

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Software

“Implementación de una aplicación móvil para la gestión de las actividades operativas de la empresa de seguridad privada Alpriseq Cía Ltda utilizando la geolocalización.”

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado ante la ilustre Universidad Técnica del Norte.

Línea de investigación:

Aplicaciones móviles y desarrollo de software.

Autor:

Ulises Iván Valencia López

Director:

Ing. Carpio Agapito Pineda Manosalvas, MSc

Ibarra – Ecuador

2025



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1005110737		
APELLIDOS Y NOMBRES:	Valencia López Ulises Iván		
DIRECCIÓN:	Ibarra, Urbanización FECOMI		
EMAIL:	uivalencial@utn.edu.ec		
TELÉFONO FIJO:	2585161	TELÉFONO MÓVIL:	0984600865

DATOS DE LA OBRA	
TÍTULO:	Implementación de una aplicación móvil para la gestión de las actividades operativas de la empresa de seguridad privada Alpriseg Cía Ltda utilizando la geolocalización.
AUTOR(ES):	Valencia López Ulises Iván
FECHA: DD/MM/AAAA	22/07/2025
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Software
DIRECTOR:	ING. Carpio Agapito Pineda Manosalvas, MSC.
ASESOR 1:	ING. Pablo Andres Landeta López, MSC.

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 22 días del mes de julio de 2025

EL AUTOR:



ESTUDIANTE

Ulises Iván Valencia López

C.I 1005110737

CERTIFICACIÓN DIRECTOR

Ibarra 22 de julio del 2025

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo Ing. Carpio Agapito Pineda Manosalvas, MSc., certifico que el Sr. Ulises Iván Valencia López portador de la cedula de ciudadanía número 1005110737, ha trabajado en el desarrollo del proyecto de grado **“Implementación de una aplicación móvil para la gestión de las actividades operativas de la empresa de seguridad privada Alpriseg Cía Ltda utilizando la geolocalización.”**, previo a la obtención del Título de Ingeniero en Software realizado con interés profesional y responsabilidad que certifico con honor de verdad.

Es todo en cuanto puedo certificar a la verdad

Atentamente

Ing. Carpio Agapito Pineda Manosalvas, MSc.

DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

A mis queridos padres, Cándida López e Iván Valencia, quienes han sido un ejemplo de perseverancia y valentía ante las adversidades. A mis hermanos, Alexander Valencia, hermano y ejemplo a seguir, que a pesar de su juventud ha sido un modelo de padre y líder; a pesar de que todo estaba en su contra, me demostró que siempre es posible seguir adelante. Y a ti, hermano Mateo Valencia, que eres valioso para toda la familia y que, sin importar lo que pase, siempre nos apoyarás. Ustedes son la razón por la que sigo persiguiendo este sueño, y mis logros son para ustedes. También dedico este esfuerzo a todas las personas que tuvieron la voluntad de darme fuerzas para continuar, a mis amigos, primos y maestros. Este trabajo es por y para ustedes.

Ulises Iván Valencia López

AGRADECIMIENTO

En primer lugar, agradezco a Dios por darme la fortaleza y la sabiduría necesarias para alcanzar este objetivo.

A mis padres, Cándida López e Iván Valencia, por ser mi pilar fundamental, por su amor incondicional, su ejemplo de vida y por enseñarme el valor del esfuerzo y la perseverancia. Gracias por creer en mí incluso en los momentos más difíciles y por apoyarme en cada paso del camino.

A mis hermanos, Alexander y Mateo Valencia, por ser una fuente constante de inspiración. Alexander, tu liderazgo y dedicación me han mostrado lo que significa ser un verdadero ejemplo, y Mateo, tu apoyo incondicional ha sido un pilar importante en mi vida. Estoy profundamente agradecido por tenerlos a ambos a mi lado.

A mis amigos, que siempre estuvieron dispuestos a darme palabras de ánimo cuando las dudas me asaltaban, y a mis primos, quienes con su compañía y apoyo me dieron la energía para seguir adelante.

Quiero extender un agradecimiento especial a mis maestros, por compartir sus conocimientos y por su paciencia y guía a lo largo de mi formación. Su dedicación fue clave para que pudiera culminar este proyecto.

Finalmente, gracias a todas aquellas personas que, de manera directa o indirecta, contribuyeron a este logro. Sin su apoyo y confianza, no habría sido posible alcanzar esta meta. Este triunfo es también de ustedes.

Ulises Iván Valencia López

TABLA DE CONTENIDOS

DEDICATORIA.....	3
AGRADECIMIENTO.....	4
TABLA DE CONTENIDOS.....	5
ÍNDICE DE FIGURAS.....	8
ÍNDICE DE TABLAS.....	12
RESUMEN.....	13
ABSTRACT.....	14
INTRODUCCIÓN.....	15
Tema.....	15
Problema.....	15
Antecedentes.....	15
Situación Actual.....	15
Prospectiva.....	15
Planteamiento del problema.....	16
Objetivos.....	17
Objetivo General.....	17
Objetivos Específicos.....	17
Alcance.....	17
Metodología.....	20
Justificación.....	22
<i>Justificación Tecnológica</i>	22
<i>Justificación Operativa</i>	23
<i>Justificación Social</i>	23
CAPÍTULO 1.....	24
Marco Teórico.....	24
1.1. Lineamientos y reglamentos del personal de seguridad.....	24

1.1.1.	Requisitos legales.....	24
1.1.2.	Normativas de seguridad privada	25
1.1.3.	Procedimientos y Responsabilidades.....	26
1.2.	Fundamentos de la metodología Scrum.....	28
1.2.1.	Introducción a Scrum.....	28
1.2.2.	Eventos en Scrum.....	30
1.2.3.	Beneficios de Scrum en el desarrollo de aplicativos móviles.	31
1.3.	Pruebas de estrés.....	32
1.3.1.	Definición y objetivos de las pruebas de estrés.....	32
1.3.2.	Métodos para realizar pruebas de estrés.	34
1.3.3.	Funcionalidades GPS.	36
1.3.4.	Herramientas para pruebas de estrés en aplicaciones móviles.	37
1.4.	Norma ISO/IEC 25023 disponibilidad.	38
1.4.1.	Introducción a la norma ISO/IEC 25023.	38
1.4.2.	Indicadores de calidad de disponibilidad.	40
1.4.3.	Métricas clave para evaluar la disponibilidad en sistemas.	42
1.5.	Desarrollo de Aplicaciones Móviles con GPS.....	43
1.5.1.	Definición y objetivos del uso de GPS en aplicaciones móviles.	43
1.5.2.	Componentes de Expo para la geolocalización.....	44
1.5.3.	Usos del GPS en la gestión operativa.	45
CAPÍTULO 2		48
Desarrollo del proyecto		48
2.1	Planificación del proyecto.....	48
2.1.1	Historias de usuario.	48
2.1.2	Product Backlog.....	52
2.1.3	Creación de Sprints.	55
2.2.	Diseño.....	55
2.2.1.	Arquitectura del sistema.	55

2.2.2.	Diagramas de clases.	57
2.2.3.	Prototipo del Administrador web	57
2.2.4.	Prototipo de la aplicación móvil	63
2.3.	Desarrollo del sistema	67
2.3.1.	Módulos del proyecto.....	67
2.3.2.	Sprint 0	68
2.3.3.	Sprint 1	89
CAPÍTULO 3		106
Validación de resultados		106
3.1	Metodología de Evaluación.....	106
3.1.1	Evaluación mediante bitácora manual.	106
3.1.2	Evaluación mediante pruebas de estrés.....	111
3.2	Análisis e interpretación de resultados.	123
CONCLUSIONES.....		124
RECOMENDACIONES		125
BIBLIOGRAFÍA		125
ANEXOS		132

ÍNDICE DE FIGURAS

Figura 1 Árbol de problemas.....	16
Figura 2 <i>Gráfica de representación del alcance</i>	20
Figura 3 <i>Metodología de desarrollo</i>	21
Figura 4 Regulaciones de Guardias de Seguridad.	25
Figura 5 Diagrama del ciclo de un Sprint en Scrum.....	29
Figura 6 Prioridad de los aspectos clave en las pruebas de estrés.	34
Figura 7 Relación entre MTBF, MTTR y Disponibilidad del Sistema.	41
Figura 8 Permisos de localización Android y IOS.	45
Figura 9 Monitoreo de rutas en trackforce.	46
Figura 10 Reportes de rutas en Silvertrac.	47
Figura 11 Priorización de MoSCoW.....	53
Figura 12 Arquitectura general del sistema.	56
Figura 13 Diagrama de clases.	57
Figura 14 Prototipo “Pantalla de Login”.	58
Figura 15 Prototipo “Pantalla principal”......	58
Figura 16 Prototipo “Vista de registrar usuarios”.....	59
Figura 17 Prototipo “Pantalla de Lista de Usuarios”.....	59
Figura 18 Prototipo “Pantalla de Registrar Puesto”.....	60
Figura 19 Prototipo “Pantalla de Lista de Puestos”.....	60
Figura 20 Prototipo “Pantalla de Asignar Puestos”.	61
Figura 21 Prototipo “Pantalla de Historial de asignación”.	61
Figura 22 Prototipo “Pantalla de Control de turnos”.....	62
Figura 23 Prototipo “Pantalla de Multas”.....	62
Figura 24 Prototipo “Pantalla de Incidentes”.....	63
Figura 25 Prototipo móvil - Pantalla de Login	63

Figura 26	Prototipo móvil - Pantalla de Ingreso del personal.....	64
Figura 27	Prototipo móvil - Pantalla principal en Administración de Incidentes.	65
Figura 28	Prototipo móvil - Pantalla principal en Evidencia del Puesto.	65
Figura 29	Prototipo móvil - Pantalla principal en Administración de multas.	66
Figura 30	Prototipo móvil - Pantalla principal en Generación de reportes.	66
Figura 31	Prototipo móvil - Pantalla principal en Control de Alertas.	67
Figura 32	Base de datos con sus colecciones.	70
Figura 33	Evidencia guardada en Firestore.	72
Figura 34	Configuración para la conexión a la BDD.	72
Figura 35	Configuración para la conexión y subida a Firestore.....	73
Figura 36	Configuración para la implementación de la API de Google Maps.	73
Figura 37	API Rest método register.....	74
Figura 38	Registro de nuevo usuario en el módulo de Administración.	75
Figura 39	API Rest método usuario-consulta.	76
Figura 40	Listado de los usuarios registrados en el Front.	76
Figura 41	Revocar permisos a un usuario.	77
Figura 42	Pantalla de edición de un usuario.....	77
Figura 43	Registro de nuevos puestos.	78
Figura 44	API Rest método POST puestos.	79
Figura 45	API Rest método GET puestos.....	80
Figura 46	Pantalla de listado de puestos.....	80
Figura 47	Pantalla de ver ubicación.....	81
Figura 48	API Rest método GET asignaciones.....	82
Figura 49	Pantalla de la gestión de asignaciones.....	83
Figura 50	Pantalla de la historial de asignaciones.....	83
Figura 51	Pantalla de detalles de asignación.	84
Figura 52	API Rest método GET registro-consulta.....	85

Figura 53 Pantalla de evidencias-control turnos.	86
Figura 54 API Rest método GET multas-consulta.	86
Figura 55 Pantalla de evidencias-multas	87
Figura 56 API Rest método GET incidentes-consulta.	87
Figura 57 Pantalla de evidencias-incidentes.	88
Figura 58 API Rest método POST login.	91
Figura 59 Método para calcular el perímetro para el ingreso.	91
Figura 60 Pantalla de login en el dispositivo móvil.	92
Figura 61 Pantalla de registro de turno.	93
Figura 62 Método para subir datos de registro de turno.	94
Figura 63 Método para crear alertas en caso de llegar tarde.	94
Figura 64 Generación de alertas por llegar tarde.	95
Figura 65 Pantalla de administración de alertas por llegar tarde.	96
Figura 66 Pantalla de incidentes.	97
Figura 67 Método para crear incidentes.	97
Figura 68 Pantalla evidencias.	98
Figura 69 Pantalla de visualización del mapa de la evidencia.	99
Figura 70 Pantalla de creación de multas.	100
Figura 71 Método para crear multas.	100
Figura 72 Ejemplo de una solicitud de memo enviada al correo.	101
Figura 73 Pantalla de reportes.	102
Figura 74 Método para crear reportes.	103
Figura 75 Reporte en formato xlsx.	104
Figura 76 Pantalla de inicio de guardias.	105
Figura 77 Datos requeridos en la bitácora.	106
Figura 78 Visualización de los datos de la bitácora manual.	108
Figura 79 Resultados obtenidos de las métricas.	109

