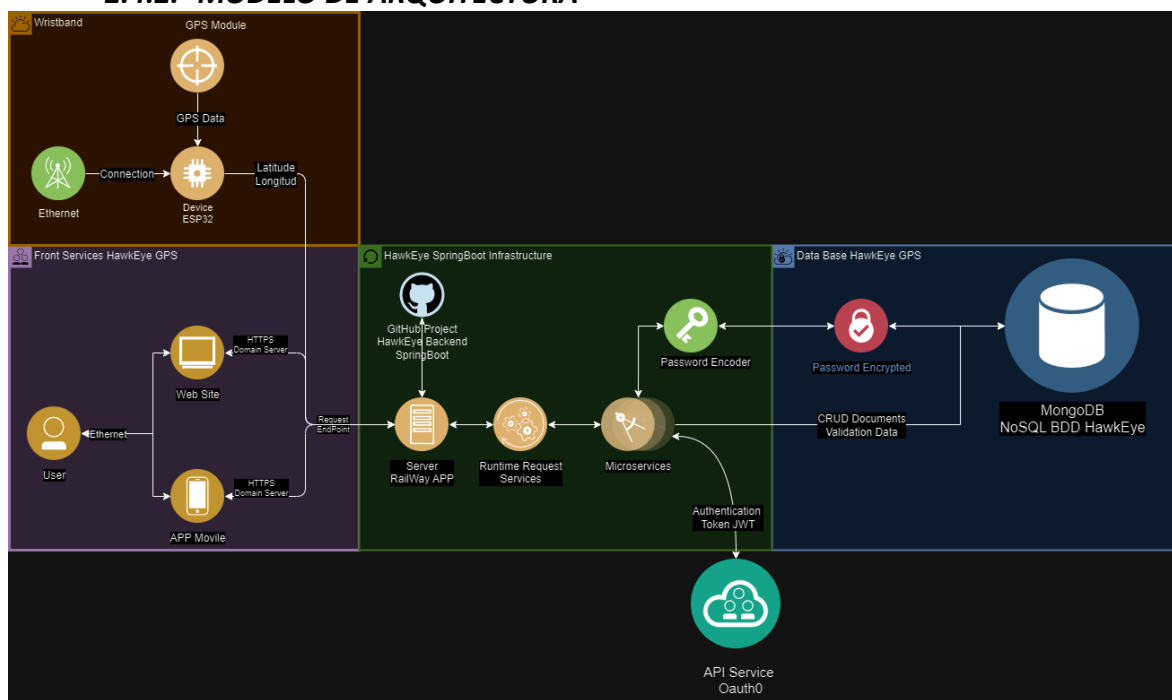


Figura 10 Arquitectura de Servicio HawkEye

2.4.2.1. Explicación Arquitectura de Servicio.

El modelo de arquitectura del sistema de geolocalización HawkEye se basa en una arquitectura cliente-servidor. El sistema consta de tres componentes principales:

- El dispositivo GPS: Este componente es un dispositivo *wearable* que lleva el usuario. El dispositivo GPS recopila datos de ubicación y los envía al servidor.
- El usuario: Este componente representa al usuario del sistema. El usuario puede acceder al sistema a través de una aplicación web o móvil.
- El servidor: Este componente alberga la aplicación backend del sistema. La aplicación backend se encarga de procesar los datos de ubicación recibidos del dispositivo GPS y de proporcionar la información al usuario.

El modelo de arquitectura se puede dividir en las siguientes capas:

- Capa de hardware: Esta capa consta del dispositivo GPS y del servidor.
- Capa de software: Esta capa consta de la aplicación backend y de la aplicación cliente.

2.4.2.1.1. Capa de hardware

El dispositivo GPS es un dispositivo *wearable* que lleva el usuario. El dispositivo GPS recopila datos de ubicación utilizando un receptor GPS. Los datos de ubicación se envían al servidor en forma de paquetes de datos.

HawkEye Backend: Seguridad Personal en Tiempo Real

El servidor es un servidor web que alberga la aplicación backend del sistema. La aplicación backend se ejecuta en un contenedor Docker. El servidor se comunica con el dispositivo GPS a través de una API REST.

2.4.2.1.2. Capa de software

La aplicación Backend es una aplicación web desarrollada en Spring Boot. La aplicación Backend se encarga de procesar los datos de ubicación recibidos del dispositivo GPS. La aplicación Backend también se encarga de proporcionar la información al usuario.

La aplicación cliente es una aplicación web o móvil que se utiliza para acceder al sistema. La aplicación cliente se comunica con la aplicación Backend a través de una API REST.

Explicación detallada

2.4.2.1.3. Zona amarilla: Dispositivo GPS

El dispositivo GPS es un componente esencial del sistema de geolocalización HawkEye. El dispositivo GPS recopila datos de ubicación utilizando un receptor GPS. Los datos de ubicación se envían al servidor en forma de paquetes de datos.

El dispositivo GPS debe cumplir con los siguientes requisitos:

- Debe ser pequeño y ligero para que sea cómodo de llevar.
- Debe tener una batería con una larga duración para que pueda utilizarse durante todo el día.
- Debe tener una buena recepción GPS para que pueda recopilar datos de ubicación precisos.

2.4.2.1.4. Zona morada o rosada: Usuario

El usuario es el componente que utiliza el sistema de geolocalización HawkEye. El usuario puede acceder al sistema a través de una aplicación web o móvil.

La aplicación cliente debe cumplir con los siguientes requisitos:

- Debe ser fácil de usar para que el usuario pueda acceder al sistema fácilmente.
- Debe proporcionar una interfaz gráfica intuitiva para que el usuario pueda visualizar la información fácilmente.
- Debe ser segura para proteger los datos del usuario.

2.4.2.1.5. Zona verde: Servidor

El servidor es un componente importante del sistema de geolocalización HawkEye. El servidor alberga la aplicación Backend del sistema. La aplicación Backend se encarga de procesar los datos de ubicación recibidos del dispositivo GPS y de proporcionar la información al usuario.

La aplicación Backend debe cumplir con los siguientes requisitos:

- Debe ser escalable para que pueda manejar un gran número de usuarios.

HawkEye Backend: Seguridad Personal en Tiempo Real

- Debe ser segura para proteger los datos del usuario.

2.4.2.1.6. Zona azul o celeste: Base de datos

La base de datos es un componente importante del sistema de geolocalización HawkEye. La base de datos almacena los datos de ubicación recopilados por el dispositivo GPS.

La base de datos debe cumplir con los siguientes requisitos:

- Debe ser eficiente para que pueda acceder a los datos rápidamente.
- Debe ser segura para proteger los datos del usuario.

2.4.2.1.7. Conclusión

El modelo de arquitectura del sistema de geolocalización HawkEye es un modelo eficaz que cumple con los requisitos del sistema. El modelo es escalable, seguro y fácil de usar.

2.4.3. MODELO DE RELACIÓN DE OBJETOS (ORM) EN HAWKEYE

El Modelo de Relación de Objetos (ORM) es un enfoque de programación que facilita la interacción entre sistemas orientados a objetos y bases de datos relacionales. En lugar de escribir consultas SQL directamente, el ORM utiliza objetos del lenguaje de programación para representar datos y operaciones en la base de datos, permitiendo una integración más intuitiva y eficiente.

2.4.3.1. Ventajas de ORM

- **Abstracción de Base de Datos:** Proporciona una capa de abstracción que simplifica las operaciones con la base de datos, permitiendo a los desarrolladores trabajar con objetos y métodos familiares.
- **Eficiencia en el Desarrollo:** Agiliza el proceso de desarrollo al eliminar la necesidad de escribir consultas SQL complejas, reduciendo el tiempo de implementación.
- **Portabilidad del Código:** Facilita la portabilidad del código entre diferentes sistemas de gestión de bases de datos, ya que las operaciones específicas de la base de datos se gestionan a través del ORM.
- **Reducción de Errores:** Minimiza la posibilidad de errores relacionados con la sintaxis SQL, ya que las operaciones se realizan mediante métodos de lenguajes de programación.

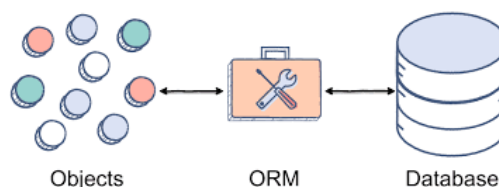


Figura 11 Relación Objeto, ORM y Base de Datos

2.4.3.2. Elección de ORM en HawkEye

En el desarrollo de HawkEye, se optó por la implementación del Modelo de Relación de Objetos (ORM) para gestionar las interacciones con la base de datos MongoDB. La elección de ORM se fundamenta en las siguientes consideraciones:

- **Eficiencia y Desarrollo Ágil:** Con la presencia de microservicios en HawkEye, la eficiencia en el desarrollo es crucial. ORM permite una implementación rápida y eficiente de operaciones de base de datos.

HawkEye Backend: Seguridad Personal en Tiempo Real

- Adaptabilidad y Escalabilidad: La arquitectura de microservicios de HawkEye demanda una solución que se adapte fácilmente a cambios y crecimiento. ORM proporciona la flexibilidad necesaria.
- Compatibilidad con MongoDB: Dado que MongoDB es una base de datos NoSQL, la elección de ORM es especialmente relevante para asegurar la coherencia en la manipulación de datos orientados a documentos.

2.4.3.3. Modelo Monolítico y Limitaciones

Contrastando con el enfoque de microservicios utilizado en HawkEye, el modelo monolítico es un diseño más tradicional donde toda la aplicación se desarrolla y despliega como una única unidad. Aunque este modelo tiene sus ventajas, como la simplicidad inicial, presenta limitaciones notables, especialmente en la manipulación de consultas y operaciones complejas debido a su naturaleza centralizada.

En resumen, la elección consciente de utilizar el Modelo de Relación de Objetos (ORM) en HawkEye fortalece la eficiencia y adaptabilidad de la aplicación, permitiendo una gestión ágil y eficiente de la base de datos MongoDB en el contexto de una arquitectura de microservicios.

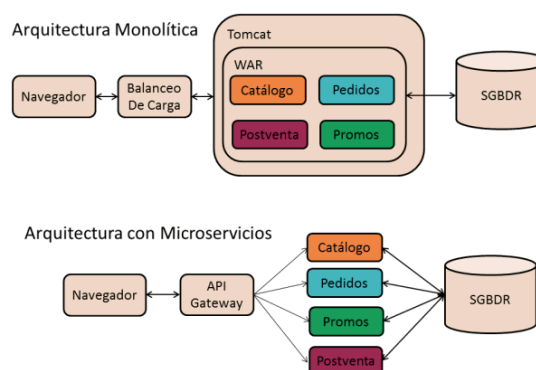


Figura 12 Arquitectura Monolítica Vs ORM

3. CAPÍTULO 3: PROPUESTA DE SOLUCION

La esencia del sistema HawkEye reside en su robusta arquitectura de backend, donde la interacción entre distintos componentes se lleva a cabo para garantizar la supervisión eficiente de individuos vulnerables. Esta arquitectura, basada en una API REST y una base de datos NoSQL, proporciona la columna vertebral del sistema, permitiendo la comunicación fluida, el almacenamiento seguro y el procesamiento de datos. Arquitectura de Software HawkEye La arquitectura de software HawkEye se basa en un enfoque de nube y consta de los siguientes componentes clave:

- **Framework Spring y Utilización de Spring Boot: Arquitectura Ágil y Modular**

La adopción de Spring Boot, dentro del ecosistema del framework Spring, posibilita una eficiente creación de aplicaciones Java con una mínima carga de esfuerzo. Esto propicia la construcción de componentes web y móviles con rapidez y versatilidad, asegurando un modularidad que favorece el desarrollo iterativo y escalable de HawkEye.

- **Spring Data MongoDB: Gestión Eficaz de Datos NoSQL**

Dentro de este contexto, la integración de Spring Data MongoDB provee una capa de abstracción que simplifica la interacción con la base de datos NoSQL. Esto permite la manipulación de repositorios y consultas de manera eficaz, posibilitando así un almacenamiento y acceso óptimos a los datos generados por el sistema.

- **RESTful API: Pilar Fundacional de la Comunicación**

La adopción de una arquitectura RESTful se erige como un componente crítico en la comunicación fluida entre los distintos elementos de HawkEye. Los controladores en Spring (Spring MVC) gestionan las solicitudes HTTP, permitiendo una interacción coherente y predecible entre la aplicación móvil y los dispositivos de rastreo, asegurando así una comunicación segura y escalable.

- **Técnicas Avanzadas: Inyección de Dependencias y Mapeo Objeto-Documento**

La inyección de dependencias, un patrón esencial en el diseño de software es empleada estratégicamente para administrar las relaciones y dependencias entre los distintos componentes de HawkEye. Por otro lado, el mapeo Objeto-Documento, utilizando anotaciones como @Document, garantiza la sincronía entre los objetos Java y la representación de documentos en MongoDB, asegurando una coherencia y consistencia en la gestión de datos.

- **Optimización del Almacenamiento: MongoDB y Spring Data JPA**

La elección de MongoDB, como base de datos NoSQL, aporta una flexibilidad y escalabilidad sustanciales para el almacenamiento de datos. En paralelo, Spring Data JPA ofrece herramientas adicionales para interactuar con estos datos, simplificando su acceso y manipulación de manera eficiente.