Figura 80 Endpoints a evaluar en las pruebas de estrés.....	112
Figura 81 Creación del Environment SegGuardias en Postman.....	113
Figura 82 Creación del Script de token.....	113
Figura 83 Environment SegGuardias.....	114
Figura 84 Colección Pruebas Estrés – GET.....	114
Figura 85 Colección Pruebas Estrés – POST.....	115
Figura 86 Colección PRUEBAS TESIS.....	115
Figura 87 Configuración de Runner para la colección de tipo GET.....	116
Figura 88 Resultado obtenido del Runner para la colección tipo GET.....	116
Figura 89 Configuración de Runner para la colección de tipo POST Login.....	117
Figura 90 Resultado obtenido del Runner para la colección tipo POST Login.....	117
Figura 91 Configuración de Runner para la colección de tipo POST.....	118
Figura 92 Resultado obtenido del Runner para la colección tipo POST.....	119
Figura 93 Pruebas de concurrencia con cinco usuarios en endpoints GET.....	119
Figura 94 Resultados de las iteraciones del tipo GET.....	120
Figura 95 Pruebas de concurrencia con cinco usuarios en endpoints POST.....	121
Figura 96 Resultados de las iteraciones del tipo POST.....	121
Figura 97 Resultados de las métricas obtenidos.....	123

ÍNDICE DE TABLAS

Tabla 1 Procedimientos y Responsabilidades de las Empresas de Seguridad Privada en Ecuador.	27
Tabla 2 Historia de usuario 1 – Control de turnos de guardias de seguridad.	48
Tabla 3 Historia de usuario 2 – Gestión del registro de personal activo.	49
Tabla 4 Historia de usuario 3 – Gestión del puesto de guardia.	50
Tabla 5 Historia de usuario 4 – Gestión de los puestos a supervisar.	51
Tabla 6 Estimación de esfuerzo con Story Points	54
Tabla 7 Product Backlog.	54
Tabla 8 Proyección de los Sprints.	55
Tabla 9 Módulos del proyecto	67
Tabla 10 Sprint 0	68
Tabla 11 Atributos de las colecciones de segguardias.	71
Tabla 12 Retrospectiva del Sprint 0	88
Tabla 13 Sprint 1	89
Tabla 14 Retrospectiva del Sprint 1	105

RESUMEN

Este documento incluye 3 capítulos que describen detalladamente el proceso de implementación de un proyecto de titulación: "Implementación de una aplicación móvil para la gestión de las actividades operativas de la empresa de seguridad privada Alpriseq Cía Ltda utilizando la geolocalización."

En la introducción, se definen los antecedentes del sector de seguridad privada, la situación actual de Alpriseq Cía Ltda, la prospectiva, identificar el problema, objetivos generales y específicos, alcance y justificación del proyecto.

En el apartado 1, se desenvuelve el marco teórico, abarcando temas como los lineamientos y reglamentos del personal de seguridad en Ecuador, los requisitos legales, los procedimientos operativos de las empresas de seguridad privada, la metodología ágil Scrum aplicada al desarrollo de la aplicación móvil, y la norma ISO/IEC 25023 para evaluar la disponibilidad del sistema.

En la sección 2, se describe la proyección del proyecto, incluyendo las historias de usuario, la estimación de esfuerzo, la priorización de tareas, y la creación del Product Backlog. Además, se detallan los sprints y las tareas propuestas como el desarrollo del administrador web, el API REST y la aplicación móvil.

Posteriormente, en el tercer apartado, se presentan los efectos de la investigación, con el análisis e interpretación de los juicios definidos, destacando el cumplimiento de los objetivos y los criterios a evaluar de la disponibilidad del sistema en base a la ISO/IEC 25023.

Por último, se encuentran conclusiones, recomendaciones, referencias y apéndices.

Palabras clave: : Aplicativo móvil, geolocalización, fórmula de Haversine, pruebas de estrés, React Native, ISO/IEC 25023

ABSTRACT

This document includes 3 chapters that detail the implementation process of a degree project: "Implementation of a mobile application for managing the operational activities of the private security company Alpriseg Cía Ltda using geolocation."

In the introduction, the background of the private security sector, the current situation of Alpriseg Cía Ltda, future prospects, problem identification, general and specific objectives, scope, and justification of the project are defined.

In Chapter 1, the theoretical framework is developed, covering topics such as the guidelines and regulations for security personnel in Ecuador, legal requirements, operational procedures for private security companies, the agile Scrum methodology applied to the development of the mobile application, and the ISO/IEC 25023 standard for evaluating system availability.

In Chapter 2, the project projection is described, including user stories, effort estimation, task prioritization, and the creation of the Product Backlog. Additionally, the sprints and proposed tasks are detailed, such as the development of the web administrator, the REST API, and the mobile application.

Subsequently, in Chapter 3, the results of the research are presented, including the analysis and interpretation of the defined judgments, highlighting the achievement of objectives and the criteria for evaluating system availability based on the ISO/IEC 25023 standard.

Finally, the document includes conclusions, recommendations, references, and appendices.

Keywords: Mobile application, geolocation, Haversine formula, stress testing, React Native, ISO/IEC 25023

INTRODUCCIÓN

Tema

Implementación de una aplicación móvil para la gestión de las actividades operativas de la empresa de seguridad privada Alpriseg Cía Ltda utilizando la geolocalización.

Problema

Antecedentes

Existen cuatro modalidades destacadas en los servicios de protección y resguardo en el ámbito privado. Dependiendo de cada modalidad, adquieren obligaciones y características específicas o compartidas con el resto de las modalidades. En Ecuador, las más comunes son la Vigilancia Móvil y la Vigilancia Fija (Cando Villa, 2020).

La Vigilancia Fija es un servicio de seguridad constante en un lugar específico para proteger personas, entidades y todo tipo de bienes (Carcelén Gutiérrez, 2022).

Situación Actual

La organización dedicada a brindar servicios de protección y vigilancia. privada Alpriseg Cía Ltda, ubicada en Ibarra, actualmente desarrolla sus actividades en diversas provincias a nivel nacional. Enfrenta un desafío común en este sector, que es la gestión del personal. Cuenta con alrededor de 60 puestos activos, repartidos entre 165 guardias en servicio y 4 supervisores que trabajan bajo la modalidad de Vigilancia Fija. Estos supervisores deben desplazarse frecuentemente a distintas provincias, lo que a veces les obliga a rotar de forma apresurada o a cambiar sus turnos debido a incidentes surgidos en algún puesto que está lejos de su ubicación.

Prospectiva

En el contexto nacional, se exige que los empleados de las empresas de seguridad privada utilicen correctamente el uniforme y los emblemas autorizados por el Ministerio de

Gobierno y Policía, y que estén presentes en el área asignada; sin embargo, en varios casos, estos requisitos no se cumplen (Servicio nacional de contratación pública, 2022).

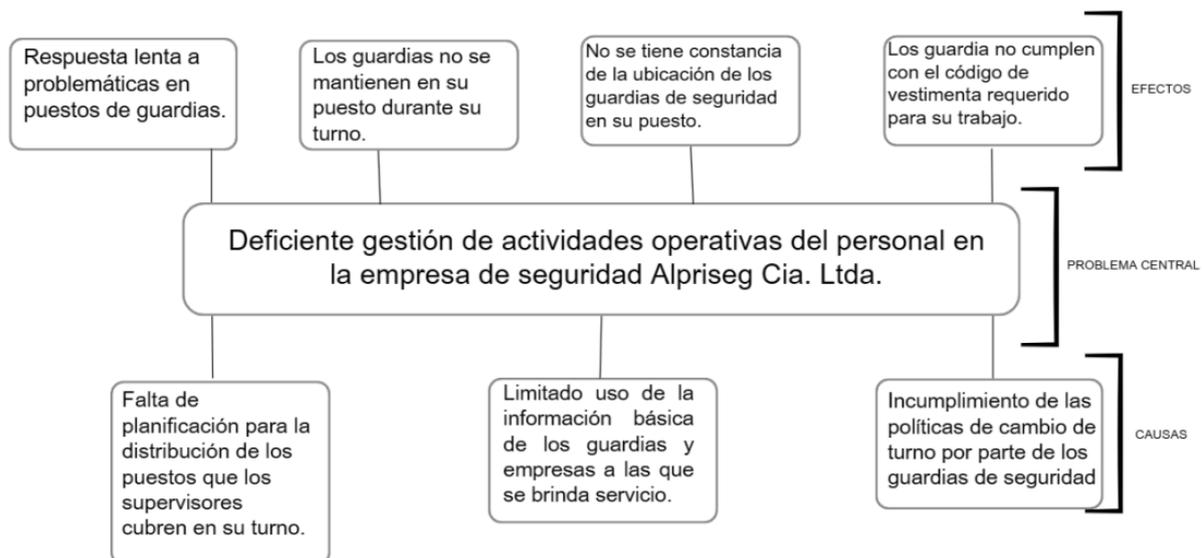
Planteamiento del problema

Esta empresa, por lo antes mencionado, presenta información errónea sobre la ubicación de los guardias, problemas con el uniforme de estos y con los cambios de turnos diurnos y nocturnos, además de la falta de evidencia al momento de la revisión de informes diarios y mensuales.

La falta de este control de los guardias ha provocado retrasos en los informes diarios por puesto activo, lo cual a su vez causa que no se puedan identificar problemas en los puestos y el personal administrativo no pueda aplicar acciones, especialmente en casos en los que se requiere una solución rápida.

Figura 1

Árbol de problemas



Nota. Autoría propia.

Objetivos

Objetivo General.

Implementar una aplicación móvil para la gestión de las actividades operativas de la empresa de seguridad privada Alpriseg Cía Ltda usando la geolocalización.

Objetivos Específicos

- Elaborar un marco teórico referente a técnicas de elaboración de apps móviles para uso de GPS en Android que apoye el desarrollo de la aplicación móvil para la gestión de la empresa privada.
- Desarrollar una aplicación móvil utilizando Scrum que gestione el control de turnos, el registro de novedades y multas, además de generar informes diarios y mensuales para la empresa.
- Validar la aplicación móvil basándose en la subcaracterística de la disponibilidad del estándar ISO/IEC 25023.

Alcance

El alcance de esta investigación se enfoca en la creación de una aplicación móvil para dispositivos Android, destinada a la empresa de seguridad Alpriseg Cía Ltda, con el objetivo de mejorar el control eficiente del personal de seguridad. Una aplicación móvil para el control del personal por su portabilidad ayudará a evitar errores en la información y proporciona un mayor control por parte de los supervisores (Acosta Flores, 2019).

Se empleará la metodología ágil Scrum para llevar a cabo el diseño y desarrollo de la aplicación. Permitirá un desarrollo iterativo y ágil, facilitando la adaptabilidad y la entrega continua de valor (Ramírez Ramírez et al., 2018).

La aplicación utilizará MongoDB como sistema de gestión de base de datos. La cual será diseñada e implementada para almacenar y gestionar de manera eficiente la

información generada y requerida por la aplicación. Debido a que se destaca por ser orientada a documentos, lo cual implica que los datos son almacenados en forma de documentos flexibles que se asemejan al formato JSON (Chauhan, 2019).

El proyecto implicará el diseño y desarrollo de una aplicación móvil utilizando React Native que, es un framework orientado al desarrollo frontend de aplicaciones móviles. "React Native facilita a los programadores la creación de aplicaciones nativas para las plataformas Android y iOS, utilizando JavaScript y React" (Drewbatgit, 2022) y NodeJS que es "utilizado en el desarrollo del backend, siendo un entorno de ejecución de JavaScript específicamente diseñado para ello" (Node.js, n.d.). La aplicación permitirá la automatización de los procesos de control de turnos de los guardias de seguridad, tomando la geolocalización, una fotografía y la hora actual como evidencia, además del control de los supervisores utilizando la información obtenida de los guardias, para agregar multas en caso de ser necesario y para generar informes diarios en formato.xlsx los cuales se harán llegar a la empresa por medio de un correo electrónico.

El proyecto constará de dos aplicaciones: una aplicación móvil principal y un backend con un administrador web para tareas administrativas específicas.

En el módulo de administración del backend, se realizarán las siguientes funciones: Mostrar, crear, editar y cambiar el estado de los guardias y supervisores, incluyendo la gestión de sus claves de acceso para el módulo de acceso. Mostrar, crear y cambiar el estado de los puestos de los guardias, así como permitir su asignación a guardias y supervisores. Generar informes de historial de asignación de puestos para guardias y supervisores.