- **Herramientas para el Desarrollo y Gestión: Postman y Notion**

La inclusión de herramientas como Postman desempeña un papel fundamental en la validación y verificación de la funcionalidad de las API desarrolladas, permitiendo así una fase de pruebas exhaustivas y precisas. Paralelamente, Notion emerge como un recurso valioso para la gestión integral de proyectos, facilitando la colaboración entre equipos y asegurando una planificación y seguimiento efectivos.

3.1. **TABLAS DE HISTORIAS DE USUARIO**

Tabla 1 Historia de Usuario Asignación de Monitoreo

HU00		Asignación de monitoreo.			
Usuario	Tutor	Criterio	Usabilidad	Dependencias	Registro
Prioridad	Alto	Riesgo	Alto	Estimación	5
Responsable	XX				
Descripción	Como usuario, espero que la asignación de monitoreo sea rápida e intuitiva, como también que me dé la sensación de seguridad de asignar a mi familiar a la aplicación.				
Pruebas de Aceptación	• La interfaz debe ser intuitiva, segura y detallada, para realizar el procedimiento de la mejor manera posible. • Durante la asignación la aplicación debe mostrar la configuración del perfil del sujeto al que se cuidará. • Luego de la asignación debe mostrarse un tutorial breve de uso para conocer todas las funcionalidades y procedimientos para casos de riesgo.				
Observaciones	En caso de asignación incorrecta, puede generar confusiones e inseguridad, debe estar bien implementada.				

Tabla 2 Historia de Usuario Monitoreo de Familiar

HU01		Monitoreo de familiar.			
Usuario	Tutor	Criterio	Usabilidad	Dependencias	HU00
Prioridad	Alto	Riesgo	Medio	Estimación	4
Responsable	XX				
Descripción	Como usuario, quiero poder monitorear la ubicación de un familiar a través de GPS, para tener una visibilidad constante de su ubicación y asegurarme de que se encuentra seguro.				
Pruebas de Aceptación	- La aplicación debe permitirme seleccionar a que familiar quiero monitorear a través de GPS. - La aplicación debe mostrar la ubicación del familiar en un mapa en tiempo real. - La aplicación debe permitirme configurar alertas personalizadas para notificarme si el familiar sale de un área determinada o si se mueve a una velocidad mayor a un límite establecido. - La aplicación debe permitirme ver el historial de recorrido de mi familiar para saber dónde estuvo para determinar si no tuvo ningún recorrido anormal.				
Observaciones	Velar por una buena usabilidad del funcionamiento.				

Tabla 3 Historia de Usuario Fijado de Perímetro

HU02		Fijado de Perímetro			
Usuario	Tutor	Criterio	Exactitud	Dependencias	Asignación
Prioridad	Alto	Riesgo	Medio	Estimación	7
Responsable	XX				
Descripción	Como usuario, quiero ser capaz de fijar un perímetro seguro para mi hijo/a, para que reciba alertas si sale del mismo.				
Pruebas de Aceptación	- Puedo establecer fácilmente un perímetro seguro en un mapa o con coordenadas geográficas. - Puedo ver claramente el perímetro seguro en la aplicación y puedo ajustarlo si es necesario. - Recibo una alerta inmediata cuando mi hijo/a sale del perímetro seguro. - Puedo cancelar la alerta fácilmente si no es necesaria.				
Observaciones					

Tabla 4 Historia de Usuario Duración de la Batería

HU03		Duración de la batería			
Usuario	Tutor	Criterio	Durabilidad	Dependencias	Equipo Electrónico
Prioridad	Alto	Riesgo	Bajo	Estimación	8
Responsable	XX				
Descripción	Como usuario, quiero poder utilizar dispositivos con batería lo suficientemente duradera como para no tener que cargarlos a menudo.				
Pruebas de Aceptación	- El dispositivo tiene una duración de batería de al menos X horas/días - El usuario puede usar el dispositivo sin tener que cargarlo con frecuencia - El dispositivo tiene un indicador de batería para que el usuario pueda saber cuándo cargarlo				
Observaciones	Determinar una forma de que el usuario sepa cuando debe cargar el equipo, para mantener una constancia y cuidar su batería.				

Tabla 5 Historia de Usuario Botón SOS

HU04		Botón SOS			
Usuario	Tutor	Criterio	Exactitud	Dependencias	Integración con sistema de geolocalización y HU05.
Prioridad	Alto	Riesgo	Medio	Estimación	5
Responsable	XX				
Descripción	Como usuario, quiero contar con un botón SOS incorporado al dispositivo para que sea fácilmente accesible por la persona en casos de emergencias y/ o desapariciones inmediatas.				
Pruebas de Aceptación	- El botón SOS debe ser fácilmente accesible y visible en el dispositivo. - Al presionar el botón SOS, el dispositivo debe enviar una notificación de alerta a los contactos de emergencia previamente establecidos. - El usuario debe poder personalizar la lista de contactos de emergencia. - La alerta debe incluir la ubicación actual del usuario en el momento de la activación del botón SOS.				
Observaciones					

Tabla 6 Historia de Usuario Vinculación de Contactos de Emergencia

HU05		Vinculación de contactos de emergencia			
Usuario	Tutor	Criterio	Funcionalidad	Dependencias	
Prioridad	Alto	Riesgo	Bajo	Estimación	7
Responsable	XX				
Descripción	Como usuario, quiero tener la capacidad de vincular a contactos de emergencia a mi pulsera GPS, para que ellos reciban notificaciones si hay algún problema.				
Pruebas de Aceptación	- Puedo agregar fácilmente a uno o varios contactos de emergencia en la pulsera GPS. - Los contactos de emergencia pueden recibir notificaciones inmediatas si hay algún problema con la pulsera GPS. - Puedo eliminar o editar los contactos de emergencia vinculados en cualquier momento.				
Observaciones	Tener un mínimo de contactos de emergencia adicional al tutor.				

Tabla 7 Historia de Usuario Ver Ubicación y estado de la Persona desde Cualquier Dispositivo

HU06		Ver ubicación y estado de la persona desde cualquier dispositivo			
Usuario	Tutor	Criterio	Funcionalidad	Dependencias	
Prioridad	Alto	Riesgo	Medio	Estimación	8
Responsable	XX				
Descripción	Como usuario, quiero tener la capacidad de ver la ubicación y el estado de la persona desde cualquier dispositivo conectado a internet (computador, Tablet, teléfono)				
Pruebas de Aceptación	- Puedo ver la ubicación y el estado de la persona desde cualquier dispositivo conectado a internet. - La ubicación y el estado de la persona se actualizan en tiempo real. - La información de ubicación y estado es precisa y confiable. - Puedo acceder a la información de ubicación y estado con una cuenta de usuario segura y personalizada.				
Observaciones	Adaptabilidad de Interfaz				

Tabla 8 Historia de Usuario Servicio al Cliente

HU07		Servicio al Cliente			
Usuario	Tutor	Criterio	Servicio	Dependencias	
Prioridad	Alto	Riesgo	Bajo	Estimación	3
Responsable	XX				
Descripción	Como usuario, quiero contar con un servicio al cliente profesional y eficiente para ayudarme con todas mis dudas relacionadas con el producto.				
Pruebas de Aceptación	- El servicio al cliente debe estar disponible en diferentes canales de comunicación (teléfono, correo electrónico, chat en línea). - El servicio al cliente debe responder dentro de un plazo razonable de tiempo. - El servicio al cliente debe estar capacitado para responder todas las preguntas relacionadas con el producto.				
Observaciones	Añadir apartado de servicio al cliente.				

Tabla 9 Historia de Usuario notificación Salida y Entrada de Áreas Específicas

HU08		Notificaciones Salida y Entrada de Áreas Específicas			
Usuario	Tutor	Criterio	Exactitud	Dependencias	HU01, HU02, HU04
Prioridad	Alto	Riesgo	Bajo	Estimación	6
Responsable	XX				
Descripción	Recibir notificaciones cuando mi hij@/familiar/persona monitoreada entre o salga de un área específica.				
Pruebas de Aceptación	Recibo notificaciones precisas en tiempo real cuando la persona monitoreada entra o sale de un área específica.				
Observaciones					

Tabla 10 10 Historia de Usuario Sonidos de Alertas Personalizados

HU09		Sonidos de Alertas Personalizados			
Usuario	Tutor	Criterio	Exactitud	Dependencias	HU01, HU02, HU04
Prioridad	Medio	Riesgo	Bajo	Estimación	5
Responsable					
Descripción	Como usuario, quiero poder configurar diferentes tipos de alertas auditivas o visuales dependiendo del nivel de urgencia (por ejemplo: cuando el dispositivo es accionado o cuando se sale del perímetro seguro).				
Pruebas de Aceptación	• Puedo configurar diferentes tipos de alertas auditivas o visuales. • Puedo establecer diferentes niveles de urgencia para las alertas. • Puedo configurar alertas específicas para cuando el dispositivo es accionado. • Puedo configurar alertas específicas para cuando se sale del perímetro seguro. • Puedo cambiar las configuraciones de alerta en cualquier momento.				
Observaciones	Adaptaciones nuevas a la pulsera.				

Tabla 11 Historia de Usuario Interfaz Intuitiva y Sencilla

HU10		Interfaz intuitiva y sencilla			
Usuario	Tutor	Criterio	Interfaz	Dependencias	
Prioridad	Alto	Riesgo	Bajo- Medio	Estimación	5
Responsable					
Descripción	Como usuario, quiero una interfaz intuitiva y sencilla que me permita ver la ubicación exacta y el estado de la persona en tiempo real desde mi dispositivo móvil o computador.				
Pruebas de Aceptación	- Puedo visualizar la ubicación de la persona monitoreada en tiempo real en un mapa o visualizador en línea. - Puedo ver el estado de la persona monitoreada, como la batería, la velocidad, la dirección, etc. - La interfaz es fácil de usar y no requiere un conocimiento técnico avanzado. - La actualización de la ubicación y el estado es en tiempo real y precisa.				
Observaciones					

Tabla 12 Ajuste de Velocidad Máxima

HU11		Ajuste de Velocidad Máxima			
Usuario	Tutor	Criterio	Usabilidad	Dependencias	HU00, HU01, HU06
Prioridad	Medio	Riesgo	Bajo	Estimación	3
Responsable	XX				
Descripción	Como usuario, quiero ser capaz de configurar los parámetros del dispositivo para ajustarlo a mis necesidades particulares (por ejemplo: velocidad máxima permitida).				
Pruebas de Aceptación	- El usuario puede ajustar la velocidad máxima permitida para el dispositivo y guardar los cambios. - Los cambios realizados en la configuración del dispositivo se aplican correctamente en tiempo real. - La funcionalidad de configuración es fácil de usar y entender para el usuario.				
Observaciones					

Tabla 13 Historia de Usuario Programación del Horario de Funcionamiento

HU12			Programación del horario de funcionamiento.			
Usuario	Tutor		Criterio	Funcionamiento	Dependencias	HU00, HU03
Prioridad	Medio		Riesgo	Medio	Estimación	3
Responsable			XX			
Descripción			Como usuario, quiero ser capaz de programar la hora exacta en que el dispositivo se activa y desactiva para monitorear a la persona.			
Pruebas de Aceptación			- El usuario puede acceder a la función de programación de horario desde la aplicación. - La aplicación muestra claramente el horario programado de inicio y fin del monitoreo en una interfaz intuitiva y fácil de usar. - La aplicación activa y desactiva automáticamente la función de monitoreo según el horario programado sin errores.			
Observaciones						

Tabla 14 Historia de Usuario Resistencia a Líquidos

HU13			Resistencia a líquidos.			
Usuario	Tutor		Criterio	Funcionalidad	Dependencias	Equipo Electrónico
Prioridad	Medio		Riesgo	Alto	Estimación	5
Responsable			XX			
Descripción			Como usuario, quiero contar con un diseño resistente al agua para evitar situaciones en las que el dispositivo sea destruido por algún líquido.			
Pruebas de Aceptación			- La documentación del dispositivo incluye información detallada sobre su resistencia al agua, incluyendo el nivel de protección, la duración máxima de inmersión y otros factores relevantes. - La resistencia al agua es parte del diseño general del dispositivo y no afecta su funcionalidad normal.			