Por otro lado, en la aplicación móvil se encontrarán los siguientes módulos: Módulo de "Acceso": Permitirá a los usuarios (guardias, supervisores y administradores) ingresar a la aplicación. Cada usuario tendrá su propio perfil con

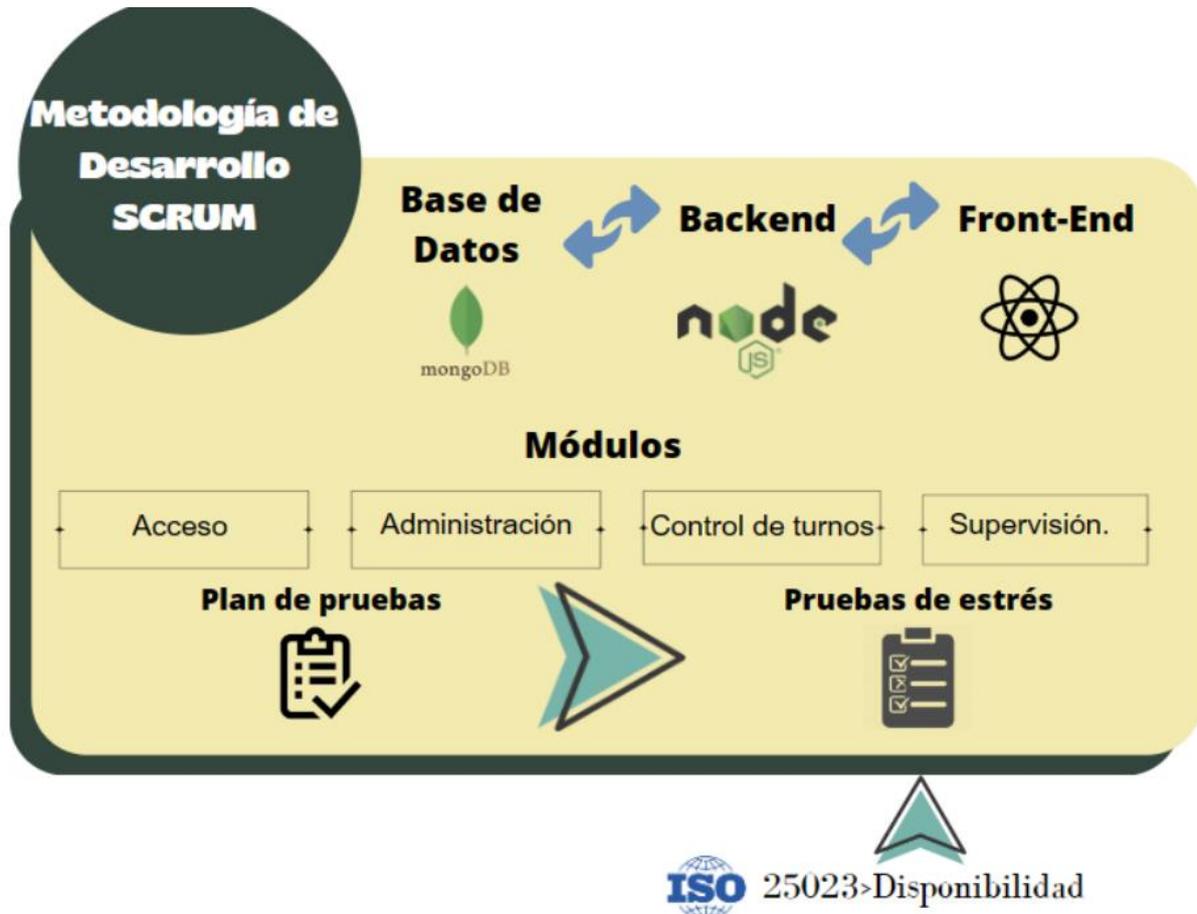
privilegios específicos. Módulo de "Control de turnos": Los guardias podrán registrar sus turnos utilizando una función que les permitirá capturar una foto como evidencia. Además, podrán documentar cualquier incidencia o suceso relevante ocurrido durante su turno para una gestión eficiente de eventos. Módulo de "Supervisión": Este módulo estará disponible para los supervisores, quienes podrán acceder a la información de los turnos de todos los guardias asignados a su puesto. Además, tendrán la capacidad de aplicar multas si es necesario, registrar incidentes y generar reportes diarios que se enviarán automáticamente al correo electrónico de la empresa.

Con estas dos aplicaciones interconectadas, se logrará una solución completa para la gestión de actividades de la empresa de seguridad privada, abarcando tanto aspectos administrativos como operativos, y proporcionando herramientas eficientes para el control y seguimiento de los turnos y eventos.

Como parte del proceso de desarrollo, se realizará una evaluación de la disponibilidad de la aplicación móvil. Se basará en la disponibilidad del estándar ISO/IEC 25023 el cual "su fundamento radica en la capacidad del sistema para operar y estar disponible en todo momento que sea requerido" (Zarour, 2020). Se realizará una prueba de estrés para verificar su disponibilidad y realizar correcciones pertinentes.

Figura 2

Gráfica de representación del alcance.



Nota. Autoría propia.

Metodología

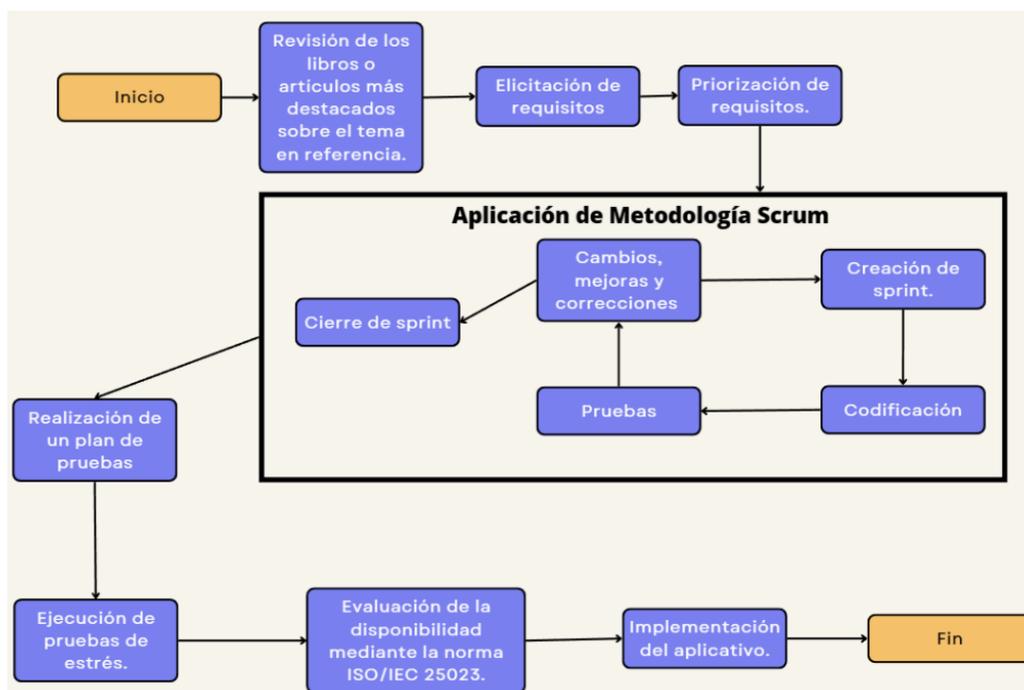
Para el primer objetivo, se llevará a cabo una búsqueda exhaustiva de libros o artículos que hagan referencia al tema o que aborden tópicos que puedan aportar al desarrollo de aplicaciones móviles en Android, que utilicen la herramienta del GPS para funciones operativas. La búsqueda propuesta ayudará a comprender aspectos importantes en la investigación, contribuirá a entender mejor las variables existentes y permitirá alcanzar conclusiones viables que solucionen la problemática (Arévalo Chávez et al., 2020). El marco teórico resultante proporcionará una base sólida de conocimientos y orientación para la creación de la aplicación móvil enfocada a la gestión de la empresa.

Para alcanzar el segundo objetivo, se llevará a cabo una elicitación de requisitos, "lo cual permitirá comprender las necesidades del proyecto y el camino por el cual se puede

orientar el desarrollo de software” (Lim et al., 2021). Además, se establecerá una priorización de estos requisitos para asegurar que los más esenciales sean atendidos antes que otros, “se llevará a cabo considerando diversas variables, determinadas por las necesidades y el contexto particular de las organizaciones” (Martínez et al., 2020). El desarrollo del software se hará utilizando la metodología Scrum, la cual, “ofrece la flexibilidad necesaria para incorporar cambios, mejoras y correcciones en las iteraciones subsiguientes” (Loaza Granda, 2019). Con el fin de alcanzar el tercer objetivo, se elaborará y se implementará un plan detallado de pruebas de estrés. Este plan definirá los escenarios de estrés, los criterios de éxito y las herramientas a emplear. Posteriormente, se ejecutará la prueba que tiene, “el propósito de sobrecargar el programa hasta un punto crítico donde puedan surgir errores potencialmente dañinos” (Contreras Mejia, 2019) y se contrastarán los resultados obtenidos con los criterios de éxito establecidos, para confirmar si la aplicación satisface los requisitos de disponibilidad dictados por el estándar ISO/IEC 25023. Todo el proceso y sus resultados se documentarán de manera adecuada.

Figura 3

Metodología de desarrollo



Nota. Autoría propia.

Justificación

Este proyecto se concentra en gran medida en el Objetivo de Desarrollo Sostenible 9: Fomentar la industria, la innovación y la infraestructura. Debido que al poner en marcha esta propuesta tecnológica, se estará mejorando la infraestructura digital, impulsando la innovación y apoyando un tipo de industrialización que sea tanto inclusiva como sostenible. Adicionalmente, este proyecto podría incidir de manera indirecta en el Objetivo de Desarrollo Sostenible 8: “Promover el trabajo decente y el crecimiento económico” (Naciones Unidas, 2019). Si se considera que el aumento de eficiencia operativa por el aplicativo, “cumple con el ODS 8 debido a que puede dar lugar a mejores condiciones laborales y a un crecimiento económico más estable a largo plazo” (Naciones Unidas, 2019).

Además, está alineado con el Objetivo 9 “El Plan de Creación de Oportunidades 2021-2025”, que busca impulsar el desarrollo sostenible y mejorar las condiciones de vida mediante políticas enfocadas en la equidad, la inclusión y el progreso económico. Mediante el desarrollo de la aplicación se pretende optimizar la gestión de actividades y mejorar la respuesta ante situaciones de riesgo gracias a la geolocalización. Esta innovación en las operaciones de seguridad privada tiene el potencial de contribuir a la seguridad ciudadana en general (Secretaría Nacional de Planificación, 2021).

Justificación Tecnológica

El proyecto tiene como objetivo principal optimizar los procesos de control del personal de seguridad y automatizar diversas operaciones inherentes a la empresa Alpriseg Cía Ltda. Para lograr esto, se desarrollará una aplicación móvil que utilizará la geolocalización como evidencia para automatizar dichos procesos y realizarlos de manera más eficiente. Esto permitirá brindar un mejor servicio a la comunidad. Además, se utilizará como guía la disponibilidad de la norma ISO/IEC 25023, aplicando pruebas de estrés como evidencia para su validación.

Justificación Operativa

La aplicación móvil diseñada para Alpriseg Cía Ltda permitirá una gestión más eficiente de las actividades de la empresa. Mediante el uso de la geolocalización, la aplicación automatizará y agilizará los procesos de control del personal de seguridad, lo que se traducirá en una asignación de recursos más efectiva, un mejor manejo de incidentes y una respuesta más rápida en situaciones de emergencia. En consecuencia, se espera que esta mejora en las operaciones conduzca a un aumento en la calidad del servicio de seguridad proporcionado.

Justificación Social

La creación de una aplicación móvil para la dirección de Alpriseg Cía Ltda ayudará a la productividad de la empresa y generará un impacto social notable. Al perfeccionar la gestión de la seguridad, la aplicación permite actuaciones más efectivas y veloces, lo que se traduce en un entorno más seguro.

CAPÍTULO 1

Marco Teórico

1.1. Lineamientos y reglamentos del personal de seguridad.

1.1.1. Requisitos legales.

Requisitos Generales.