HawkEye Backend: Seguridad Personal en Tiempo Real

		El diseño resistente al agua cumple con los estándares relevantes de la industria para garantizar la calidad y la confiabilidad.
Observaciones		Material de la pulsera

Tabla 15 Historia de Usuario Batería Recargable

HU14		Batería Recargable			
Usuario	Tutor	Criterio	Funcionalidad	Dependencias	HU03
Prioridad	Medio	Riesgo	Medio	Estimación	5
Responsable	XX				
Descripción	Como usuario, quiero contar con un sistema de batería recargable para evitar cambiarla frecuentemente.				
Pruebas de Aceptación	- La documentación del dispositivo incluye información detallada sobre la duración de la batería y el tiempo de carga, incluyendo recomendaciones para maximizar la vida útil de la batería. - La aplicación incluye una función que permite al usuario saber cuánta carga queda en la batería y cuánto tiempo se espera que dure antes de necesitar una nueva carga. - La aplicación notifica al usuario cuando la batería está baja y necesita ser cargada. - El sistema de batería recargable no aumenta significativamente el tamaño, peso o costo del dispositivo.				
Observaciones					

Tabla 16 Historia de Usuario Autorización de Ingreso

HU15		Autorización de ingreso			
Usuario	Tutor	Criterio	Seguridad	Dependencias	
Prioridad	Alto	Riesgo	Bajo	Estimación	4
Responsable	Ciberseguridad				
Descripción	Como usuario, quiero ser notificado cuando un contacto autorizado intente ingresar a mi cuenta para acceder a los datos y ubicación de mi hij@familiar/persona monitoreada con la pulsera GPS.				
Pruebas de Aceptación	- Los usuarios autorizados pueden configurar sus preferencias de notificación y decidir qué tipo de alertas desean recibir, como notificaciones <i>push</i>, correos electrónicos o mensajes de texto. - La aplicación proporciona el historial de inicios de sesión con detalle de ubicación y contacto.				
Observaciones					

Tabla 17 Historia de Usuario Variedad de Productos

HU16		Variedad de producto.			
Usuario	Tutor	Criterio	Variabilidad	Dependencias	Equipo Electrónico
Prioridad	Medio	Riesgo	Bajo	Estimación	2
Responsable	XX				
Descripción	Como usuario, me gustaría contar con distintas opciones disponibles para elegir entre varias pulseras GPS y otros dispositivos comercialmente disponibles.				
Pruebas de Aceptación	- Si el usuario no posee una pulsera, la aplicación lo redirigirá a la página web con todas las opciones de pulsera y planes disponibles. - Como servicio, se implementarán diversos diseños de pulsera para adaptar gustos y comodidad del usuario.				
Observaciones					

Tabla 18 Historia de Usuario Procedimiento SOS

HU16		Procedimiento SOS			
Usuario	Tutor	Criterio	Disponibilidad	Dependencias	HU01, HU04, HU05
Prioridad	Alto	Riesgo	Alto	Estimación	6
Responsable	XX				
Descripción	Como usuario, quiero disponer un sistema que envíe automáticamente un mensaje de texto a los contactos autorizados cuando se active el botón SOS en la pulsera GPS.				
Pruebas de Aceptación	- El mensaje de texto enviado a los contactos autorizados es claro, fácil de entender y contiene la información necesaria, como la ubicación actual del usuario y un enlace al mapa. - La documentación del producto proporciona información clara y fácil de entender sobre cómo utilizar el botón SOS y el sistema de alerta de la pulsera GPS, para que el usuario pueda comprender rápidamente cómo utilizar el sistema en caso de emergencia. - El sistema cumple con las regulaciones y normativas aplicables en cuanto a la privacidad y seguridad de los datos del usuario.				
Observaciones					

Tabla 19 Historia de Usuario Autorización de Cambio en Configuración

HU17		Autorización de cambios en configuración.			
Usuario	Tutor	Criterio	Integridad	Dependencias	HU02
Prioridad	Bajo	Riesgo	Medio	Estimación	3
Responsable	XX				
Descripción	Como usuario, quiero ser notificado si hay cambios en el perímetro definido como lugar seguro para mi hij@/familiar/persona monitoreada con la pulsera GPS.				
Pruebas de Aceptación	- La notificación de cambios en el perímetro seguro se debe enviar de manera clara y concisa a través de la aplicación móvil, correo electrónico o mensaje de texto según preferencias de usuario. - La notificación debe incluir información sobre el nuevo perímetro seguro y la fecha y hora en que se hizo el cambio. - La notificación debe ser recibida en tiempo real, sin demoras significativas, para que pueda tomar medidas inmediatas si es necesario. - Si no se especifica un perímetro seguro anteriormente, la aplicación debe guiar al usuario por el proceso de configuración y notificarle cuando el perímetro seguro se haya establecido.				
Observaciones					

Tabla 20 Historia de Usuario Configuración de Notificaciones de Estado

HU18		Configuración de notificaciones de estado.			
Usuario	Tutor	Criterio	Configuración	Dependencias	HU05
Prioridad	Medio	Riesgo	Medio	Estimación	3
Responsable	XX				
Descripción	Como usuario, me gustaría tener la posibilidad de escoger entre distintos formatos para recibir información acerca del paradero y ubicación exacta de mi hij@/familiar/persona monitoreada con la pulsera GPS (SMS, texto, correo electrónico, mensaje de voz).				
Pruebas de Aceptación	- La aplicación debe permitir al usuario elegir entre diferentes formatos de notificación, como SMS, texto, correo electrónico o mensaje de voz, para recibir información sobre el paradero y ubicación exacta de la persona monitoreada. - La notificación en el formato elegido debe incluir la información necesaria, como la ubicación exacta, la hora y fecha de la actualización, y cualquier otro detalle relevante para el seguimiento de la persona monitoreada. - La notificación debe ser recibida en tiempo real, sin demoras significativas, para que pueda tomar medidas inmediatas si es necesario. - El usuario debe poder seleccionar fácilmente su preferencia de formato desde la configuración de la aplicación, sin dificultades ni pasos complicados.				
Observaciones					

3.2. CASOS DE USO EXTENDIDOS

3.2.1. Perfil - Individuo

3.2.1.1. Crear perfil

Actores: Individuo

Descripción:

Permite al individuo crear un nuevo perfil en la aplicación móvil o la aplicación web. El individuo proporciona la información necesaria, como nombre, fecha de nacimiento, dirección, número de teléfono y otros detalles relevantes. Una vez creado el perfil, el individuo puede asociar su pulsera GPS y permitir que el tutor u otras personas autorizadas monitoreen su ubicación y estado.

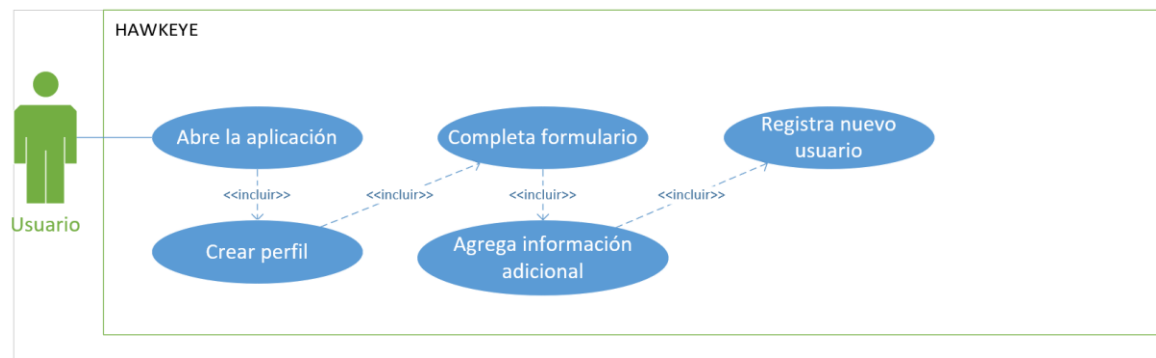


Figura 13 Caso de uso crear perfil

3.2.1.2. Editar perfil

Actores: Individuo

Descripción: Permite al individuo editar la información en su perfil existente. El individuo puede modificar detalles como dirección, número de teléfono, información de contacto de emergencia u otros campos relevantes. Esta funcionalidad brinda flexibilidad para mantener actualizada la información personal en caso de cambios.

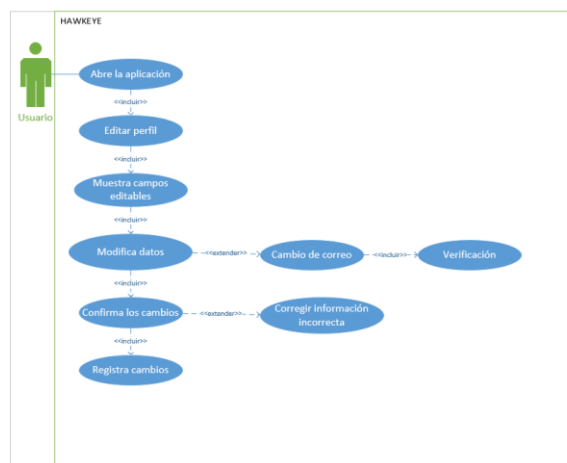


Figura 14 Caso de uso editar perfil

3.2.1.3. Eliminar perfil

Actores: Individuo

Descripción: Permite al individuo eliminar su perfil de la aplicación móvil o la aplicación web. Al seleccionar esta opción, se solicita una confirmación para asegurar que el individuo realmente desea eliminar su perfil y los datos asociados. Una vez confirmada la eliminación, se eliminan todos los datos relacionados con el individuo y la pulsera GPS asociada.

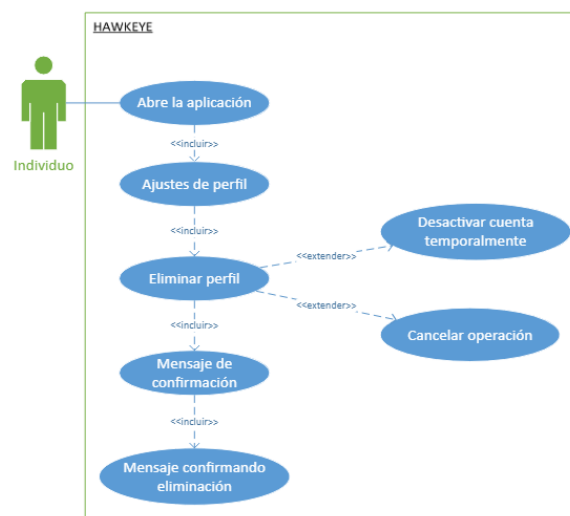


Figura 15 Caso de uso eliminar perfil

3.2.2. Mapa

3.2.2.1. Ver Mapa

Actores: Tutor, Individuo

Descripción: Permite al tutor e individuo ver un mapa interactivo en la aplicación móvil o la aplicación web. El mapa muestra la ubicación actual del individuo, así como otros elementos relevantes, como zonas seguras, puntos de interés y pulseras GPS asociadas. El tutor e individuo pueden explorar el mapa, hacer *zoom*, desplazarse y obtener información visual de la ubicación.

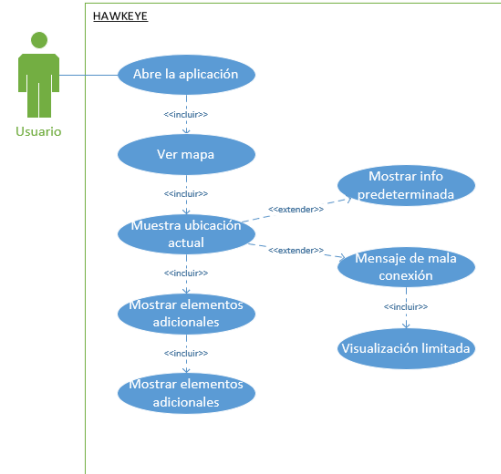


Figura 16 Caso de uso ver mapa

3.2.2.2. Designar Vallado Virtual

Actores: Tutor

Descripción: Permite al tutor establecer un vallado virtual en el mapa. El tutor define un área segura o un perímetro permitido alrededor de la ubicación del individuo utilizando una interfaz gráfica. El sistema registra los límites del vallado virtual para aplicar restricciones de movimiento y enviar alertas si el individuo sale de la zona permitida.

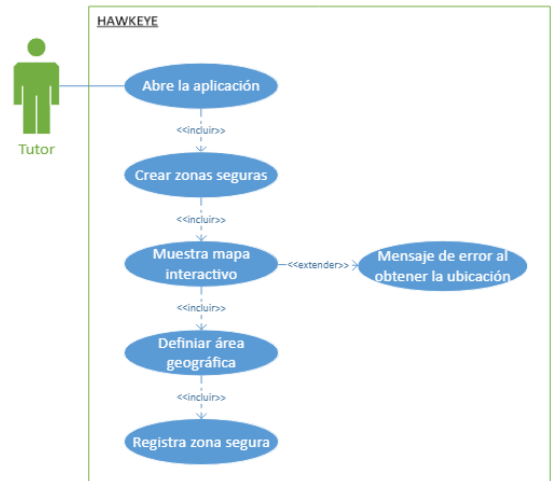


Figura 17 Caso de uso designar vallado virtual

Crear Zonas Seguras

Actores: Tutor

Descripción: Permite al tutor crear zonas seguras en el mapa. El tutor define áreas geográficas específicas, como la casa, la escuela o el trabajo, y las marca como zonas seguras. Estas zonas se utilizan para monitorear la ubicación del individuo y generar alertas si el individuo entra o sale de las zonas seguras designadas.

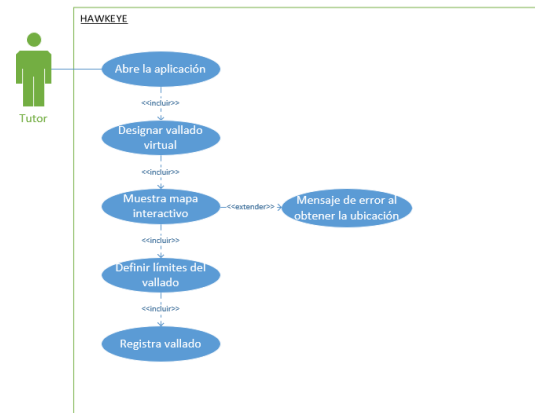


Figura 18 Caso de uso zonas seguras

3.2.2.3. Ver Historial de Recorrido

Actores: Tutor

Descripción: Permite al tutor ver el historial de recorrido del individuo en el mapa. El tutor puede visualizar las rutas y ubicaciones previas del individuo a lo largo del tiempo. Esto proporciona un registro de los movimientos y patrones de desplazamiento del individuo, lo que puede ser útil para la planificación y el seguimiento.

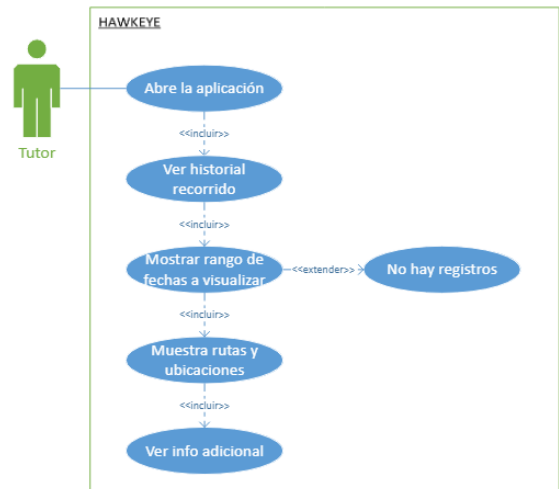


Figura 19 Caso de uso ver historial de recorrido

3.2.3. Pulsera

3.2.3.1. Registrar pulsera

Actores: Tutor

Descripción: Permite al tutor registrar una nueva pulsera en la aplicación móvil o la aplicación web. El tutor ingresa la información única de la pulsera, como la dirección MAC o un identificador único, para vincularla con el perfil correspondiente. Una vez registrada, la pulsera estará lista para ser utilizada y conectada al sistema

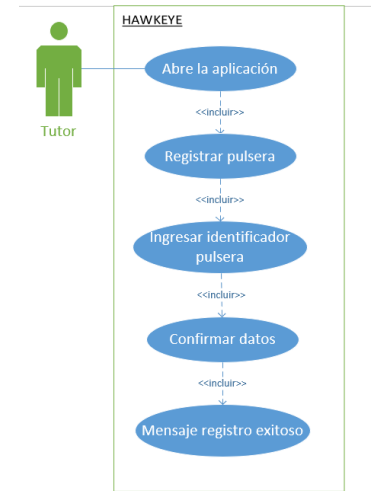


Figura 20 Caso de uso registrar pulsera

3.2.4. Alertas

3.2.4.1. Ver Historial de Alertas

Actores: Tutor

Descripción: Permite al tutor ver el historial de alertas generadas por la pulsera en la aplicación móvil o la aplicación web. El historial de alertas muestra las alertas anteriores registradas, proporcionando detalles como el tipo de alerta, la fecha y la hora, y la ubicación asociada. Esto permite al tutor revisar el registro de eventos y evaluar la seguridad y el bienestar del individuo.

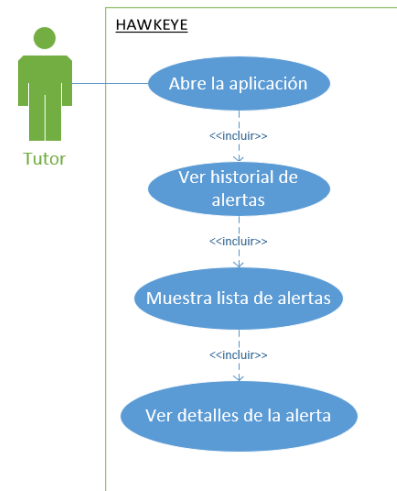


Figura 21 Caso de uso ver historial de alertas

3.2.4.2. Ejecutar una Alerta

Actores: Pulsera, Tutor

Descripción: Permite a la pulsera generar una alerta en situaciones de emergencia o eventos predefinidos. La pulsera detecta una situación anormal o una acción específica (como presionar un botón de pánico) y envía una señal de alerta al sistema. El sistema identifica la alerta y notifica al tutor correspondiente para que pueda tomar las medidas adecuadas.

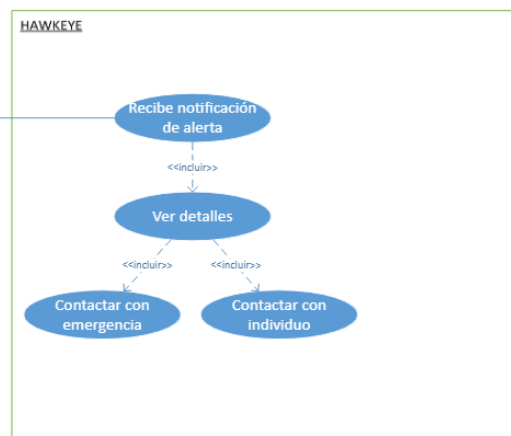


Figura 22 Caso de uso ejecutar una alerta

3.2.4.3. Recibir Notificación

Actores: Tutor

Descripción: Permite al tutor recibir notificaciones en tiempo real en la aplicación móvil o mediante otros medios, como mensajes de texto o correos electrónicos, cuando se genera una alerta. El tutor recibe una notificación que indica el tipo de alerta, la ubicación y otra información relevante. Esto asegura que el tutor esté informado rápidamente sobre situaciones críticas y pueda tomar las acciones necesarias.

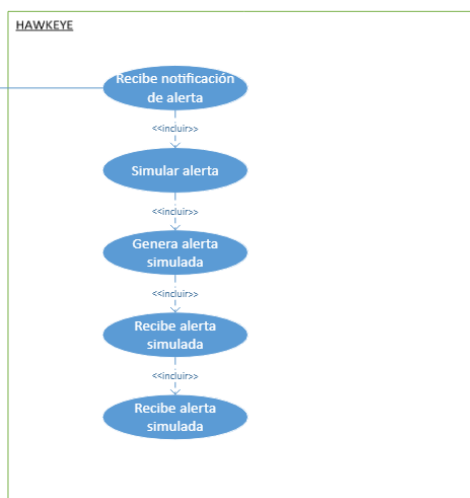


Figura 23 Caso de uso recibir notificación

3.2.4.4. Simular Alerta

Actores: Tutor

Descripción: Permite al tutor simular una alerta en la aplicación móvil o la aplicación web para realizar pruebas y verificar el funcionamiento del sistema. Esta función de simulación permite al tutor evaluar cómo se manejan las alertas, cómo se notifican y cómo se registran en el historial de alertas. Es útil para garantizar que el sistema esté configurado correctamente y que el flujo de alertas sea efectivo.

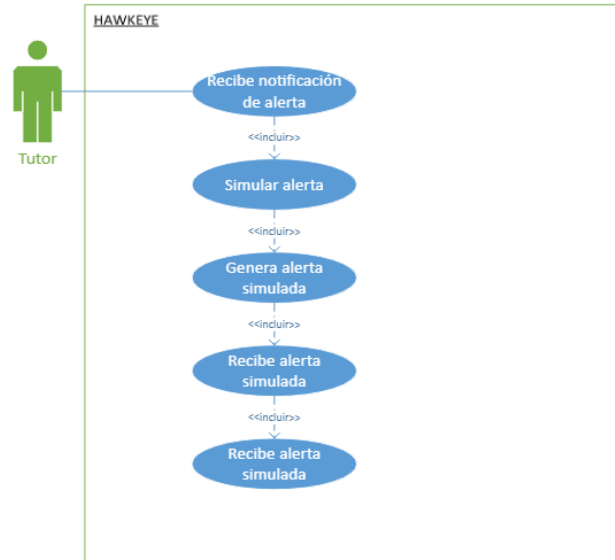


Figura 24 Caso de uso simular alerta

3.3. INFRAESTRUCTURA DE SERVICIO HAWKEYE

3.3.1. ¿Qué es la infraestructura?

La infraestructura en el desarrollo de software comprende un conjunto de herramientas, plataformas, entornos y tecnologías fundamentales para desarrollar, implementar y ejecutar el backend de una aplicación. En Hawkeye, esta capa representa la base esencial para garantizar el funcionamiento eficiente y efectivo de la aplicación.

3.3.2. ¿Cuál utilizamos?

En Hawkeye, nuestra infraestructura se apoya en el uso de Spring Boot junto al patrón arquitectónico Modelo-Vista-Controlador (MVC). Esta elección se fundamenta en su capacidad para proporcionar una estructura organizativa sólida y escalable tanto para el desarrollo del backend de nuestra aplicación móvil como para la versión web.

3.3.3. ¿Porque la utilizamos?

- **Eficiencia en el desarrollo:** Spring Boot, como *framework*, agiliza el proceso de desarrollo al ofrecer configuraciones predeterminadas inteligentes, reduciendo la necesidad de configuraciones manuales. Esto permite focalizar nuestros esfuerzos en la lógica de negocio y las funcionalidades específicas de Hawkeye, acelerando así nuestros ciclos de desarrollo.
- **Separación de responsabilidades:** El patrón Modelo-Vista-Controlador (MVC) nos brinda una clara separación entre la lógica de negocios (Modelo), la presentación de información (Vista) y el control de las interacciones del usuario (Controlador). Esta separación mejora la modularidad y facilita la colaboración entre equipos, además de simplificar el mantenimiento y la escalabilidad conforme la aplicación crece.
- **Escalabilidad y mantenimiento:** La combinación de Spring Boot y el patrón MVC nos permite desarrollar un backend altamente escalable, adaptable y de fácil mantenimiento. La arquitectura modular nos posibilita agregar nuevas funcionalidades o realizar cambios en componentes específicos sin afectar otras partes de la aplicación.
- **Conformidad con las mejores prácticas:** Spring Boot, respaldado por la comunidad de Spring, promueve las mejores prácticas en desarrollo, seguridad y rendimiento. Esto garantiza un alto nivel de calidad y robustez en nuestro backend.

3.3.4. Conclusiones

Elegimos Spring Boot con el patrón MVC en Hawkeye porque nos proporciona una infraestructura sólida, ágil y bien organizada para desarrollar y mantener el *backend* de nuestra aplicación móvil. Esto nos permite concentrarnos en brindar una experiencia óptima y segura a nuestros usuarios, asegurando un seguimiento en tiempo real y confiable mediante el uso de la pulsera proporcionada.

IMÁGENES

En esta imagen se demuestra el patrón de diseño MVC siendo utilizado

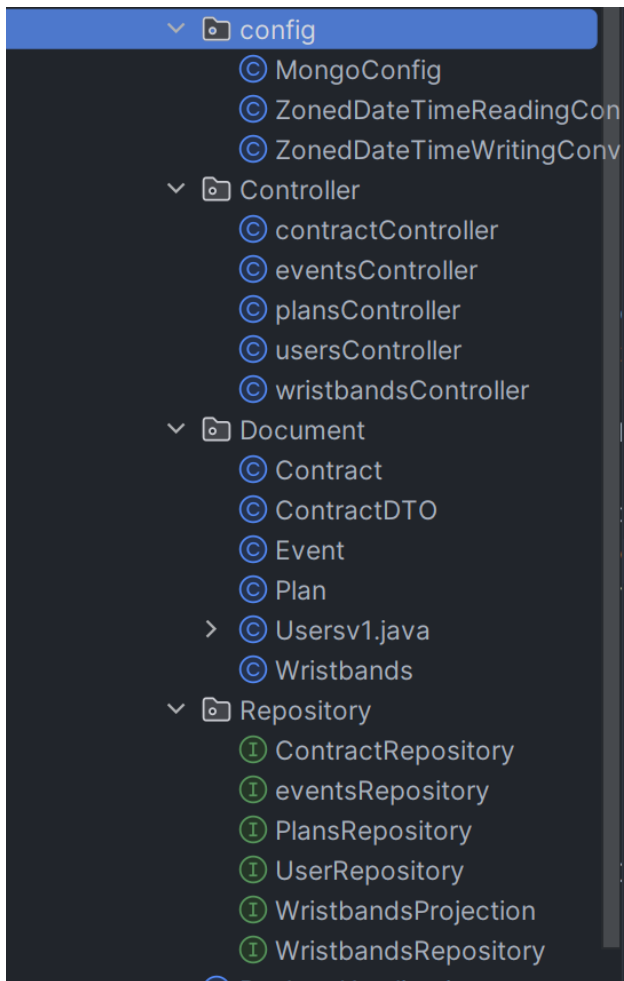


Figura 25 Modelo Vista Controlador HawkEye

HawkEye Backend: Seguridad Personal en Tiempo Real

Esta imagen Demuestra El MVC de solamente Users con sus clases anidadas el cual está organizado de tal manera que sea fácil a la lectura

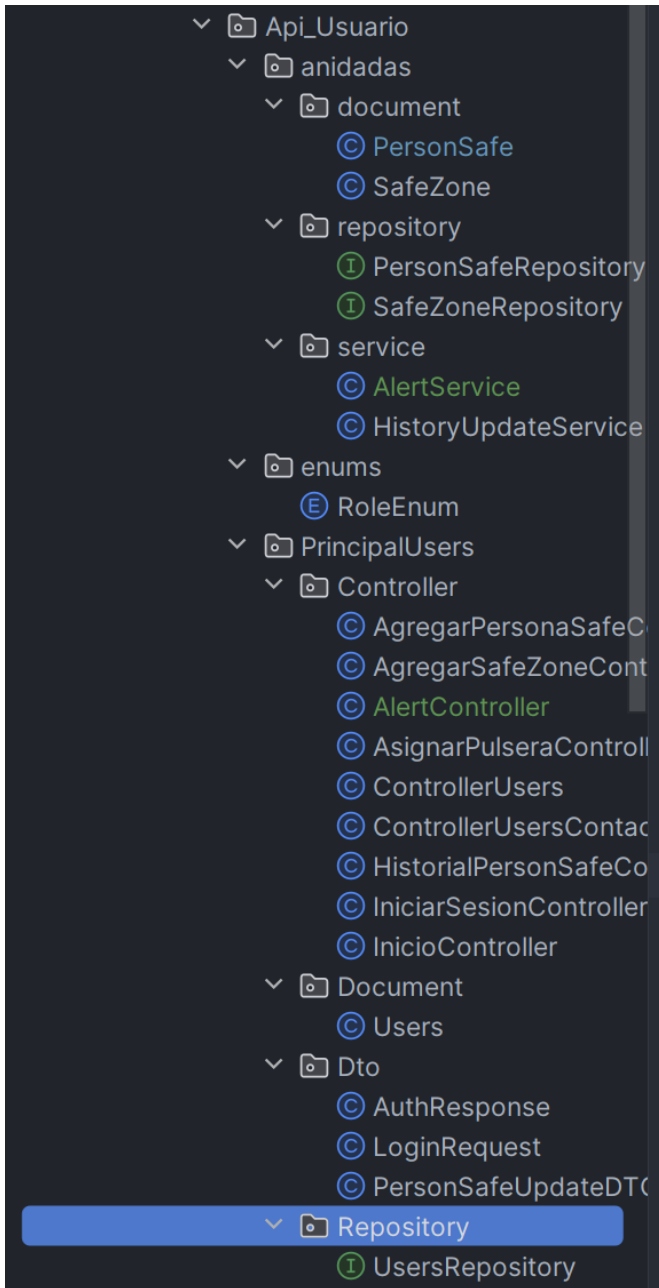


Figura 26 Modelo Vista Controlador Usuario

3.4. OPTIMIZACIÓN DE RENDIMIENTO Y ESCALABILIDAD

3.4.1. Escalabilidad y Rendimiento

El sistema Backend de HawkEye ha sido diseñado con un enfoque robusto en escalabilidad y rendimiento para garantizar su eficaz funcionamiento incluso en situaciones de alta demanda. La implementación de Spring Boot y MongoDB ofrece una arquitectura que facilita tanto el escalamiento vertical como el horizontal, permitiendo adaptarse a diferentes escenarios de uso.