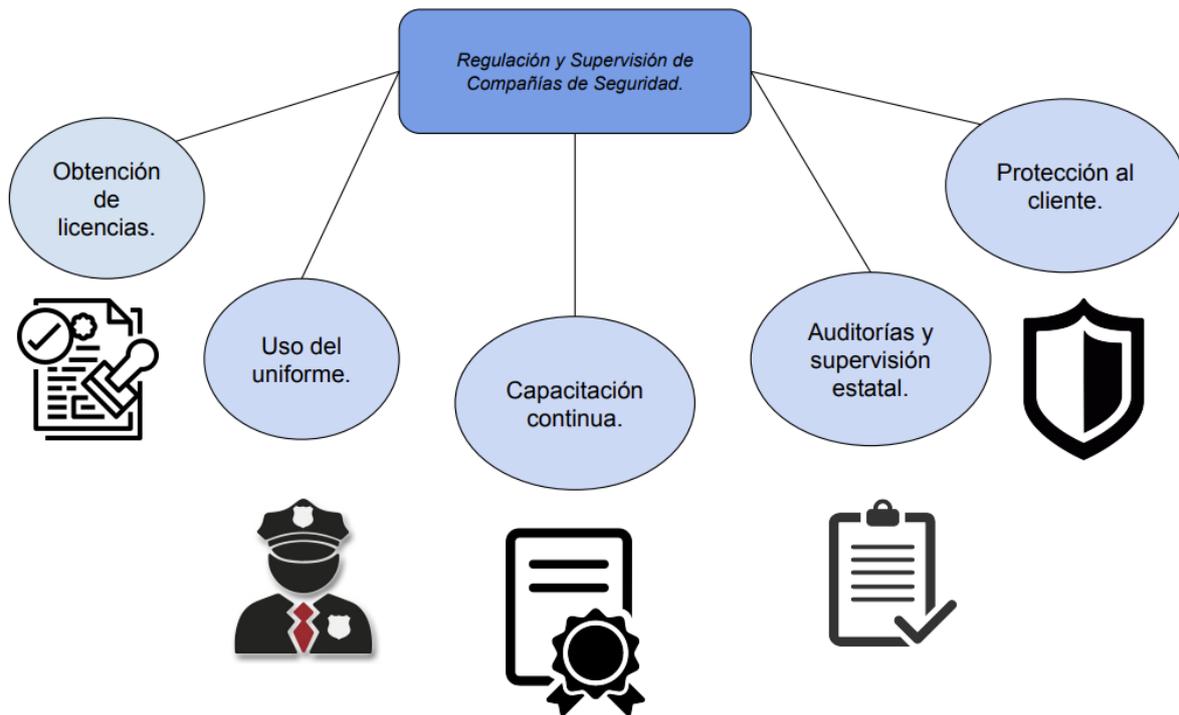
En Ecuador, los guardias de seguridad deben cumplir con lo dispuesto por la “*Ley de Vigilancia y Seguridad Privada*” (Gobierno del Ecuador, 2008). Entre estos requisitos se incluye ser ciudadano ecuatoriano o extranjero residente, ser mayor de 18 años y disponer de educación básica completa (Gobierno del Ecuador, 2008). Estos requisitos aseguran que los guardias tengan una base mínima de educación y sean legalmente reconocidos en el país.

Regulación y Supervisión.

Las compañías de seguridad privada están obligadas a cumplir con un conjunto de normativas diseñadas para garantizar la protección de los clientes y el cumplimiento legal de sus operaciones. Estos aspectos se muestran en la **Figura 4**, donde se incluyen el registro, la supervisión estatal, y el uso obligatorio del uniforme. Además, se podrían incluir la obligación de registrarse y obtener las licencias correspondientes para operar legalmente (Briones, 2024). La supervisión estatal garantiza que las empresas cumplan con las normativas y estándares establecidos, protegiendo así a los clientes y al público en general.

Figura 4

Regulaciones de Guardias de Seguridad.



Nota. Adaptado de (Gobierno del Ecuador, 2008).

Funciones y Responsabilidades.

Los guardias de seguridad tienen responsabilidades específicas que deben cumplir dentro del marco legal. Estas incluyen la protección de personas y bienes, la prevención de delitos, y la colaboración con las autoridades en situaciones de emergencia (Velásquez, 2018). Las funciones de los guardias están claramente definidas para asegurar que actúen dentro de los límites de la ley y con profesionalismo.

1.1.2. Normativas de seguridad privada

Marco Legal General.

Las regulaciones de seguridad privada en Ecuador están principalmente bajo el control de la normativa de seguridad privada. Esta proporciona una estructura legal para proporcionar servicios de vigilancia y seguridad destinados a entidades naturales y legales, además de proteger bienes muebles, inmuebles y valores. Las empresas privadas en el campo de la

supervisión y la seguridad deben ser debidamente registradas y cumplir con los requisitos estipulados en la ley (Gobierno del Ecuador, 2008).

En febrero de 2024, se oficializó en el Registro Oficial una nueva normativa, cuyo objetivo principal es establecer y regular el sistema antes establecido. Esta ley establece reglas claves para empresas que brindan servicios de monitoreo y seguridad privada, incluye requisitos esenciales para el trabajo y la profesionalización de los empleados (Congreso Nacional de Ecuador, 2024).

La reciente normativa de seguridad privada incluye la regulación oficial sobre el porte de armas por parte del personal dedicado a estas labores. Esta normativa busca modernizar y reconceptualizar el sector de seguridad privada en el país, garantizando una regulación más eficiente y adecuada a los tiempos actuales (Fernanda Araujo, 2023).

Finalmente, la normativa dispone que las empresas de seguridad privada deben constituirse como sociedades de responsabilidad limitada y establece un capital mínimo requerido para su creación. Además, define las responsabilidades de los administradores de estas compañías, incluyendo la obligación de garantizar los derechos laborales del personal (BORRMART, 2024).

1.1.3. Procedimientos y Responsabilidades.

Procedimientos.

Las empresas de seguridad privada en Ecuador deben cumplir con una serie de procedimientos y responsabilidades, que incluyen la obtención de licencias, capacitación continua, control interno y colaboración con las autoridades, como se resume en **Tabla 1**. Uno de los principales requisitos es la obtención de una licencia de funcionamiento, la cual es otorgada por el Ministerio del Interior. Esta licencia se renueva anualmente y requiere que las empresas presenten documentación que acredite su capacidad operativa y financiera (Reglamento de aplicación de la Ley de Vigilancia y Seguridad Privada, 2008).

Otro aspecto crucial es la implementación de sistemas de control interno. Las empresas deben llevar un registro detallado de todas sus operaciones, incluyendo la contratación de personal, la asignación de servicios y el manejo de equipos. Estos registros son auditados periódicamente por las autoridades para asegurar el cumplimiento de las normativas vigentes (Moreseg Cia. Ltda., 2024).

Tabla 1

Procedimientos y Responsabilidades de las Empresas de Seguridad Privada en Ecuador.

Aspecto	Descripción
Obtención de licencia	Las empresas deben obtener una licencia de funcionamiento otorgada por el Ministerio del Interior, renovada anualmente con documentación adecuada.
Capacitación continua	Formación obligatoria para los guardias en áreas como derechos humanos, uso adecuado de la fuerza y gestión de emergencias, con el objetivo de asegurar su profesionalismo.
Control interno	Implementación de sistemas de control que registren operaciones, contratación de personal, asignación de servicios y manejo de equipos.
Seguridad y bienestar	Asegurar condiciones laborales equitativas, proporcionar equipos de protección personal adecuados y cumplir con las normativas laborales y de seguridad vigentes.

Nota. Adaptado de (Congreso Nacional de Ecuador, 2024).

Responsabilidades.

Las compañías de seguridad son responsables de garantizar la seguridad y la salud de sus trabajadores. Esto incluye proporcionar equipos de protección personal adecuados, así como asegurar condiciones laborales justas y seguras. La normativa exige que las empresas cumplan con todas las disposiciones laborales y de seguridad ocupacional (Congreso Nacional de Ecuador, 2024).

Finalmente, las empresas están obligadas a trabajar de manera coordinada con las fuerzas de seguridad pública. Esta colaboración incluye la comunicación y coordinación en situaciones de emergencia, así como la participación en programas de prevención del delito.

La cooperación entre el sector privado y las fuerzas de seguridad pública es esencial para preservar la seguridad y el orden dentro de la comunidad. (Normativa Orgánica sobre Vigilancia y Seguridad Privada, 2024).

1.2. Fundamentos de la metodología Scrum

1.2.1. Introducción a Scrum.

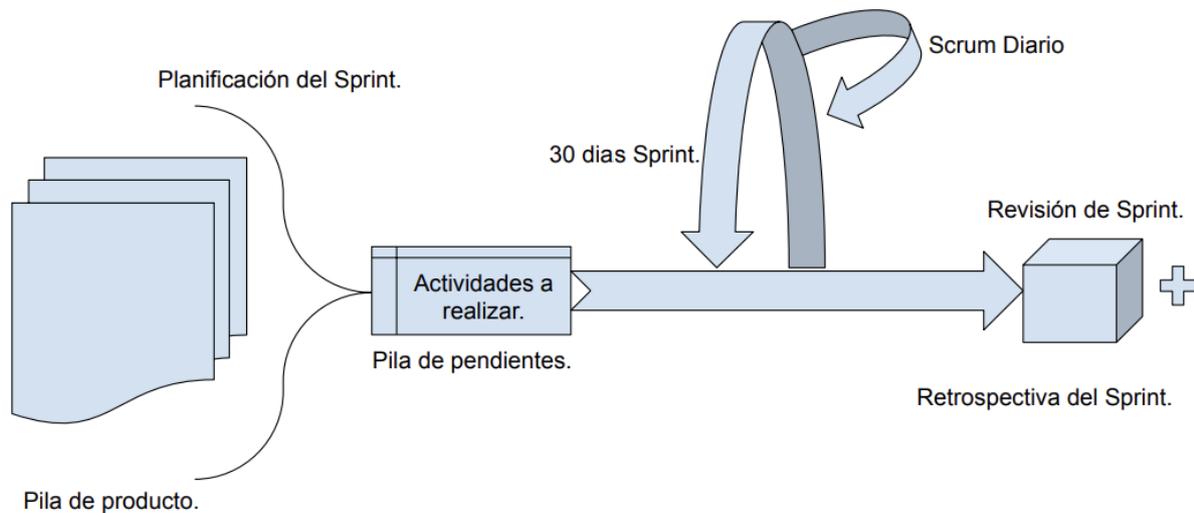
Concepto.

Scrum es un enfoque ágil diseñado para gestionar proyectos complejos, particularmente en el ámbito del desarrollo de software, utilizando ciclos iterativos conocidos como sprints, cada sprint culmina con un incremento del producto potencialmente entregable. Según Schwaber y Sutherland (2012), Scrum fomenta transparencia, verificación y adaptación, lo que permite a los grupos adaptarse rápidamente a los cambios y promover la mejora continua. Como se ilustra en la **Figura 5**, el ciclo de un sprint comprende eventos fundamentales que garantizan la entrega progresiva de productos.

Además, se fomenta la autoorganización de los equipos, permitiendo que los miembros del equipo decidan cómo abordar el trabajo y resolver problemas. Este enfoque se basa en la suposición de que equipos autoorganizados tienden a ser más efectivos y creativos (Verwijs, 2020). Un estudio de Verwijs (2020), encontró que los equipos autoorganizados en Scrum muestran una mayor capacidad para innovar y adaptarse a los cambios del entorno del proyecto.

Figura 5

Diagrama del ciclo de un Sprint en Scrum.



Nota. Adaptado de (Medina-Rojas et al., 2017).

Por ende, la colaboración y la comunicación efectiva son esenciales en Scrum. Los equipos Scrum trabajan juntos de manera estrecha y mantienen una comunicación constante para garantizar que las personas correspondan a los objetivos del sprint. Según un artículo de Verwijs (2021), la colaboración en equipos distribuidos de Scrum puede ser un desafío, pero con las prácticas adecuadas, se puede lograr una comunicación efectiva y una alta cohesión del equipo.

Principio de Control Empírico del Proceso.

Uno de los pilares esenciales de Scrum es el control empírico del proceso, sustentado sobre transparencia, verificación y adaptación. Este principio habilita para equipos, el desarrollo para ajustar sus procesos y productos con base en la retroalimentación constante. Según un estudio de Verwijs (2021), la implementación del control empírico en Scrum mejora significativamente la capacidad de los equipos para gestionar la incertidumbre y la complejidad en proyectos de software.

Priorización y Gestión del Backlog.

En Scrum, la priorización de tareas es crucial para asegurar que el equipo trabaje en los elementos más importantes primero. Los Product Owner son responsables de administrar el backlog del producto y priorizar los elementos según el valor que ofrecen al cliente. Según lo dicho por DigitalOcean (2010), una gestión eficiente del backlog en Scrum incrementa la satisfacción del cliente y garantiza una mayor calidad del producto final.

1.2.2. Eventos en Scrum.

Sprint.

El Sprint, considerado el núcleo de Scrum, es un ciclo de trabajo con una duración fija de hasta cuatro semanas de crecimiento del producto. Según Schwaber y Sutherland (2020a), "el Sprint es un contenedor para todos los demás eventos de Scrum". Este ciclo permite que el equipo trabaje gradualmente, garantizando un progreso constante hacia el cumplimiento del objetivo del producto.