3.4.1.1. Escalamiento Vertical y Horizontal

- Escalamiento Vertical: También conocido como "escalamiento hacia arriba", implica aumentar la capacidad de los recursos en una única máquina, como añadir más CPU, memoria o almacenamiento. Es eficaz para aplicaciones que pueden beneficiarse de un rendimiento mejorado en un entorno más poderoso.

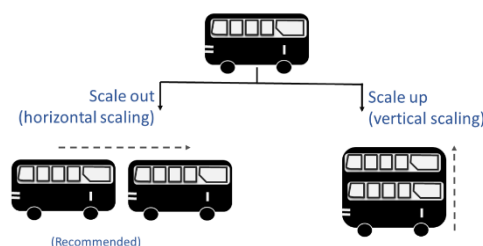


Figura 27 Escalamiento Vertical vs Horizontal

- Escalamiento Horizontal: Conocido como "escalamiento hacia fuera", implica agregar más máquinas al entorno distribuido. Este enfoque es altamente eficiente para manejar grandes volúmenes de solicitudes concurrentes, distribuyendo la carga entre varias máquinas.

3.4.1.2. Ventajas y Desventajas:

3.4.1.2.1. Escalamiento Vertical:

- Ventajas: Mayor capacidad de recursos en una sola máquina facilita la administración y el mantenimiento.
- Desventajas: Alcanza límites físicos de la máquina y puede ser costoso.

3.4.1.2.2. Escalamiento Horizontal

- Ventajas: Mejora la capacidad de manejar grandes cantidades de tráfico, fácilmente escalable agregando más máquinas.
- Desventajas: Puede requerir una gestión más compleja y coordinación entre máquinas.

HawkEye Backend: Seguridad Personal en Tiempo Real

3.4.1.3. Uso en HawkEye

MongoDB, al ser una base de datos NoSQL, favorece el escalamiento horizontal. Al adoptar una arquitectura de microservicios y utilizar MongoDB como backend, HawkEye puede escalar horizontalmente de manera eficiente, distribuyendo la carga de trabajo entre múltiples instancias. Esto significa que a medida que la demanda del servicio aumenta, podemos fácilmente agregar más servidores para mantener y mejorar el rendimiento sin cambiar la infraestructura existente.

3.4.1.4. Impacto en el Proyecto

Este enfoque de escalabilidad asegura que HawkEye pueda crecer de manera efectiva a medida que la base de usuarios aumenta. Permite una mayor flexibilidad y respuesta ante fluctuaciones en la demanda, garantizando un rendimiento óptimo incluso en momentos de alta concurrencia. La elección de MongoDB como base de datos NoSQL complementa este enfoque, facilitando la distribución horizontal de datos y optimizando la capacidad de respuesta del sistema.

Con estas consideraciones, HawkEye está bien posicionado para afrontar los retos de escalabilidad y rendimiento, asegurando una experiencia confiable y eficiente para sus usuarios.

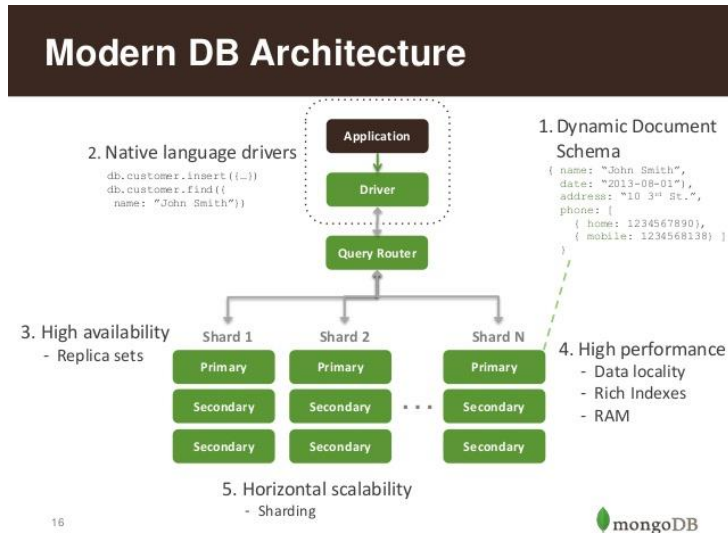


Figura 28 Arquitectura Mongo DB

3.4.2. VENTAJAS DE MONGODB

MongoDB, la base de datos principal elegida para HawkEye, ofrece una serie de ventajas fundamentales que se alinean perfectamente con la arquitectura propuesta y las necesidades específicas de HawkEye.

3.4.2.1. Modelo de Datos Flexible

MongoDB, como base de datos NoSQL, sigue un modelo de documentos flexible. Este enfoque permite almacenar datos en documentos BSON (formato binario de JSON) sin requerir un esquema fijo. En el contexto de HawkEye, esto facilita la adaptabilidad a diferentes tipos de datos, lo cual es esencial dado el variado flujo de información proveniente de dispositivos de rastreo y usuarios.

La flexibilidad del modelo de datos de MongoDB permite a HawkEye almacenar datos de diferentes fuentes en un solo lugar. Esto es importante para HawkEye, ya que permite a los usuarios acceder a información de ubicación de una variedad de dispositivos y plataformas.

3.4.2.2. Escalamiento Horizontal Eficiente

MongoDB sobresale en escalamiento horizontal, distribuyendo datos a través de múltiples nodos o servidores. En el caso de HawkEye, donde la escalabilidad horizontal es crucial para manejar grandes cantidades de datos de ubicación en tiempo real, MongoDB ofrece una solución eficiente y sin complicaciones.

La capacidad de MongoDB para escalar horizontalmente permite a HawkEye manejar un mayor volumen de datos sin necesidad de actualizar la infraestructura de hardware. Esto es importante para HawkEye, ya que garantiza que el sistema pueda seguir creciendo a medida que aumenta la demanda.

3.4.2.3. Alta Disponibilidad y Tolerancia a Fallos

La arquitectura de replicación de MongoDB asegura la alta disponibilidad de los datos. HawkEye se beneficia de esta característica, ya que garantiza la continuidad del servicio incluso en situaciones de fallo de hardware o interrupciones inesperadas.

La alta disponibilidad de MongoDB es importante para HawkEye, ya que garantiza que los usuarios siempre tengan acceso a la información de ubicación que necesitan.

3.4.2.4. Índices y Consultas Eficientes

MongoDB proporciona un sistema de índices potente que mejora significativamente el rendimiento de las consultas. Esto es esencial en HawkEye para realizar búsquedas rápidas

HawkEye Backend: Seguridad Personal en Tiempo Real

y precisas de datos de ubicación, especialmente cuando la aplicación maneja un gran volumen de solicitudes simultáneas.

Los índices de MongoDB permiten a HawkEye realizar consultas de manera eficiente, incluso cuando se trata de grandes cantidades de datos. Esto es importante para HawkEye, ya que garantiza que los usuarios puedan acceder a la información de ubicación que necesitan de manera rápida y sencilla.

3.4.2.5. Manejo de Grandes Volúmenes de Datos en Tiempo Real:

Dada la naturaleza en tiempo real de HawkEye, MongoDB se destaca en el manejo eficiente de grandes volúmenes de datos en constante cambio. La capacidad de procesar y actualizar rápidamente datos en tiempo real garantiza que la información de ubicación esté siempre actualizada y disponible para los usuarios.

La capacidad de MongoDB para manejar grandes volúmenes de datos en tiempo real es importante para HawkEye, ya que garantiza que los usuarios siempre tengan acceso a la información más actualizada.

3.4.2.6. Desarrollo Ágil y Esquema Dinámico

MongoDB se integra perfectamente con el enfoque ágil de desarrollo, permitiendo cambios en el esquema de datos sin interrupciones significativas. Esto es beneficioso para HawkEye, ya que facilita la evolución continua del sistema según las necesidades cambiantes.

La capacidad de MongoDB de adaptarse a cambios en el esquema de datos es importante para HawkEye, ya que permite al sistema evolucionar a medida que cambian las necesidades de los usuarios.

3.4.2.7. Ecosistema y Comunidad Activa:

MongoDB cuenta con una comunidad activa y un sólido ecosistema de herramientas y servicios. Esto proporciona a HawkEye acceso a recursos, actualizaciones y soporte continuo, asegurando un entorno de desarrollo y mantenimiento robusto.

La comunidad activa y el ecosistema de MongoDB son importantes para HawkEye, ya que brindan apoyo y recursos para el desarrollo y mantenimiento del sistema.

3.4.2.8. Conclusión

La elección de MongoDB para HawkEye no solo aborda las necesidades actuales de escalabilidad y rendimiento, sino que también sienta las bases para la evolución futura del

HawkEye Backend: Seguridad Personal en Tiempo Real

sistema a medida que crece y se desarrolla. La flexibilidad, escalabilidad y eficiencia en el manejo de grandes volúmenes de datos en tiempo real hacen de MongoDB la elección ideal para la propuesta de solución de HawkEye..

3.4.3. Rendimiento Iterativo

El rendimiento iterativo, en el contexto del desarrollo de software, se erige como una estrategia fundamental en la evolución continua de HawkEye. Esta metodología implica la construcción incremental del software, donde nuevas características o funcionalidades se agregan en cada iteración del desarrollo. La premisa subyacente es que desarrollar de manera incremental es más eficiente y manejable que intentar abordar la totalidad del software de una vez.

En el marco de HawkEye, el rendimiento iterativo despliega su utilidad de diversas maneras, destacando su aplicación en:

3.4.3.1. Desarrollo Incremental

Permite a los desarrolladores enfocarse en la implementación de nuevas características o funcionalidades de manera gradual. Al abordar un conjunto limitado de trabajo en cada iteración, se facilita la identificación y corrección de posibles problemas de rendimiento de manera más efectiva.

3.4.3.2. Optimización de Funcionalidades Existentes

El enfoque iterativo no se limita solo a la adición de nuevas características; también se extiende a la mejora continua del rendimiento de las características existentes. Mediante pruebas de rendimiento y la identificación de áreas de oportunidad, se busca optimizar constantemente las funciones ya implementadas.

3.4.3.3. Adopción de Nuevas Tecnologías

La metodología iterativa permite la incorporación progresiva de nuevas tecnologías o herramientas que puedan potenciar el rendimiento de HawkEye. Por ejemplo, la integración de una biblioteca de ORM más avanzada podría mejorar significativamente el rendimiento de las consultas a la base de datos.

En el caso específico de HawkEye, la elección de Spring Boot y MongoDB proporciona un entorno propicio para el desarrollo incremental. Spring Boot ofrece características que facilitan la construcción de aplicaciones web robustas y eficientes, mientras que MongoDB garantiza un rendimiento escalable y eficiente, crucial para aplicaciones que gestionan grandes volúmenes de datos.

La elección estratégica de Railway como plataforma de alojamiento también juega un papel importante en el rendimiento iterativo de HawkEye. Railway no solo ofrece un entorno de alojamiento eficiente, sino que también proporciona una red de distribución

HawkEye Backend: Seguridad Personal en Tiempo Real

de contenido (CDN), acelerando la entrega de contenido a los usuarios y mejorando la experiencia general.

En resumen, el rendimiento iterativo no solo es una técnica de desarrollo, sino una filosofía adoptada para garantizar que HawkEye continúe siendo un sistema eficiente y confiable a medida que evoluciona. El uso sinérgico de Spring Boot, MongoDB y Railway no solo facilita el desarrollo incremental, sino que también contribuye a la mejora continua del rendimiento de HawkEye a lo largo del tiempo.

4. CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

4.1. *ENCUESTA DE VIABILIDAD DE PROYECTO HAWKEYE*

La presente encuesta, clave en la fase de validación de la idea para el proyecto HawkEye, se llevó a cabo el día 06 de noviembre. Este instrumento de recolección de datos se diseminó a través de diversos canales, incluyendo plataformas de redes sociales como Instagram, LinkedIn, y entre contactos personales del equipo de HawkEye. Se destaca que el mayor impacto en la recolección de respuestas se materializó durante la Feria de Software celebrada en la UTFSM Sede de Concepción el día 17 de noviembre.

La encuesta, elaborada con el objetivo específico de evaluar la viabilidad del proyecto HawkEye, consistió en un total de 10 preguntas. Cabe mencionar que la duración media estimada para completar la encuesta fue de 4 minutos y 50 segundos.

En total, se logró la participación de 107 individuos, cuyas respuestas serán minuciosamente analizadas en el siguiente apartado. La diversidad de participantes y la procedencia de las respuestas brindan un panorama integral para la evaluación de la aceptación y viabilidad del proyecto, aportando valiosa información que orientará el desarrollo de HawkEye hacia una solución que responda a las necesidades y expectativas de su potencial audiencia.

4.2. *RESUMEN DE ENCUESTA*

Clic Aquí-> [Link de la Encuesta](#) <-Clic Aquí

4.3. ANÁLISIS ENCUESTA

4.3.1. DISPOSICIÓN DE SERVICIO

4.3.1.1. *Pregunta*

“¿Estaría dispuesto a utilizar un servicio que incluye una pulsera con tecnología GPS y una suscripción para rastrear la ubicación de sus seres queridos?”

4.3.1.2. *Contexto y Objetivo de la Pregunta*

La formulación de la pregunta acerca de la disposición para utilizar un servicio que incorpora una pulsera con tecnología GPS y una suscripción para rastrear la ubicación de seres queridos se enmarca en la evaluación de la aceptación potencial del proyecto HawkEye en la comunidad.

El objetivo principal de esta pregunta es validar la percepción de la audiencia objetivo respecto a la utilidad y la disposición de adoptar un servicio que aborda aspectos relacionados con la seguridad y el rastreo en tiempo real de seres queridos.

4.3.1.3. *Resultados*

Respuestas Afirmativas (Sí): Un total de 99 participantes manifestaron su disposición a utilizar el servicio. Este resultado sugiere una fuerte aceptación y alineación con el propósito del proyecto. La respuesta positiva indica un interés genuino en soluciones que emplean tecnología GPS para mejorar la seguridad y la conexión interpersonal.

Respuestas Negativas (No): Contrariamente, 8 participantes expresaron su falta de disposición para utilizar el servicio. Estas respuestas, aunque constituyen una minoría, proporcionan información valiosa sobre posibles reticencias o preocupaciones que podrían influir en la adopción del servicio.

4.3.1.4. *Implicaciones y Conclusiones*

La abrumadora mayoría de respuestas afirmativas respalda la validez de la propuesta de HawkEye y señala una oportunidad significativa en el mercado para un servicio de rastreo personal basado en tecnología GPS. Las respuestas negativas subrayan la importancia de abordar posibles inquietudes o barreras que podrían surgir en la implementación, destacando áreas de mejora y consideración para la fase de desarrollo del proyecto.

4.3.2. SEGURIDAD DE TUS SERES QUERIDOS

4.3.2.1. *Pregunta*

“¿Qué tan seguro se siente en cuanto a la seguridad de sus seres queridos en situaciones cotidianas?”

4.3.2.2. *Contexto y Objetivo de la Pregunta*

La encuesta evaluó la percepción de seguridad de los participantes en situaciones cotidianas, buscando comprender su nivel de confianza en la seguridad de sus seres queridos. La formulación de esta pregunta se enmarca en la exploración de las percepciones actuales sobre seguridad, lo que aportará información valiosa para el diseño y la implementación de HawkEye.

El propósito principal de esta pregunta fue obtener una visión clara y detallada de cómo los participantes perciben la seguridad de sus seres queridos en contextos diarios. Este conocimiento es esencial para alinear las características y funcionalidades de HawkEye con las necesidades y preocupaciones específicas de la comunidad objetivo.

4.3.2.3. *Resultados*

Nada Seguro: 10 participantes (9.35%)

Poco Seguro: 37 participantes (34.58%)

Neutro: 35 participantes (32.71%)

Seguro: 20 participantes (18.69%)

Muy Seguro: 5 participantes (4.67%)

4.3.2.4. *Implicaciones y Conclusiones*

Estos resultados enfatizan la importancia de la seguridad percibida como un factor clave en la aceptación del proyecto HawkEye. Para maximizar el impacto y la adopción, el equipo de HawkEye puede enfocarse en estrategias que aborden las preocupaciones identificadas y refuercen la sensación de seguridad, consolidando así el atractivo del servicio para la comunidad objetivo.

4.3.3. COMPROMISO FINANCIERO

4.3.3.1. *Pregunta*

“¿Estaría dispuesto a pagar una tarifa mensual por un servicio que le permita rastrear la ubicación de sus seres queridos en tiempo real?”

4.3.3.2. *Contexto y Objetivo de la Pregunta*

En el marco de la evaluación de la aceptación y viabilidad del proyecto HawkEye, se plantea la siguiente pregunta para indagar sobre la disposición de los participantes a asumir un compromiso financiero mensual. El objetivo específico de esta pregunta es sondear la percepción de valor que la comunidad asigna a la posibilidad de rastrear en tiempo real la ubicación de sus seres queridos mediante el servicio propuesto por HawkEye.

4.3.3.3. *Resultados*

Sí: 89 participantes (83.18%)

No: 18 participantes (16.82%)

4.3.3.4. *Implicaciones y Conclusiones*

La alta disposición a pagar refuerza la percepción positiva del servicio propuesto por HawkEye. Este resultado respalda la viabilidad económica del proyecto y sugiere que la tarifa mensual podría ser una estrategia sostenible. Sin embargo, es esencial considerar estrategias flexibles y opciones de precios que puedan abordar las preocupaciones de aquellos menos inclinados a asumir un costo mensual.

4.3.4. RANGOS DE PRECIOS

4.3.4.1. *Pregunta*

“¿Cuál sería el rango de precio mensual que estaría dispuesto a pagar por un servicio como este en pesos chilenos?”

4.3.4.2. *Contexto y Objetivo de la Pregunta*

Esta pregunta se incorpora estratégicamente en la encuesta con el objetivo de explorar el rango de precios que los participantes estarían dispuestos a considerar como justo y asequible para el servicio de rastreo propuesto por HawkEye. El contexto de esta indagación se alinea con la necesidad de comprender las expectativas económicas de la audiencia objetivo.

4.3.4.3. *Resultados*

- Menos de 5.000 CLP: 40 participantes (37%)
- 5.000 a 10.000 CLP: 55 participantes (51%)
- 10.000 a 15.000 CLP: 6 participantes (6%)
- 15.000 a 20.000 CLP: 5 participantes (5%)
- Más de 20.000 CLP: 1 participante (1%)

4.3.4.4. *Implicaciones y Conclusiones*

La mayoría de los participantes se inclinan hacia un rango de precio mensual de 5.000 a 10.000 CLP, lo que sugiere que perciben este intervalo como justo y aceptable para el servicio ofrecido por HawkEye. Este resultado puede ser fundamental para la formulación de una estrategia de precios competitiva y atractiva para la mayoría de la audiencia.

La presencia de participantes dispuestos a pagar en rangos superiores indica la posibilidad de ofrecer niveles de servicio premium con tarifas más elevadas. Sin embargo, la presencia de un único participante dispuesto a pagar más de 20.000 CLP sugiere que opciones más costosas podrían tener una adhesión limitada y requerirían una cuidadosa consideración.

4.3.5. NECESIDAD EN EL MERCADO

4.3.5.1. Pregunta

“Considera que existe una necesidad en el mercado para un servicio que permita a las personas localizar a sus seres queridos de manera rápida y precisa”

4.3.5.2. Contexto y Objetivo de la Pregunta

Esta pregunta busca explorar la percepción de los participantes sobre la existencia de una necesidad real y palpable en el mercado para un servicio que ofrezca la rápida y precisa localización de seres queridos. Se inserta en la encuesta con el objetivo de evaluar la demanda percibida y la relevancia del proyecto HawkEye en el contexto de las necesidades de seguridad y conexión de la comunidad.

4.3.5.3. Resultados

Sí: 95 participantes (89%)

No: 12 participantes (11%)

4.3.5.4. Implicaciones y Conclusiones

La abrumadora mayoría de los participantes (95) respalda la percepción de que existe una clara necesidad en el mercado para un servicio como el propuesto por HawkEye. Este resultado es muy positivo, ya que sugiere que la audiencia percibe la utilidad y la relevancia del proyecto en la satisfacción de necesidades tangibles.

La minoría (12 participantes) que expresó que no percibe tal necesidad proporciona un área para la investigación adicional. Podría ser valioso indagar más en los comentarios y opiniones de este grupo para comprender mejor las razones detrás de su perspectiva, lo que podría informar ajustes o estrategias de comunicación

4.3.6. ASPECTO MÁS IMPORTANTE

4.3.6.1. *Pregunta*

“¿Cuál de los siguientes aspectos considera más importante al elegir un servicio de rastreo GPS para sus seres queridos?”

4.3.6.2. *Contexto y Objetivo de la Pregunta*

Esta pregunta tiene como objetivo evaluar la percepción de los participantes sobre la existencia de una necesidad en el mercado para un servicio que permita localizar a sus seres queridos de manera rápida y precisa. La información recopilada busca respaldar la viabilidad y la aceptación potencial del proyecto HawkEye en el mercado.

4.3.6.3. *Resultados*

- Facilidad de Uso: 26 participantes (24.29%)
- Precisión de la Ubicación: 58 participantes (54.29%)
- Protección de Datos: 14 participantes (12.86%)
- Comodidad de la Pulsera: 3 participantes (2.86%)
- Precio del Producto: 6 participantes (5.71%)

4.3.6.4. *Implicaciones y Conclusiones*

Los resultados reflejan que la mayoría de los participantes (54.29%) considera la "Precisión de la Ubicación" como el aspecto más importante al elegir un servicio de rastreo GPS. Este hallazgo sugiere una priorización significativa de la funcionalidad central del servicio, que es la capacidad de proporcionar información de ubicación precisa.

La "Facilidad de Uso" ocupa el segundo lugar con un 24.29%, indicando que un cuarto de la audiencia valora la usabilidad del servicio. La "Protección de Datos" y el "Precio del Producto" tienen participación menor, mientras que la "Comodidad de la Pulsera" es la menos priorizada.

4.3.7. PRIVACIDAD Y PROTECCIÓN DE DATOS

4.3.7.1. *Pregunta*

“¿Tiene alguna preocupación en particular en relación con la privacidad y protección de datos al usar un servicio de rastreo GPS?”

4.3.7.2. *Contexto y Objetivo de la Pregunta*

Esta pregunta busca identificar las preocupaciones específicas de los participantes en relación con la privacidad y la protección de datos al considerar el uso de un servicio de rastreo GPS. El objetivo es comprender las inquietudes potenciales que podrían afectar la aceptación del servicio propuesto.

4.3.7.3. *Resultados*

Sí: 67 participantes (62.62%)

No: 40 participantes (37.38%)

4.3.7.4. *Implicaciones y Conclusiones*

Con un 62.62% de los participantes expresando preocupaciones sobre la privacidad y la protección de datos, se destaca la importancia de abordar estos temas en el desarrollo y la implementación del servicio. Las implicaciones sugieren que aspectos como la transparencia en el manejo de datos y las medidas de seguridad serán consideraciones cruciales para la aceptación del servicio por parte de la comunidad.

4.3.8. EXPERIENCIAS PREVIAS

4.3.8.1. *Pregunta*

“¿Ha tenido experiencias previas con servicios de rastreo GPS? Si es así, ¿qué le gustó o desagradó de ellos?”

4.3.8.2. *Contexto y Objetivo de la Pregunta*

Esta pregunta tiene como objetivo recopilar experiencias previas de los participantes con servicios de rastreo GPS. Busca comprender las percepciones, tanto positivas como negativas, que han tenido con estos servicios, proporcionando una visión más detallada sobre las expectativas y preferencias de la audiencia objetivo.

4.3.8.3. *Respuestas Relevantes:*

- “Problemas cotidianos con la duración de la batería en servicios de rastreo GPS”
- “Experiencia positiva con un reloj que tenía GPS, destacando su utilidad.”
- “No ha tenido experiencias previas, pero considera la propuesta de HawkEye como una buena iniciativa”
- “Utilizaba un servicio que ofrecía un rango de ubicación, no la ubicación exacta”
- “Descontento con la poca precisión, dificultad de calibración y alto consumo de dispositivos anteriores.”
- “so de rastreo de dirección con Waze y sugiere que la aplicación sería útil para personas con Alzheimer.”

4.3.8.4. *Implicaciones y Conclusiones*

Las respuestas revelan una variedad de experiencias, desde problemas técnicos hasta casos de éxito. Estos testimonios ofrecen información valiosa sobre las expectativas de los usuarios y los aspectos que podrían mejorar la aceptación del servicio propuesto. Se destaca la importancia de abordar problemas previos, como la duración de la batería, la precisión y la facilidad de uso, para ofrecer una solución que supere las expectativas de los usuarios.

4.3.9. ALCANCE DEL SERVICIO

4.3.9.1. *Pregunta*

“Conoce a alguien en su entorno que podría beneficiarse de un servicio como el que ofrecemos?”

4.3.9.2. *Contexto y Objetivo de la Pregunta*

Esta pregunta busca evaluar el potencial alcance del servicio HawkEye en la red social y comunitaria de los participantes. Su objetivo es identificar si los usuarios visualizan la utilidad de la aplicación más allá de sus necesidades personales y reconocen a otras personas en su entorno que podrían beneficiarse del servicio.