Daily Scrum.

Daily Scrum es una breve reunión diaria, donde un grupo de programadores coordina sus actividades y determina el plan de acción en las próximas 24 horas. Según un estudio de Cartaxo, Araújo, Barreto y Soares (2013), "el Daily Scrum mejora la comunicación, elimina otros encuentros, identifica obstáculos y promueve la toma rápida de decisiones". Esta práctica es esencial para mantener la transparencia y la colaboración dentro del equipo.

Sprint Review.

El Sprint Review se lleva a cabo al terminar cada Sprint para verificar y ajustar el Product Backlog según sea necesario. Según un artículo de Cartaxo, Araújo, Barreto y Soares (2013), "la Sprint Review proporciona una oportunidad para obtener feedback del cliente y otros stakeholders, lo cual es crucial para el éxito del proyecto". Este evento asegura que el producto evolucione de acuerdo con las necesidades del cliente.

Sprint Retrospective.

Sprint Retrospective es una sesión en la que el grupo analiza el Sprint realizado y diseña estrategias para implementar mejoras en el siguiente Sprint. Según un estudio de Lárusdóttir, Cajander y Simader (2014), "la Retrospective es fundamental para la mejora continua del equipo, permitiendo identificar y eliminar impedimentos". Este evento fomenta un ambiente de aprendizaje y adaptación constante.

Sprint Planning.

La Sprint Planning es la reunión que da inicio al Sprint. En esta etapa, el equipo establece qué elementos se pueden completar en el próximo Sprint y determina la estrategia para alcanzarlos. Según un artículo de Hema, Thota, Naresh Kumar, Padmaja, Rama Krishna y Mahender (2020), "una planificación efectiva del Sprint es crucial para alinear las expectativas del equipo y del Product Owner, asegurando que todos trabajen hacia un objetivo común". Esta reunión establece una base sólida para el trabajo del equipo durante el Sprint.

1.2.3. Beneficios de Scrum en el desarrollo de aplicativos móviles.

Beneficios en aplicativos móviles.

Scrum permite a los equipos de desarrollo de aplicaciones móviles enfocarse en la entrega de incrementos funcionales y de alta calidad en cada Sprint. Según Appinventiv (2024), "la metodología Scrum mejora la gestión de la calidad y asegura que cada incremento del producto cumpla con los estándares requeridos". Esto es crucial en el desarrollo de aplicaciones móviles, donde la calidad del producto final puede determinar su éxito en el mercado.

Además, la naturaleza iterativa de Scrum facilita que los programadores se ajusten rápidamente a los cambios en los requisitos del proyecto. Un artículo de Chandarana, Gada y Kukkadapu (2024) destaca que "el marco ágil de Scrum permite a los equipos de desarrollo de aplicaciones móviles adaptarse rápidamente a las nuevas tecnologías y cambios en las

demandas de los usuarios". Esta flexibilidad es esencial en un entorno tan dinámico como el desarrollo de aplicaciones móviles.

Es importante acotar que, Scrum promueve la transparencia y la colaboración entre los miembros del equipo, garantizando el éxito en todos los proyectos de desarrollo aplicaciones móviles. Según la página IdeaUsher (2024), "Scrum mejora la transparencia del equipo y facilita una mejor colaboración, lo que resulta en una mayor eficiencia y productividad". Esto garantiza que todos los integrantes del equipo estén sincronizados y enfocados en alcanzar un objetivo común.

Por otro lado, la metodología Scrum permite a los equipos de desarrollo de aplicaciones móviles entregar productos funcionales en ciclos cortos, lo que reduce significativamente el tiempo de comercialización. Un estudio de Bhimanapati, Jain, Aggarwal (2024) menciona que "Scrum acelera el proceso de desarrollo y permite a las empresas llevar sus aplicaciones móviles al mercado más rápidamente". Esto es especialmente importante en un mercado competitivo donde el tiempo es un factor crítico.

Finalmente, la Sprint Retrospective en Scrum permite a los equipos de desarrollo de aplicaciones móviles reflexionar sobre su desempeño y buscar continuamente formas de mejorar. Según JustCoded (2024), "Scrum promueve la mejora continua a través de la reflexión y el ajuste constante de procesos y prácticas". Esta mejora continua es necesario para fomentar la innovación en el desarrollo aplicaciones móviles.

1.3. Pruebas de estrés.

1.3.1. Definición y objetivos de las pruebas de estrés.

Definición.

Las pruebas de estrés en software son una técnica de evaluación que se utiliza para determinar la estabilidad y confiabilidad de un sistema bajo condiciones extremas. Estas pruebas consisten en exponer el software a cargas de trabajo que superen sus límites

operativos habituales, con el objetivo de identificar fallos y vulnerabilidades. Según el estudio de Goseva-Popstojanova y Trivedi (2001), "las pruebas de estrés son esenciales para garantizar que el software pueda manejar picos inesperados de carga sin fallar"

También se utilizan para optimizar el rendimiento de las aplicaciones móviles. Al identificar los límites de rendimiento y las áreas que necesitan mejoras, los desarrolladores pueden hacer ajustes y optimizaciones que aseguren un funcionamiento más eficiente y rápido de la aplicación bajo cargas pesadas (Xie et al., 2020).

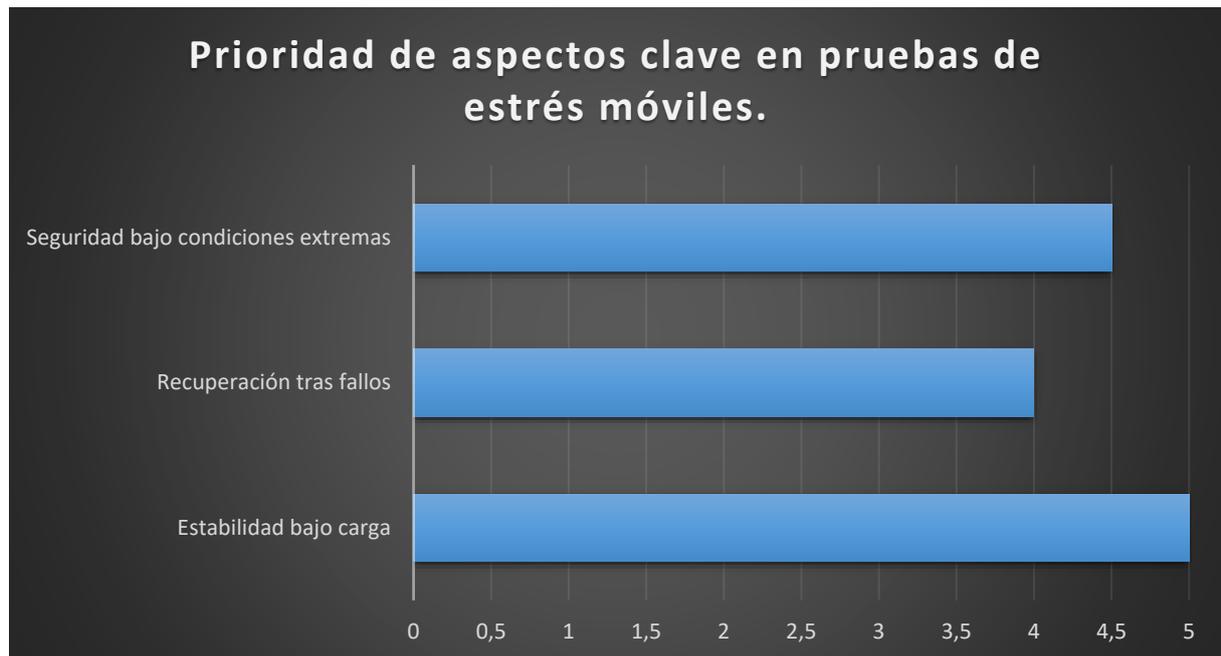
Objetivos.

Uno de los principales objetivos de las pruebas de estrés en aplicaciones móviles es evaluar la estabilidad del software cuando se enfrenta a condiciones extremas. Esto incluye situaciones como un alto volumen de usuarios simultáneos, uso intensivo de recursos y condiciones de red adversas. Estas pruebas permiten identificar y corregir posibles fallos antes de que afecten a los usuarios finales, asegurando así una experiencia de usuario óptima incluso en los peores escenarios (Iwama & Fukuda, 2019). En la **Figura 6** se muestra la prioridad de los aspectos clave en las pruebas de estrés de aplicaciones móviles, destacando la estabilidad bajo carga, la seguridad y la capacidad de recuperación como puntos principales.

De la misma forma, otro objetivo crucial es asegurar que la aplicación pueda recuperarse adecuadamente después de un fallo. Las pruebas de estrés permiten simular escenarios en los que el sistema falla y evaluar cómo se recupera. Esto es esencial para aplicaciones críticas donde la disponibilidad y la fiabilidad son primordiales, como en aplicaciones bancarias o de salud (Medeiros et al., 2020).

Figura 6

Prioridad de los aspectos clave en las pruebas de estrés.



Nota. Adaptado de (Berihun et al., 2023).

Finalmente, pueden ayudar a identificar vulnerabilidades de seguridad que podrían ser explotadas bajo condiciones de carga extrema. Al someter la aplicación a situaciones de estrés, es posible descubrir fallos de seguridad que no serían evidentes en condiciones normales, permitiendo a los desarrolladores corregir estos problemas antes de que puedan ser explotados por atacantes (Steiner, 2024).

1.3.2. Métodos para realizar pruebas de estrés.

Pruebas de Estrés en Aplicaciones Móviles

En aplicaciones móviles son esenciales para evaluar su rendimiento bajo condiciones extremas. Estas pruebas permiten identificar cuellos de botella y errores críticos antes de que los usuarios finales los experimenten. Un enfoque comúnmente utilizado es la simulación de múltiples usuarios simultáneos mediante herramientas como Apache JMeter. Esta herramienta genera carga sobre la aplicación, permitiendo medir su capacidad de respuesta y estabilidad ante una demanda inesperadamente alta (Autili et al., 2021). Este proceso es

fundamental para garantizar que las aplicaciones puedan manejar grandes volúmenes de tráfico sin colapsar.

De la misma forma, otro método crucial es la evaluación del rendimiento del servidor y las APIs, en estas pruebas se evalúa la capacidad del sistema para gestionar grandes volúmenes de datos y manejar múltiples solicitudes concurrentes. Estas pruebas no solo se centran en la aplicación, sino también en su infraestructura de soporte, identificando problemas de conexión y latencia bajo condiciones de estrés (Pecorelli et al., 2022). Investigaciones recientes han demostrado que el comportamiento del servidor bajo carga es un factor determinante para el éxito de las aplicaciones móviles en entornos reales (Sumner et al., 2023).

Cabe destacar que, se recomienda que las pruebas de estrés deben integrarse en el ciclo de desarrollo ágil. En lugar de realizarse solo antes de eventos específicos como lanzamientos, deben formar parte de los procesos continuos de integración y despliegue. Esto facilita la detección y corrección de problemas durante el desarrollo, evitando que se conviertan en fallos críticos en el entorno de producción. De esta manera, los desarrolladores pueden mejorar la calidad del producto de manera constante (Autili et al., 2021).

Consumo de Recursos del Dispositivo

Asimismo, es fundamental monitorear el consumo de recursos del dispositivo durante las pruebas de estrés. Estudios recientes han destacado que el uso excesivo de la batería o la memoria bajo condiciones de carga extrema puede impactar negativamente la experiencia del usuario. Optimizar el uso de recursos es crucial para asegurar que la aplicación funcione de manera eficiente, incluso en situaciones de alta demanda (Weichbroth, 2024).

Pruebas en Diversos Dispositivos y Plataformas.

Finalmente, las pruebas deben abarcar una variedad de dispositivos y plataformas, especialmente en un mercado tan fragmentado como el de los dispositivos móviles. Según

estudios, realizar pruebas en una amplia gama de dispositivos asegura que la aplicación funcione correctamente en diversas configuraciones de hardware y sistemas operativos, minimizando los problemas de compatibilidad (Autili et al., 2021).

1.3.3. Funcionalidades GPS.

Aplicación de pruebas de estrés en funcionalidades de GPS.

Las capacidades de GPS en aplicaciones móviles pueden verse afectadas cuando muchos usuarios intentan obtener su ubicación simultáneamente o cuando la señal GPS es débil. Las pruebas de estrés en este contexto buscan determinar si la aplicación puede manejar eficientemente la concurrencia de solicitudes y mantener un nivel aceptable de precisión. Estas pruebas son esenciales para aplicaciones que dependen del GPS, como las de navegación o seguridad, que necesitan asegurar una localización precisa en todo momento (Khandelwal, 2020).