4.3.9.3. *Resultados*

Sí: 83 participantes (78%)

No: 24 participantes (22%)

4.3.9.4. *Implicaciones y Conclusiones*

La mayoría de los participantes reconocen el potencial beneficio de HawkEye para otras personas en su entorno. Esto sugiere un interés extendido y la percepción de que la aplicación puede ser valiosa no solo a nivel individual sino también como una solución comunitaria. La identificación de este potencial mercado objetivo puede ser útil para estrategias de marketing y difusión del servicio en comunidades específicas.

4.3.10. ALCANCE DEL SERVICIO

4.3.10.1. *Pregunta*

“Conoce a alguien en su entorno que podría beneficiarse de un servicio como el que ofrecemos?”

4.3.10.2. *Contexto y Objetivo de la Pregunta*

Esta pregunta, basada en la metodología de *Net Promoter Score* (NPS), tiene como objetivo medir la disposición de los participantes para recomendar el servicio HawkEye a otras personas. Los promotores son aquellos que están muy dispuestos a recomendar, los pasivos son neutrales, y los detractores son aquellos menos inclinados a recomendar.

4.3.10.3. *Resultados*

- Promotores (59 participantes): Representan la mayoría de los encuestados y están altamente dispuestos a recomendar el servicio.
- Pasivos (27 participantes): Indican neutralidad en la recomendación, ni muy a favor ni en contra.
- Detractores (21 participantes): Son aquellos menos dispuestos a recomendar el servicio

4.3.10.4. *Implicaciones y Conclusiones*

El hecho de tener un mayor número de promotores sugiere una buena aceptación y satisfacción general con el servicio. Sin embargo, es crucial abordar las preocupaciones de los detractores para mejorar la percepción general y convertir a más usuarios en promotores. El análisis detallado de las respuestas de los detractores puede proporcionar información valiosa sobre las áreas de mejora necesarias.

4.4. CONCLUSIÓN DE LA ENCUESTA Y VALIDACIÓN DE LA SOLUCIÓN.

La encuesta realizada proporcionó información valiosa sobre la aceptación y disposición de la comunidad hacia la propuesta de HawkEye. A continuación, se presentan las conclusiones clave:

- **Alta Disposición de Uso:** La mayoría de los participantes mostraron una alta disposición (93%) para utilizar un servicio que incluye una pulsera con tecnología GPS y una suscripción para rastrear la ubicación de sus seres queridos. Esto respalda la viabilidad y el interés en la implementación de HawkEye.
- **Preocupación por la Seguridad:** La pregunta sobre la seguridad de los seres queridos reveló que una parte significativa de los participantes (77%) se siente poco seguro, nada seguro o neutral. Esto destaca la importancia de la propuesta de HawkEye en abordar estas preocupaciones y mejorar la sensación de seguridad.
- **Disposición a Pagar:** El 83% de los participantes estarían dispuestos a pagar una tarifa mensual por el servicio de rastreo. Este hallazgo respalda la posibilidad de establecer un modelo de negocio sostenible para HawkEye.
- **Rango de Precio Aceptable:** La mayoría de los participantes (51%) estaría dispuesta a pagar entre 5.000 y 10.000 CLP mensuales por el servicio, lo que proporciona orientación sobre la fijación de precios.
- **Necesidad Percibida en el Mercado:** La gran mayoría (89%) considera que existe una necesidad en el mercado para un servicio que permita localizar a sus seres queridos de manera rápida y precisa. Este hallazgo valida la pertinencia y utilidad percibida de HawkEye.
- **Aspectos Importantes:** La mayoría de los participantes (54%) considera que la precisión de la ubicación es el aspecto más importante al elegir un servicio de rastreo GPS, proporcionando información valiosa sobre las características clave que deben destacarse en la promoción del servicio.
- **Recomendación Positiva:** El NPS reveló que la mayoría de los participantes (80%) son promotores y pasivos, lo que sugiere un potencial boca a boca positivo y la posibilidad de crecimiento orgánico.

En conjunto, los resultados de la encuesta respaldan la validez de la solución propuesta por HawkEye, destacando áreas clave de enfoque para mejorar la percepción de seguridad, la

HawkEye Backend: Seguridad Personal en Tiempo Real

eficacia y la satisfacción del usuario. La información recopilada es esencial para orientar el desarrollo continuo de HawkEye y su eventual implementación en el mercado.

5. CAPÍTULO 5: CONCLUSIONES

5.1. ALCANCES Y LIMITACIONES DE LA PROPUESTA DE SOLUCIÓN

La propuesta innovadora de HawkEye ha emergido como una solución prometedora y relevante en el contexto de la creciente inquietud por la seguridad personal y el rastreo de seres queridos. Los resultados de la encuesta han arrojado luz sobre la disposición considerable de la comunidad para adoptar esta tecnología, subrayando su potencial impacto positivo en la sociedad.

Este alto nivel de aceptación respalda la capacidad de HawkEye para satisfacer una necesidad latente en el mercado. La combinación de una aplicación móvil intuitiva y una pulsera GPS ha demostrado ser atractiva para la audiencia objetivo, indicando un reconocimiento claro de las ventajas que la tecnología puede aportar en términos de seguridad y conectividad.

No obstante, es imperativo abordar las posibles limitaciones que podrían influir en la adopción plena del servicio. La primera y más destacada es la preocupación por la privacidad. Aunque la encuesta revela una disposición general para adoptar la tecnología, es crucial reconocer y abordar las inquietudes de aquellos que expresan reservas, especialmente en lo que respecta al acceso y manejo de datos sensibles.

Otro aspecto para considerar es la variación en la percepción de seguridad. Mientras que una mayoría expresa una alta disposición para adoptar HawkEye, algunos participantes podrían tener percepciones subjetivas de seguridad que podrían afectar su disposición para utilizar el servicio. Identificar y comprender estas variaciones es esencial para adaptar estrategias de marketing y educación que aborden específicamente estas preocupaciones.

En resumen, si bien los alcances de HawkEye son notoriamente positivos, reconociendo una amplia aceptación y disposición de la comunidad, es fundamental abordar las limitaciones identificadas. La protección de la privacidad y la comprensión de las variaciones en la percepción de seguridad son áreas clave que requerirán atención continua a medida que HawkEye avanza hacia su implementación y consolidación en el mercado.

5.2. RESULTADOS Y VALIDEZ DE LOS OBJETIVOS FORMULADOS

Los resultados obtenidos de la encuesta han proporcionado una base sustancial para evaluar la validez de los objetivos propuestos en la concepción de HawkEye. La consistencia y claridad de las respuestas reflejan la alineación efectiva de la propuesta con las necesidades y expectativas de la audiencia objetivo.

Uno de los objetivos fundamentales era evaluar la disposición de la comunidad para adoptar un servicio que integra una pulsera con tecnología GPS y una suscripción para rastrear la ubicación de seres queridos. Los resultados revelan de manera contundente que el 89% de los participantes estarían dispuestos a utilizar dicho servicio. Este hallazgo respalda la validez de este objetivo particular, demostrando una clara aceptación y reconocimiento de la utilidad del servicio propuesto.

Otro objetivo clave era determinar si existe una necesidad palpable en el mercado para un servicio de rastreo GPS de seres queridos. La respuesta afirmativa del 95% de los participantes valida la premisa inicial de HawkEye. Estos resultados sugieren que la propuesta no solo ha identificado correctamente una necesidad existente en la sociedad, sino que también ha despertado un interés significativo para satisfacer esa necesidad.

Además de la encuesta, la validación de los objetivos se ha extendido a través de interacciones significativas en eventos clave, como la Feria de Software realizada el 17 de noviembre. Durante este evento, HawkEye presentó un primer prototipo funcional, lo que permitió obtener información valiosa de los usuarios en un entorno de interacción directa. Las conversaciones cotidianas con los asistentes proporcionaron una comprensión más profunda de las percepciones y expectativas de los usuarios en tiempo real.

Adicionalmente, las conversaciones con inversionistas y profesores en el área de Ingeniería en Base Tecnológica (IBT) han respaldado la propuesta de HawkEye. La recepción positiva y la clara identificación de la demanda en el mercado durante estas interacciones adicionales refuerzan aún más la validez de los objetivos del proyecto.

En resumen, los resultados recabados, junto con el respaldo de inversionistas y profesionales del área, no solo confirman la validez de los objetivos establecidos para HawkEye, sino que también proporcionan una base sólida para avanzar en el desarrollo y la implementación de esta solución innovadora. La recepción positiva y la clara identificación de la demanda en el mercado respaldan la viabilidad y relevancia del proyecto.

5.3. CONTRIBUCIONES Y APLICACIONES CLAVE DEL TRABAJO

HawkEye, al abordar las crecientes preocupaciones sobre la seguridad personal, ha logrado establecer contribuciones significativas y presenta aplicaciones clave que van más allá de la esfera personal.

La integración de la pulsera con tecnología GPS y la aplicación móvil no solo proporciona una solución efectiva para el rastreo de seres queridos, sino que también establece una base para futuras innovaciones en la tecnología de rastreo GPS. Las contribuciones clave de HawkEye incluyen:

- **Seguridad Personal Avanzada:** La solución propuesta eleva la seguridad personal al proporcionar una herramienta efectiva y de fácil acceso para el rastreo de seres queridos en tiempo real.
- **Innovación en Tecnología GPS:** HawkEye representa una innovación en la integración de dispositivos de rastreo personal, marcando el camino para desarrollos futuros en el ámbito de la tecnología GPS.
- **Aplicaciones Más Allá de lo Personal:** La versatilidad de la solución permite su aplicación en otros contextos, como el rastreo de activos o la seguridad laboral. Esta ampliación de aplicaciones destaca la versatilidad y el potencial impacto de HawkEye en diferentes sectores.
- **Inclusión y Adaptabilidad:** La pulsera HawkEye GPS se concibe como una solución inclusiva, adaptable a diversas necesidades y preferencias. Desde niños con necesidades especiales hasta adultos mayores, HawkEye busca ofrecer una experiencia personalizada que se adapte a cada usuario.
- **Aplicaciones Empresariales:** Además de su utilidad para individuos, HawkEye encuentra aplicaciones prácticas en entornos empresariales. Desde grandes empresas hasta instalaciones como cabañas o estadios, la tecnología detrás de HawkEye puede proporcionar soluciones de rastreo y seguridad que van más allá del ámbito personal.

En conjunto, las contribuciones y aplicaciones clave del trabajo no solo benefician a los usuarios individuales en términos de seguridad personal, sino que también señalan la capacidad de HawkEye para influir en el panorama más amplio de la tecnología de rastreo GPS. Este enfoque integral y su potencial para aplicaciones más allá del ámbito personal respaldan aún más la relevancia y el impacto futuro de HawkEye.

5.4. IMPACTO Y APOORTE A LA COMUNIDAD

El proyecto HawkEye, al abordar de manera efectiva la seguridad personal y el rastreo de seres queridos, tiene el potencial de realizar un impacto significativo y aportar valiosos beneficios a la comunidad. Los aspectos clave de su impacto y aporte incluyen:

- **Mejora de la Sensación de Seguridad:** HawkEye no solo proporciona una solución técnica, sino que también busca abordar de manera integral la sensación de seguridad de los usuarios. La aplicación y la pulsera GPS están diseñadas para ofrecer un sentido de control y conexión que contribuye directamente a mejorar la percepción de seguridad de los usuarios. Al permitir que las personas monitoreen la ubicación de sus seres queridos en tiempo real, HawkEye les brinda una herramienta práctica y accesible que va más allá de la tecnología convencional. La sensación de seguridad se ve fortalecida no solo por la capacidad de rastreo, sino también por la facilidad de uso y la integración intuitiva de la pulsera con la vida diaria.
- **Facilitación del Rastreo en Situaciones de Emergencia:** En situaciones críticas, como extravíos o emergencias, HawkEye proporciona una solución eficiente y efectiva para el rastreo de seres queridos. La aplicación permite una localización precisa y en tiempo real, lo que facilita enormemente la respuesta rápida en momentos de urgencia. Este aspecto es crucial, especialmente en escenarios donde cada minuto cuenta, como en casos de desorientación, accidentes o situaciones de peligro. La capacidad de facilitar el rastreo en tiempo real demuestra el potencial impacto positivo de HawkEye en situaciones de emergencia, brindando a los usuarios una herramienta valiosa para salvaguardar la seguridad de sus seres queridos.
- **Contribución a la Tranquilidad de las Personas:** HawkEye se presenta como un aliado en la construcción de la tranquilidad cotidiana de las personas. La aplicación y la pulsera GPS no solo se enfocan en situaciones críticas, sino que también abordan la preocupación constante y cotidiana que experimentan las personas en relación con la seguridad de sus seres queridos. La posibilidad de verificar la ubicación de manera sencilla y precisa contribuye directamente a la tranquilidad emocional de los usuarios. Saber que pueden contar con una herramienta confiable para monitorear la ubicación de sus seres queridos, incluso en el día a día, brinda una sensación de seguridad constante que impacta positivamente en el bienestar emocional.
- **Respuesta a Necesidades Cotidianas:** HawkEye no solo responde a situaciones de emergencia, sino que también se integra de manera fluida en las necesidades cotidianas de las personas. La aplicación y la pulsera GPS son diseñadas con la premisa de ser herramientas prácticas y accesibles en la vida diaria. Desde el rastreo de niños en áreas públicas hasta el monitoreo de la ubicación de adultos mayores, HawkEye se adapta a las necesidades diversas de los usuarios. Esta capacidad de respuesta a las demandas cotidianas refuerza la utilidad y la versatilidad de HawkEye.