Además, cuando la señal GPS es débil, la precisión de la localización puede verse afectada y generar retrasos en la respuesta de la aplicación. Las pruebas de estrés simulan estas condiciones para evaluar cómo se comporta la app en áreas con baja cobertura GPS o en entornos cerrados. El objetivo es determinar si la app sigue funcionando adecuadamente o si sufre caídas de rendimiento, permitiendo así a los desarrolladores identificar posibles mejoras en la tolerancia a fallos (Khandelwal, 2020).

En definitiva, las pruebas de estrés no solo son esenciales para detectar fallos bajo condiciones extremas, también juega un papel decisivo para asegurar la escalabilidad de la aplicación. Particularmente para las funcionalidades de GPS, estas pruebas pueden mostrar el comportamiento de la app en situaciones excepcionales, como eventos masivos o emergencias, permitiendo a los desarrolladores realizar ajustes necesarios para mejorar la resiliencia y estabilidad de la aplicación en escenarios críticos (Talreja, 2023).

1.3.4. Herramientas para pruebas de estrés en aplicaciones móviles.

Apache JMeter.

Apache JMeter, ampliamente usada para llevar a cabo pruebas de estrés. Su eficacia ha sido comprobada en la simulación de grandes volúmenes de tráfico y en la evaluación de la respuesta de las aplicaciones bajo condiciones de estrés (Matam & Jain, 2017). En particular, JMeter ha sido mejorado con plugins como DYNAMOJM, que ajusta dinámicamente la carga de trabajo durante las pruebas, permitiendo detectar errores de rendimiento de manera más eficiente que las técnicas tradicionales de prueba estática (Huerta et al., 2019). Estas características lo hacen ideal para aplicaciones móviles, donde los patrones de uso pueden variar significativamente

BlazeMeter.

BlazeMeter amplía las capacidades de JMeter, es otra herramienta de pruebas ampliamente utilizada. Facilita la ejecución de pruebas de estrés en la nube, simulando miles de usuarios simultáneos desde diversas ubicaciones geográficas, lo que resulta especialmente beneficioso para aplicaciones móviles con alcance global. Además, su integración con JMeter facilita la migración de scripts existentes (BlazeMeter, 2024).

Gatling.

Gatling es una herramienta potente para realizar simulaciones avanzadas de carga y estrés. Es particularmente útil para probar APIs REST en aplicaciones móviles, permitiendo evaluar el rendimiento durante picos de tráfico. Un análisis reciente resaltó la capacidad de Gatling para configurarse y manejar diversos escenarios y simulaciones personalizadas, lo que la convierte en una opción versátil y robusta para aplicaciones móviles (Khan et al., 2024).

Locust.

Locust es una herramienta de pruebas de carga y estrés escrita en Python, reconocida por su capacidad para simular millones de usuarios concurrentes. Se emplea principalmente

en aplicaciones móviles y web para replicar escenarios de uso real, esto simplifica la detección de cuellos de botella y problemas de rendimiento. Los desarrolladores pueden definir y personalizar escenarios de prueba. Además, permite monitorear en tiempo real, el tiempo de respuesta y tasa de errores. Su facilidad de uso y escalabilidad la convierten en una opción sólida para proyectos de gran envergadura (HyScaler, 2024).

Firestore Test Lab.

Firestore Test Lab es una plataforma de inspección en la nube para desarrolladores, permite realizar pruebas automatizadas con dispositivos físicos y virtuales. Diseñado específicamente para aplicaciones móviles, asegura que las aplicaciones funcionen correctamente en una amplia gama de dispositivos y configuraciones de red. Un estudio reciente destacó su utilidad para pruebas de localización y seguimiento en tiempo real, como en sistemas de transporte público, optimizando la precisión de los datos de ubicación en diversas condiciones (Skhosana & Ezugwu, 2021).

Postman.

Postman es una plataforma especializada para el desarrollo y prueba de APIs RESTful que tiene funcionalidades como, diseñar, documentar, testear y monitorear servicios web de manera eficiente. Además de sus funcionalidades para pruebas funcionales, cuenta con una herramienta llamada Runner, que permite realizar pruebas de carga y estrés en aplicaciones web y móviles. Estas a su vez son configurables mediante un número de solicitudes y tiempos de espera (Chaudhary, 2023).

1.4. Norma ISO/IEC 25023 disponibilidad.

1.4.1. Introducción a la norma ISO/IEC 25023.

La norma ISO/IEC 25023 es una parte fundamental de la familia de normas ISO/IEC 25000, conocida como SQuaRE (Software Quality Requirements and Evaluation). Esta norma ofrece un conjunto de datos para medición de manera cuantitativa la calidad de productos de

software y sistemas. Según la ISO (2016), ISO/IEC 25023:2016 establece índices de calidad para evaluar cuantitativamente la calidad de software y productos del sistema basados en ciertas características, establecidas en la ISO/IEC 25010.

Es importante enfatizar que una de las características más destacadas de ISO/IEC 25023 es su enfoque en la evaluación de la calidad tanto interna como externa del software. Esto abarca aspectos como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. En un estudio realizado por Maila, Intriago e Ibarra (2019), se utilizó la norma ISO/IEC 25023 para evaluar software de código abierto, destacando la importancia de estas métricas para obtener resultados precisos y objetivos.

Además, la adopción de ISO/IEC 25023 no solo beneficia a los desarrolladores de software, sino también a los usuarios finales. Al proporcionar productos de mayor calidad, se mejora la satisfacción del usuario y se reducen los costos asociados con fallos y mantenimientos. Un análisis de la norma ISO/IEC 25023 destaca que su implementación puede resultar compleja para personas sin experiencia en el área de calidad de software, debido a la amplia variedad de técnicas de medición que presenta (Calabrese & Muñoz, 2018).

Igualmente, se utiliza en conjunto con otras normas de la serie ISO/IEC 25000, como la 25010, que establece un modelo de calidad de software. La interacción entre estas normas facilita una evaluación más integral y precisa de la calidad del software. La norma ISO/IEC 25023:2016 contiene un conjunto básico de medidas de calidad para cada característica y subcaracterística, y una explicación de cómo aplicar estas medidas (Caiza Chacón, 2019).

Finalmente, la norma ISO/IEC 25023 juega un papel decisivo en la estandarización de la calidad del software a nivel mundial. Esto es especialmente relevante en un mundo cada vez más interconectado, donde los productos de software deben cumplir con estándares internacionales para ser competitivos (International Organization for Standardization, 2016).

1.4.2. Indicadores de calidad de disponibilidad.

La ISO/IEC 25023 determina un conjunto de indicadores para evaluar de manera cuantitativa, la calidad del software y productos del sistema basados en características especificadas en la norma ISO 25010. Entre estas medidas, los indicadores de disponibilidad son fundamentales para evaluar la capacidad de un sistema de estar operativo y accesible en el momento en que se requiere. Según ISO (2016), la disponibilidad se mide a través de la proporción de tiempo en que el sistema está operativo durante un período específico.

Importancia de la disponibilidad.

En los sistemas de software radica en su impacto directo en la satisfacción del usuario y en la eficiencia operativa. ISO (2016) destaca que la disponibilidad no solo depende de la infraestructura tecnológica, sino también de la gestión adecuada de los recursos y la planificación de mantenimiento. La Norma ISO/IEC 25023 proporciona directrices claras para medir estos aspectos y asegurar que los sistemas cumplan con los requisitos de disponibilidad.

Además, la relación entre disponibilidad y confiabilidad es particularmente relevante en el contexto de las aplicaciones móviles, donde los usuarios esperan un rendimiento constante y sin interrupciones. Un estudio de Herrera, Najjar, Palavecino y Goñi (2013) muestra que la confiabilidad de una aplicación móvil influye directamente en su disponibilidad, ya que una aplicación confiable tiene menos fallos y, por lo tanto, mayor tiempo de operación. La norma ofrece herramientas para evaluar y mejorar tanto la confiabilidad como la disponibilidad.

Métodos de medición.

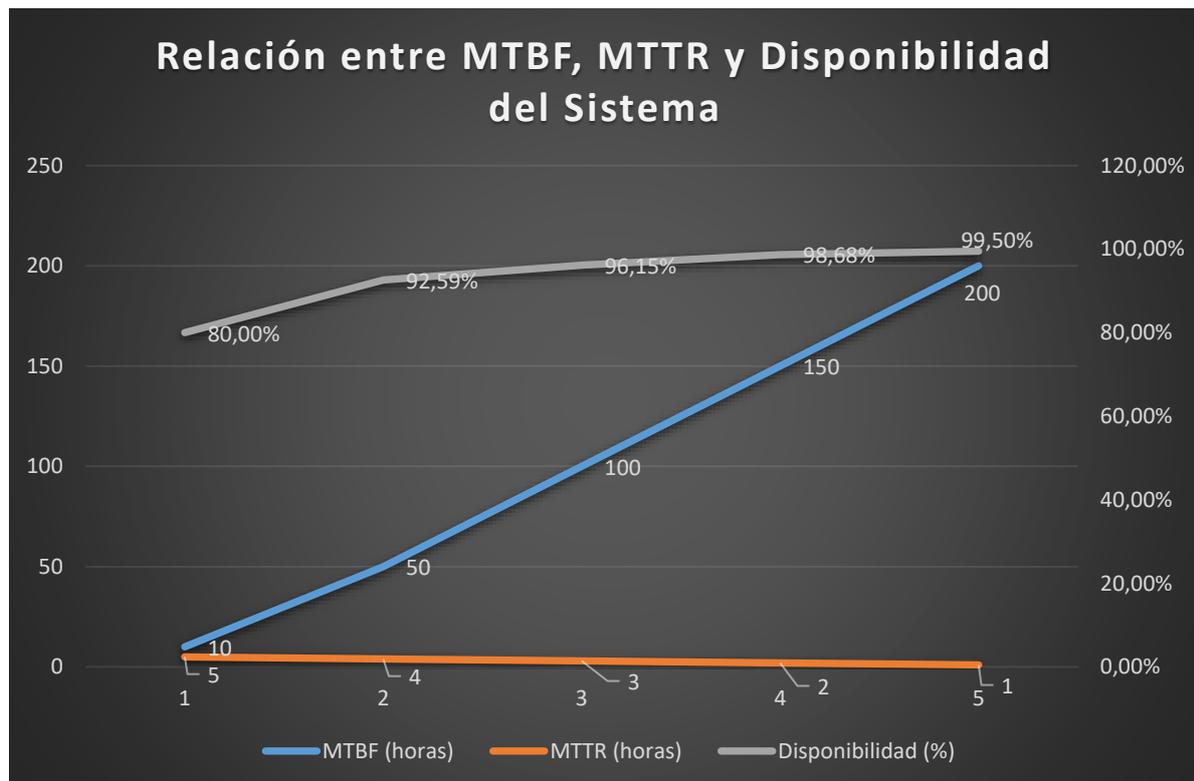
Los métodos para medir la disponibilidad incluyen métricas como el tiempo promedio entre la falla (MTBF) y el tiempo de reparación promedio (MTTR). En su investigación, Calabrese y Muñoz (2018) explican que estas métricas son esenciales para detectar áreas de mejora e implementar estrategias de mantenimiento preventivo. Como se muestra en la

Figura 7, la relación entre MTBF, MTTR y la disponibilidad del sistema refleja que un aumento en el MTBF, combinado con una disminución del MTTR, resulta en una mayor disponibilidad del sistema. La Norma ISO/IEC 25023 ofrece un marco estructurado para aplicar estas métricas de manera consistente y precisa.

La implementación de indicadores de disponibilidad según la Norma ISO/IEC 25023 en aplicaciones móviles puede ser desafiante debido a la diversidad de dispositivos y sistemas operativos. La ISO (2016) sugiere que la integración de herramientas de monitoreo avanzadas y la capacitación del personal es necesaria para superar desafíos de disponibilidad. La norma no solo define las métricas, sino que también ofrece orientación sobre cómo aplicarlas eficazmente en diferentes contextos.

Figura 7

Relación entre MTBF, MTTR y Disponibilidad del Sistema.



Nota. Adaptado de (Żyluk et al., 2023).

1.4.3. Métricas clave para evaluar la disponibilidad en sistemas.

Tiempo medio entre fallos (MTBF).

El MTBF mide el lapso promedio entre fallos de un software y es fundamental para evaluar la fiabilidad y la disponibilidad. Cuanto mayor sea el MTBF, más confiable será el sistema. Según el enfoque propuesto por Meeker, Escobar y Pascual (2022), el MTBF es una métrica central para medir la disponibilidad y se utiliza en conjunto con el tiempo de reparación para calcular el porcentaje de disponibilidad. El cálculo es importante para determinar la capacidad del sistema de mantener operaciones continuas y se calcula utilizando la fórmula $MTBF = T/N$ donde T representa el tiempo total de operación del sistema, y N es el número total de fallos ocurridos en ese período (Meeker et al., 2022).

Tiempo medio de reparación (MTTR).

El MTTR mide el tiempo promedio que tarda un sistema en recuperarse después de un fallo y se calcula mediante la fórmula $MTTR = T_R/N$ donde T_R es el tiempo total dedicado a las reparaciones y N es el número total de fallos ocurridos durante el período evaluado (Meeker et al., 2022). Esta métrica, junto con el MTBF, se utiliza para calcular la disponibilidad total del sistema. Según un estudio de Meeker, Escobar y Pascual (2022), el tiempo de restauración depende de la capacidad de soporte técnico y la infraestructura, siendo crucial para asegurar que los sistemas cumplan con los acuerdos de nivel de servicio (SLA).

Porcentaje de disponibilidad.

El porcentaje de disponibilidad, calculado mediante la fórmula $\% Disponibilidad = MTBF/(MTBF + MTTR)$, proporciona una visión cuantitativa del tiempo que un sistema permanece operativo. SEBoK (2024) destaca que este valor es clave para realizar una evaluación de la disponibilidad, ya que refleja tanto la frecuencia de fallos como el tiempo de recuperación, y es una métrica indispensable en sistemas críticos.

1.5. Desarrollo de Aplicaciones Móviles con GPS.

1.5.1. Definición y objetivos del uso de GPS en aplicaciones móviles.

Definición del GPS en aplicaciones móviles.

El Sistema de Posicionamiento Global (GPS) permite a las aplicaciones móviles determinar con precisión la ubicación de los usuarios mediante la triangulación de señales satelitales. Esta tecnología es crucial para servicios basados en la localización, como la navegación, el monitoreo y las aplicaciones de seguridad. El GPS ha revolucionado la interacción de los dispositivos móviles con su entorno, proporcionando datos de ubicación en tiempo real y facilitando la toma de decisiones informadas (Zangenehnejad & Gao, 2021). Según un estudio de Zangenehnejad y Gao (2021), los smartphones han mejorado significativamente en la recepción de señales del Sistema Global de Navegación por Satélite (GNSS) desde la implementación de mediciones GNSS crudas en dispositivos Android.

Objetivos de integrar la geolocalización en la gestión operativa

La incorporación del GPS en aplicaciones de gestión operativa busca optimizar la eficiencia, incrementar la seguridad y disminuir los costos requeridos. Por ejemplo, en el control de flotas, los administradores monitorean vehículos en tiempo real, mejorando la asignación de recursos y reduciendo los tiempos de inactividad. Esta integración permite una supervisión más precisa de las operaciones y aumenta la productividad, como lo menciona Schmidtke (2020) al analizar el impacto del GPS en aplicaciones móviles durante la pandemia de COVID-19.

En definitiva, el GPS permite a las empresas tomar decisiones informadas en tiempo real, basándose en la ubicación actualizada de sus activos o personal. Esta capacidad es crucial en sectores como la seguridad privada y el transporte, donde la optimización de rutas y la supervisión del personal pueden mejorar notablemente la eficiencia operativa. El uso de GPS en aplicaciones móviles ha demostrado cómo los datos en tiempo real permiten a los

usuarios ajustar sus rutas y evitar problemas de tráfico, optimizando así su tiempo de desplazamiento (Laor & Galily, 2022).

1.5.2. Componentes de Expo para la geolocalización

Descripción del módulo expo-location.

El módulo expo-location, ofrecido por Expo, permite a las aplicaciones móviles obtener datos de geolocalización en tiempo real. Este módulo utiliza los servicios de localización del dispositivo para determinar la ubicación exacta del usuario, ya sea a través de GPS, redes Wi-Fi o datos móviles. Las aplicaciones pueden solicitar la ubicación actual o suscribirse a actualizaciones periódicas de la ubicación. Según la documentación oficial de Expo, expo-location es compatible con iOS y Android, y proporciona acceso a información como latitud, longitud, altitud y velocidad del dispositivo (Expo Documentation, 2024).

Asimismo, para que una aplicación pueda utilizar el módulo expo-location, es necesario que los usuarios concedan permisos de localización como se ve en la **Figura 8**. Estos permisos pueden ser solicitados tanto para el uso en primer plano como en segundo plano, dependiendo de si la aplicación necesita rastrear la ubicación mientras está abierta o cuando se ejecuta en segundo plano. Además, se puede ajustar la precisión de la ubicación solicitada, ya que expo-location permite elegir entre alta precisión mediante GPS o una precisión menor utilizando redes móviles o Wi-Fi. Configurar adecuadamente los permisos es esencial para asegurar una experiencia de usuario fluida (Expo Documentation, 2024).

En conclusión, facilita la realización de operaciones de geocodificación, lo que implica convertir direcciones físicas en coordenadas de latitud y longitud. Esta capacidad es crucial para aplicaciones que necesitan determinar la ubicación exacta de un lugar a partir de una dirección proporcionada por el usuario (Expo Documentation, 2024). De acuerdo con Japara y Arifin (2023), la geocodificación es particularmente valiosa para verificar la posición del usuario en aplicaciones que requieren confirmación de ubicación.

Figura 8

Permisos de localización Android y IOS.



Nota. Adaptado de (Expo Documentation, 2024).

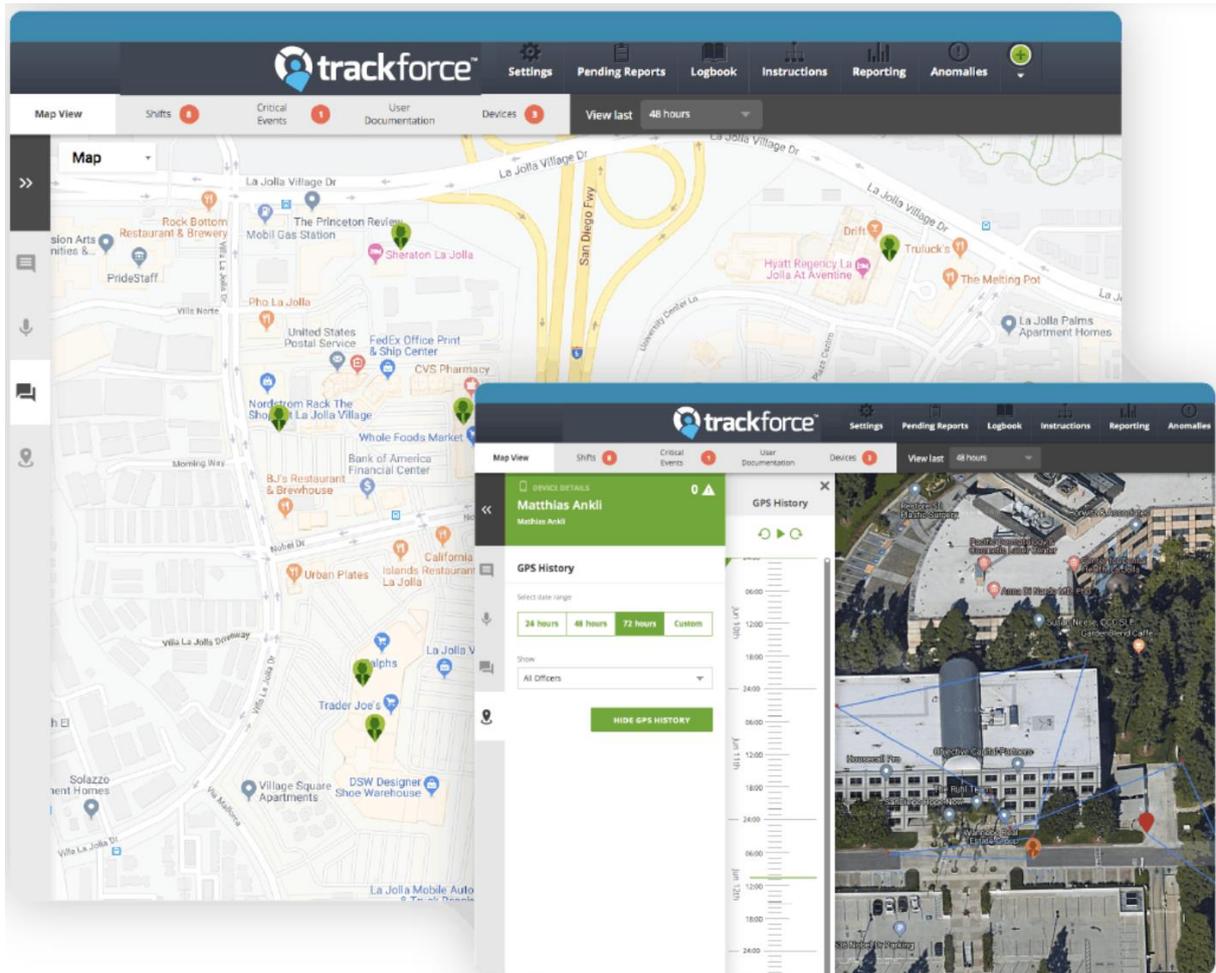
1.5.3. Usos del GPS en la gestión operativa.

Monitoreo en tiempo real de rutas de seguridad.

La integración del GPS con el módulo expo-location en aplicaciones móviles permite a las empresas de seguridad supervisar en tiempo real las rutas de patrullas y agentes. Esto no solo mejora la eficiencia operativa, también permite una respuesta rápida a incidentes. Por ejemplo, los supervisores pueden recibir notificaciones cuando los agentes se desvían de sus rutas establecidas, facilitando así una intervención inmediata. Esta tecnología también se ha utilizado en sistemas de monitoreo de soldados, mejorando la visibilidad de su ubicación y facilitando la coordinación en situaciones de emergencia (Sheeba et al., 2023). Un ejemplo de este tipo de uso es la app trackforce como se ejemplifica en la **Figura 9**.

Figura 9

Monitoreo de rutas en trackforce.



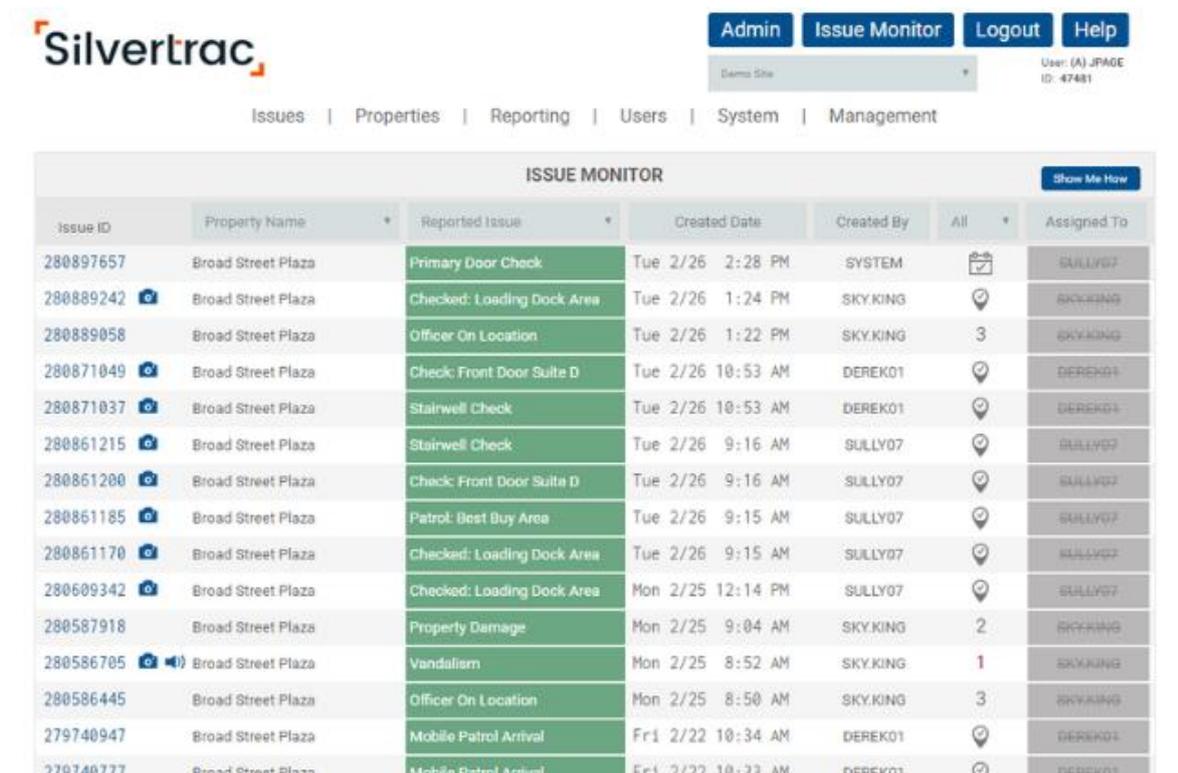
Nota. Adaptado de (Trackforce, 2020).