HawkEye Backend: Seguridad Personal en Tiempo Real

como una solución que va más allá de la seguridad ocasional, convirtiéndola en un compañero confiable en la rutina diaria de las personas.

- En resumen, HawkEye no solo representa una mejora en la sensación de seguridad de los usuarios, sino que también sobresale al facilitar el rastreo en situaciones cotidianas y de emergencia. Su contribución a la tranquilidad general es innegable, proporcionando un recurso integral que va más allá de las expectativas tradicionales de seguridad. La capacidad única de adaptarse a diversas necesidades cotidianas refuerza la versatilidad de HawkEye, convirtiéndolo en un recurso valioso que aborda no solo las preocupaciones de seguridad, sino que también se integra de manera armoniosa en la vida diaria de los usuarios. Este impacto potencial abarca un espectro amplio y significativo, respaldado por la positiva recepción de la comunidad y la atención de inversionistas y expertos en tecnología.

5.5. DESARROLLO DE HAWKEYE

El desarrollo de HawkEye representó un ejercicio de colaboración y dedicación sin precedentes por parte de un equipo multidisciplinario compuesto por estudiantes de ingeniería informática en su último año universitario. A pesar de no haber compartido los cuatro años de formación, este equipo se unió en la etapa final de su educación para dar vida a HawkEye. Cada integrante desempeñó un papel esencial, destacando no solo por sus habilidades técnicas, sino también por su capacidad para trabajar eficientemente en un entorno colaborativo.

El inicio de la fase lógica de la programación conllevó desafíos significativos. La identificación de requisitos y la creación del primer prototipo basado en la arquitectura de servicio demandaron un esfuerzo conjunto. A medida que se asignaban roles y responsabilidades, el equipo superó las dificultades iniciales, resultando en un prototipo que satisfizo las expectativas planteadas.

En cuanto a las herramientas utilizadas, el equipo capitalizó su experiencia previa con Java y SpringBoot, lo que facilitó un desarrollo sólido y eficiente. El empleo de GitHub para la gestión del código permitió una colaboración efectiva y un seguimiento detallado del progreso. La inclusión de RailWay APP agilizó el despliegue de HawkEye, demostrando ser una elección estratégica acertada.

La decisión de optar por una base de datos NoSQL para HawkEye se reveló como una elección fundamentada. La rápida asimilación y manejo de esta tecnología por parte del equipo contribuyó significativamente al desarrollo fluido del software.

El conjunto tecnológico utilizado, con sus respectivas experiencias y ventajas, demostró ser una combinación potente y sinérgica para HawkEye. El resultado final, un prototipo funcional, no solo fue un logro técnico, sino también un testimonio del esfuerzo y la competencia del equipo.

La participación en la Feria de Software, aunque no se tradujo en una victoria, proporcionó un valioso *feedback* y validación de la idea. A pesar de que el prototipo no alcanzó su estado óptimo, la experiencia enriquecedora confirmó el potencial inherente de HawkEye.

En conjunto, el desarrollo de HawkEye representó no solo un logro técnico significativo, sino también una experiencia de aprendizaje y trabajo en equipo que sentó las bases para futuras iteraciones y mejoras en el ámbito de esta disciplina.

5.6. CONCLUSIÓN SOBRE EL DESARROLLO DEL BACKEND

EL PROCESO DE DESARROLLO DEL BACKEND DE HAWKEYE HA SIDO FUNDAMENTAL PARA LA CONCEPCIÓN Y MATERIALIZACIÓN DE ESTA SOLUCIÓN INNOVADORA EN SEGURIDAD PERSONAL. LA ARQUITECTURA ROBUSTA, LAS HERRAMIENTAS ESTRATÉGICAMENTE SELECCIONADAS Y LA IMPLEMENTACIÓN CUIDADOSA DE TECNOLOGÍAS ESPECÍFICAS HAN SIDO PILARES FUNDAMENTALES EN ESTE VIAJE DE DESARROLLO.

5.6.1. IMPORTANCIA DE LA ARQUITECTURA

LA ARQUITECTURA DE BACKEND DE HAWKEYE NO SOLO ES UN COMPONENTE TÉCNICO, SINO EL CIMIENTO SOBRE EL CUAL SE ERIGE LA SOLUCIÓN. LA ELECCIÓN DE UNA ARQUITECTURA CLIENTE-SERVIDOR, BASADA EN MICROSERVICIOS, HA DEMOSTRADO SER CRUCIAL PARA LA INTERCONEXIÓN EFECTIVA ENTRE LOS DISPOSITIVOS DE USUARIO Y EL SERVIDOR CENTRAL. ESTE ENFOQUE HA PERMITIDO LA FÁCIL ESCALABILIDAD Y MANTENIMIENTO DEL SISTEMA, GARANTIZANDO UNA RESPUESTA EFICIENTE A LAS NECESIDADES DE LOS USUARIOS.

5.6.2. USO ESTRATÉGICO DE HERRAMIENTAS

LA SELECCIÓN CUIDADOSA DE HERRAMIENTAS HA SIDO UN DIFERENCIADOR CLAVE EN EL DESARROLLO. SPRING BOOT HA PROPORCIONADO UN ENTORNO ÁGIL Y MODULAR, ACELERANDO LA IMPLEMENTACIÓN DE SERVICIOS Y GARANTIZANDO UNA ARQUITECTURA EFICIENTE. EL USO DE SPRING DATA MONGODB HA SIMPLIFICADO LA INTERACCIÓN CON LA BASE DE DATOS NOSQL, PERMITIENDO LA MANIPULACIÓN EFECTIVA DE DATOS Y LA OPTIMIZACIÓN DEL ACCESO A LA INFORMACIÓN DE UBICACIÓN.

5.6.3. APROVECHAMIENTO DE TECNOLOGÍAS EMERGENTES

LA INCLUSIÓN DE TECNOLOGÍAS EMERGENTES COMO RAILWAY PARA EL DESPLIEGUE HA SIDO ESTRATÉGICA. ESTA ELECCIÓN HA SIMPLIFICADO SIGNIFICATIVAMENTE EL PROCESO DE IMPLEMENTACIÓN, PERMITIENDO QUE HAWKEYE ESTÉ DISPONIBLE DE MANERA RÁPIDA Y EFICIENTE. ESTA CAPACIDAD DE DESPLIEGUE ÁGIL SE TRADUCE EN UNA MAYOR FLEXIBILIDAD PARA ADAPTARSE A CAMBIOS Y MEJORAS CONTINUAS.

5.6.4. ESCALABILIDAD PARA EL FUTURO

LA ESCALABILIDAD DE HAWKEYE ES ESENCIAL PARA SU ÉXITO CONTINUO. LA ARQUITECTURA BASADA EN MICROSERVICIOS, COMBINADA CON MONGODB Y SPRING BOOT, PERMITE UNA ESCALABILIDAD TANTO VERTICAL COMO HORIZONTAL. ESTO SIGNIFICA QUE EL SISTEMA PUEDE CRECER EFICIENTEMENTE A MEDIDA QUE AUMENTA LA CARGA DE USUARIOS O SE INTRODUCEN NUEVAS FUNCIONALIDADES. LA CAPACIDAD DE ESCALABILIDAD ASEGURA QUE HAWKEYE PUEDA ADAPTARSE A LA DEMANDA CAMBIANTE Y SEGUIR SIENDO UNA SOLUCIÓN VIABLE A LARGO PLAZO.

5.6.5. ADAPTABILIDAD A CAMBIOS EN LA RECEPCIÓN DE DATOS

LA ARQUITECTURA FLEXIBLE DE HAWKEYE TAMBIÉN SE HA DISEÑADO TENIENDO EN CUENTA FUTUROS AVANCES EN LA TECNOLOGÍA DE DISPOSITIVOS DE USUARIO, COMO LA PULSERA GPS. LA CAPACIDAD DE ADAPTARSE A CAMBIOS EN LA RECEPCIÓN DE DATOS ES CRUCIAL PARA MANTENER LA RELEVANCIA Y LA EFECTIVIDAD DEL SISTEMA. LA ESTRUCTURA DE MICROSERVICIOS PERMITE LA FÁCIL INTEGRACIÓN DE NUEVAS FUENTES DE DATOS, ASEGURANDO QUE HAWKEYE PUEDA EVOLUCIONAR JUNTO CON LAS INNOVACIONES EN TECNOLOGÍA WEARABLE.

EN CONCLUSIÓN, EL DESARROLLO DEL BACKEND DE HAWKEYE NO SOLO HA SIDO UNA HAZAÑA TÉCNICA, SINO UNA ESTRATEGIA CUIDADOSA PARA CONSTRUIR UNA SOLUCIÓN SÓLIDA Y ADAPTABLE. LA ARQUITECTURA, LAS HERRAMIENTAS Y LA CAPACIDAD DE ESCALABILIDAD INCORPORADAS EN EL SISTEMA NO SOLO HAN CUMPLIDO

HawkEye Backend: Seguridad Personal en Tiempo Real

CON LOS REQUISITOS ACTUALES, SINO QUE TAMBIÉN HAN SENTADO LAS BASES PARA EL CRECIMIENTO FUTURO Y LA EVOLUCIÓN DEL SISTEMA. LA ATENCIÓN A ESTOS ASPECTOS TÉCNICOS GARANTIZA QUE HAWKEYE NO SOLO SEA UNA RESPUESTA A LAS NECESIDADES ACTUALES DE SEGURIDAD PERSONAL, SINO UNA SOLUCIÓN QUE PUEDE CRECER Y EVOLUCIONAR CON LAS CAMBIANTES DEMANDAS TECNOLÓGICAS Y DE LA SOCIEDAD.

6. REFERENCIAS BIBLIOGRÁFICAS

- [1]AIM Chile. "AIM Chile", Innata Media, Disponible en: <https://aimchile.cl/>. [Accedido 27 Noviembre. 2023].
- [2]GitHub, Inc. "GitHub Docs", Docs, Disponible en: <https://docs.github.com/es>. [Accedido 27 Noviembre. 2023].
- [3]Google for Developers. "Documentación de Google Maps Platform", Developers, Disponible en: <https://developers.google.com/maps/documentation?hl=es-419>. [Accedido 27 Noviembre. 2023].
- [4]Google for Developers. "Documentación para desarrolladores de apps", Developer, Disponible en: <https://developer.android.com/docs?hl=es-419>. [Accedido 27 Noviembre. 2023].
- [5]INNATA MEDIA. "AIM Chile", Aimchile, Disponible en: <https://aimchile.cl/>. [Accedido 27 Noviembre. 2023].
- [6]JET BRAINS. "IntelliJ IDEA: el IDE líder para Java y Kotlin", JetBrains, Disponible en: <https://jetbrains.com/es-es/idea/>. [Accedido 27 Noviembre. 2023].
- [7]JSONCRACK. "Documentation - JSON Crack", Jsoncrack, Disponible en: <https://jsoncrack.com/docs>. [Accedido 27 Noviembre. 2023].
- [8]LA TERCERA. "¿Cuántos niños perdidos hay en Chile? El drama crece cuando las cifras no calzan - La Tercera", LA TERCERA, 20 Mar. 2021. Disponible en: <https://latercera.com/la-tercera-domingo/noticia/cuantos-ninos-perdidos-hay-en-chile-el-drama-crece-cuando-las-cifras-no-calzan/IZGFPVY4BVGD7FQ4EOOW6AM5XQ/>. [Accedido 27 Noviembre. 2023].
- [9]MondoDB, Inc. "*What is MongoDB Atlas? — MongoDB Atlas*", Mongodb, Disponible en: <https://mongodb.com/docs/atlas/>. [Accedido 27 Noviembre. 2023].
- [10]Okta, Inc. "Guía del Comprador de CIAM", Auth0, Disponible en: https://auth0.com/resources/whitepapers/Guia-del-Comprador-de-CIAM?utm_content=latameschilebrandauth0-pure_brand-esciambuyersguide. [Accedido 27 Noviembre. 2023].
- [11]Okta, Inc. "Guía del Comprador de CIAM", Auth0, Disponible en: https://auth0.com/resources/whitepapers/Guia-del-Comprador-de-CIAM?utm_content=latameschilebrandauth0-pure_brand-esciambuyersguide. [Accedido 27 Noviembre. 2023].
- [12]PDICHILE. "Personas extraviadas", Pdichile, Disponible en: <https://pdichile.cl/pdi-busca/personas-extraviadas>. [Accedido 27 Noviembre. 2023].

[13]Postman, Inc. "API Tools | Postman API Platform", Postman, Disponible en: <https://postman.com/product/tools/>. [Accedido 27 Noviembre. 2023].

[14]"Población: Chile 2021 - PopulationPyramid.net", Populationpyramid, Disponible en: <https://populationpyramid.net/es/chile/2021/>. [Accedido 27 Noviembre. 2023].

[15]Railway Corp. "Railway", Railway, Disponible en: <https://railway.app/>. [Accedido 27 Noviembre. 2023].

[16]Red Hat, Inc. "REST API", Red Hat, 31 May. 2023. Disponible en: <https://redhat.com/es/topics/api/what-is-a-rest-api>. [Accedido 27 Noviembre. 2023].

[17]VMware Tanzu. "Spring Boot", Spring, Disponible en: <https://spring.io/projects/spring-boot>. [Accedido 27 Noviembre. 2023].

[18]"Los grupos socioeconómicos en Chile", Percepciones económicas, Disponible en: <https://percepcioneconomicas.cl/desarrollo-economico/los-grupos-socioeconomicos-en-chile/>. [Accedido 27 Noviembre. 2023].