Reportes personalizados y automatizados.

Las aplicaciones de gestión de seguridad que incorporan GPS permiten crear informes detallados y personalizados basados en la ubicación y las actividades de los guardias. Estas herramientas son útiles para identificar patrones de comportamiento o áreas que necesitan mayor atención, mejorando así la toma de decisiones y asegurando el cumplimiento de las normativas de seguridad. Además, la automatización de los informes disminuye la carga administrativa y aumenta la eficiencia operativa (THERMS, 2023). Este es el caso de Silvertrac como se ve en la **Figura 10** que permite la administración de rutas mostrando un reporte de estas.

Figura 10

Reportes de rutas en Silvertrac.



The screenshot displays the Silvertrac Issue Monitor interface. At the top, there are navigation buttons for 'Admin', 'Issue Monitor', 'Logout', and 'Help'. Below these, the user is identified as 'User: (A) JPM0E ID: 47481'. A breadcrumb trail shows 'Issues | Properties | Reporting | Users | System | Management'. The main section is titled 'ISSUE MONITOR' and contains a table with the following columns: Issue ID, Property Name, Reported Issue, Created Date, Created By, All, and Assigned To. The table lists 18 issues, all for 'Broad Street Plaza', with various reported issues such as 'Primary Door Check', 'Checked: Loading Dock Area', 'Officer On Location', 'Check: Front Door Suite D', 'Stairwell Check', 'Patrol: Best Buy Area', 'Property Damage', and 'Vandalism'. Each row includes a date and time, the user who created the issue, and the number of assigned personnel.

Issue ID	Property Name	Reported Issue	Created Date	Created By	All	Assigned To
280897657	Broad Street Plaza	Primary Door Check	Tue 2/26 2:28 PM	SYSTEM	1	SULLY07
280889242	Broad Street Plaza	Checked: Loading Dock Area	Tue 2/26 1:24 PM	SKY.KING	1	SKY.KING
280889058	Broad Street Plaza	Officer On Location	Tue 2/26 1:22 PM	SKY.KING	3	SKY.KING
280871049	Broad Street Plaza	Check: Front Door Suite D	Tue 2/26 10:53 AM	DEREK01	1	DEREK01
280871037	Broad Street Plaza	Stairwell Check	Tue 2/26 10:53 AM	DEREK01	1	DEREK01
280861215	Broad Street Plaza	Stairwell Check	Tue 2/26 9:16 AM	SULLY07	1	SULLY07
280861200	Broad Street Plaza	Check: Front Door Suite D	Tue 2/26 9:16 AM	SULLY07	1	SULLY07
280861185	Broad Street Plaza	Patrol: Best Buy Area	Tue 2/26 9:15 AM	SULLY07	1	SULLY07
280861170	Broad Street Plaza	Checked: Loading Dock Area	Tue 2/26 9:15 AM	SULLY07	1	SULLY07
280609342	Broad Street Plaza	Checked: Loading Dock Area	Mon 2/25 12:14 PM	SULLY07	1	SULLY07
280587918	Broad Street Plaza	Property Damage	Mon 2/25 9:04 AM	SKY.KING	2	SKY.KING
280586705	Broad Street Plaza	Vandalism	Mon 2/25 8:52 AM	SKY.KING	1	SKY.KING
280586445	Broad Street Plaza	Officer On Location	Mon 2/25 8:50 AM	SKY.KING	3	SKY.KING
279740947	Broad Street Plaza	Mobile Patrol Arrival	Fr 1 2/22 10:34 AM	DEREK01	1	DEREK01
279740777	Broad Street Plaza	Mobile Patrol Arrival	Fr 1 2/22 10:33 AM	DEREK01	1	DEREK01

Nota. Adaptado de (Silvertrac Software, 2024).

En resumen, el uso del GPS en la gestión operativa, especialmente con herramientas como expo-location, ha optimizado el monitoreo en tiempo real, la supervisión remota y la generación de reportes automatizados. Estas tecnologías aumentan la eficiencia, mejoran la seguridad y permiten tomar decisiones más rápidas y precisas, siendo esenciales en sectores como la seguridad privada y la logística (THERMS, 2023).

CAPÍTULO 2

Desarrollo del proyecto

2.1 Planificación del proyecto.

2.1.1 Historias de usuario.

En metodologías ágiles como scrum, las historias de usuario son descripciones concisas de la funcionalidad desde la perspectiva del usuario, lo cual es un elemento crucial. Al convertir los requisitos del cliente en tareas manejables, facilitan la planificación de sprints y la estimación del esfuerzo a través de puntos de la historia, lo que permite asignar tareas según su valor (Amin Kuhail & Lauesen, 2022).

Para el proyecto pertinente se presentan las siguientes historias de usuario, la **Tabla 2** expone los detalles de la historia de usuario uno, considerando los campos: ID, nombre, descripción y verificación.

Tabla 2

Historia de usuario 1 – Control de turnos de guardias de seguridad.

Historia de Usuario

ID: HU-1

Usuario: Jefe operativo.

Nombre de historia: Control de turnos de guardias de seguridad.

Descripción:

Como jefe operativo, requiero controlar el ingreso del personal al puesto de trabajo mediante evidencias (fotográficas, ubicación, fecha y hora), visualizando dicha información para cada guardia y supervisor en cada puesto de trabajo.

Validación:

- Puedo ingresar al aplicativo web con mi usuario y contraseña.
-

- Puedo visualizar las evidencias (fotográficas, ubicación, fecha y hora) de cada puesto.
- Puedo visualizar las evidencias puntuales de cada guardia y supervisor en su respectivo puesto.

Nota. Autoría propia.

Seguidamente, la **Tabla 3** expone los detalles de la historia de usuario dos, considerando los mismos campos.

Tabla 3

Historia de usuario 2 – Gestión del registro de personal activo.

Historia de Usuario

ID: HU-2

Usuario: Jefe operativo.

Nombre de historia: Gestión del registro de personal activo.

Descripción:

Como jefe operativo, quiero registrar al personal con su nombre, apellido, rol, correo electrónico y contraseña, asignarles un puesto ya creado, gestionar su acceso, incluyendo dar de baja a los usuarios cuando sea necesario, y visualizar informes de historial de asignación de puestos para guardias y supervisores.

Validación:

- Puedo registrar nuevos usuarios para que accedan al aplicativo móvil.
 - Puedo asignar puestos previamente creados a los usuarios registrados.
 - Puedo revocar el acceso a los usuarios existentes.
-

- Puedo visualizar informes de historial de asignación de puestos de guardias y supervisores.

Nota. Autoría propia.

A continuación, la **Tabla 4** expone los detalles de la historia de usuario tres, usando los campos requeridos.

Tabla 4

Historia de usuario 3 – Gestión del puesto de guardia.

Historia de Usuario	
ID: HU-3	Usuario: Guardia.
Nombre de historia: Gestión del puesto de guardia.	
Descripción: Como guardia, quiero registrar mi turno al ingresar al aplicativo tomando la fotografía solicitada. Además, podré documentar cualquier incidente mediante una fotografía (opcional), descripción, fecha, hora y ubicación del incidente.	
Validación: <ul style="list-style-type: none">• Puedo ingresar al aplicativo móvil con mi usuario y contraseña• Puedo registrar mi ingreso al puesto mediante una fotografía.• Puedo documentar incidentes sucedidos en el puesto mediante una fotografía (opcional), descripción, fecha, hora y ubicación del incidente.	

Nota. Autoría propia.

Finalmente, la **Tabla 5**, expone los detalles de la historia de usuario cuatro, con los valores requeridos.

Tabla 5

Historia de usuario 4 – Gestión de los puestos a supervisar.

Historia de Usuario	
ID: HU-4	Usuario: Supervisor.
Nombre de historia: Gestión de los puestos a supervisar.	
Descripción:	
<p>Como supervisor, quiero registrar mi turno al ingresar al aplicativo tomando la fotografía solicitada. Además, quiero documentar cualquier incidente mediante una fotografía (opcional), junto con la descripción, fecha, hora y ubicación del incidente. También quiero ver la información de ingreso de los guardias (fotografía, ubicación, fecha y hora) de cada puesto, además de poder agregar avisos de multas a dichos guardias en caso de anomalías. Finalmente, quiero generar reportes con toda la información obtenida durante el día, los cuales se enviarán automáticamente al correo de la empresa tras una confirmación.</p>	
Validación:	
<ul style="list-style-type: none">• Puedo ingresar al aplicativo móvil con mi usuario y contraseña• Puedo registrar mi ingreso al puesto mediante una fotografía.• Puedo documentar incidentes sucedidos en el puesto mediante una fotografía (opcional), descripción, fecha, hora y ubicación del incidente.• Puedo visualizar las evidencias (fotográficas, ubicación, fecha y hora) de cada puesto.• Puedo agregar avisos de multas a guardias como un incidente en el puesto.• Puedo generar reportes con la información de los puestos obtenida durante el día y que se envíe al correo de la empresa automáticamente.	

Nota. Autoría propia.

2.1.2 Product Backlog.

Un Product Backlog es un registro detallado de todos los requisitos y tareas necesarias para asegurar el desarrollo o la mejora de un producto. El Product Owner es la principal persona responsable de gestionarlo. El actor tiene la tarea de identificar los elementos en función de su valor para el negocio y las necesidades del cliente. Según Schwaber y Sutherland (2020b), “el Product Owner es el único responsable de la administración del Product Backlog, asegurando que esté claro, transparente y correctamente priorizado para maximizar el valor del trabajo del equipo Scrum”.

Hay dos apartados importantes para el desarrollo del Product Backlog, la priorización de tareas y evaluar los esfuerzos de cada HU en el proyecto.

Priorización de tareas.

El método de MoSCoW es una herramienta útil para gestionar los requisitos en proyectos ágiles priorizando tareas. La **Figura 11** muestra las cuatro categorías de tareas que se pueden organizar utilizando este método. El uso de este enfoque permite a los equipos priorizar problemas críticos y ajustar sus entregas en función del tiempo y los recursos disponibles. Además que es flexible y puede adaptarse para maximizar el valor entregado al cliente en un entorno de desarrollo ágil (Kravchenko et al., 2022).

Figura 11

Priorización de MoSCoW.

Priorización de MoSCoW.



Nota. Adaptado de (Libre Scrum, 2020).

Estimación de Esfuerzo.

Las metodologías ágiles a menudo incorporan Story Points para medir el esfuerzo relativo de las tareas o historias de usuarios en el desarrollo de software, como se muestra en **Tabla 6**. Al centrarse en la complejidad, la incertidumbre y el esfuerzo necesario, facilitan que los equipos tengan una mentalidad más flexible y adaptable. Además, investigaciones recientes han revelado que su implementación reduce la carga de los equipos de desarrollo, permitiéndoles concentrarse en crear valor en lugar de entregar tiempo a los clientes (Mallidi & Sharma, 2021).

Tabla 6*Estimación de esfuerzo con Story Points*

Story Points	Estimación
1	1 - 2 horas
2	2 - 4 horas
3	4 - 8 horas
5	8 - 16 horas
8	16 - 24 horas
13	24 - 40 horas
21	40+ horas

Nota. Adaptado de (Mallidi & Sharma, 2021).

Creación del Product Backlog.

Con todos los aspectos ya mencionados se puede crear el Product Backlog que se exhibe a continuación en la **Tabla 7**.

Tabla 7*Product Backlog.*

ID	Alias	Prioridad	Esfuerzo
HU-01	Control de turnos de guardias de seguridad.	Must have	13
HU-02	Gestión del registro de personal activo.	Must have	13
HU-03	Gestión del puesto de guardia.	Must have	13
HU-04	Gestión de los puestos a supervisar.	Must have	21

Nota. Autoría propia.

2.1.3 Creación de Sprints.

Mediante las historias de usuario y la creación del Product Backlog se procede a crear los Sprints que ayudaran a garantizar la resolución del proyecto mediante entregas de valor interactivas. Como se puede ver en la tabla cada Sprint consta de la historia de usuario con un lapso para cumplirla.

Tabla 8

Proyección de los Sprints.

ID	Seudónimo	Sprint	Fecha
HU-1	Control de turnos de guardias de seguridad.	0	23/09/2024 -
HU-2	Gestión del registro de personal activo.		02/12/2024
HU-3	Gestión del puesto de guardia.	1	03/12/2024 –
HU-4	Gestión de los puestos a supervisar.		27/04/2025

Nota. Autoría propia.

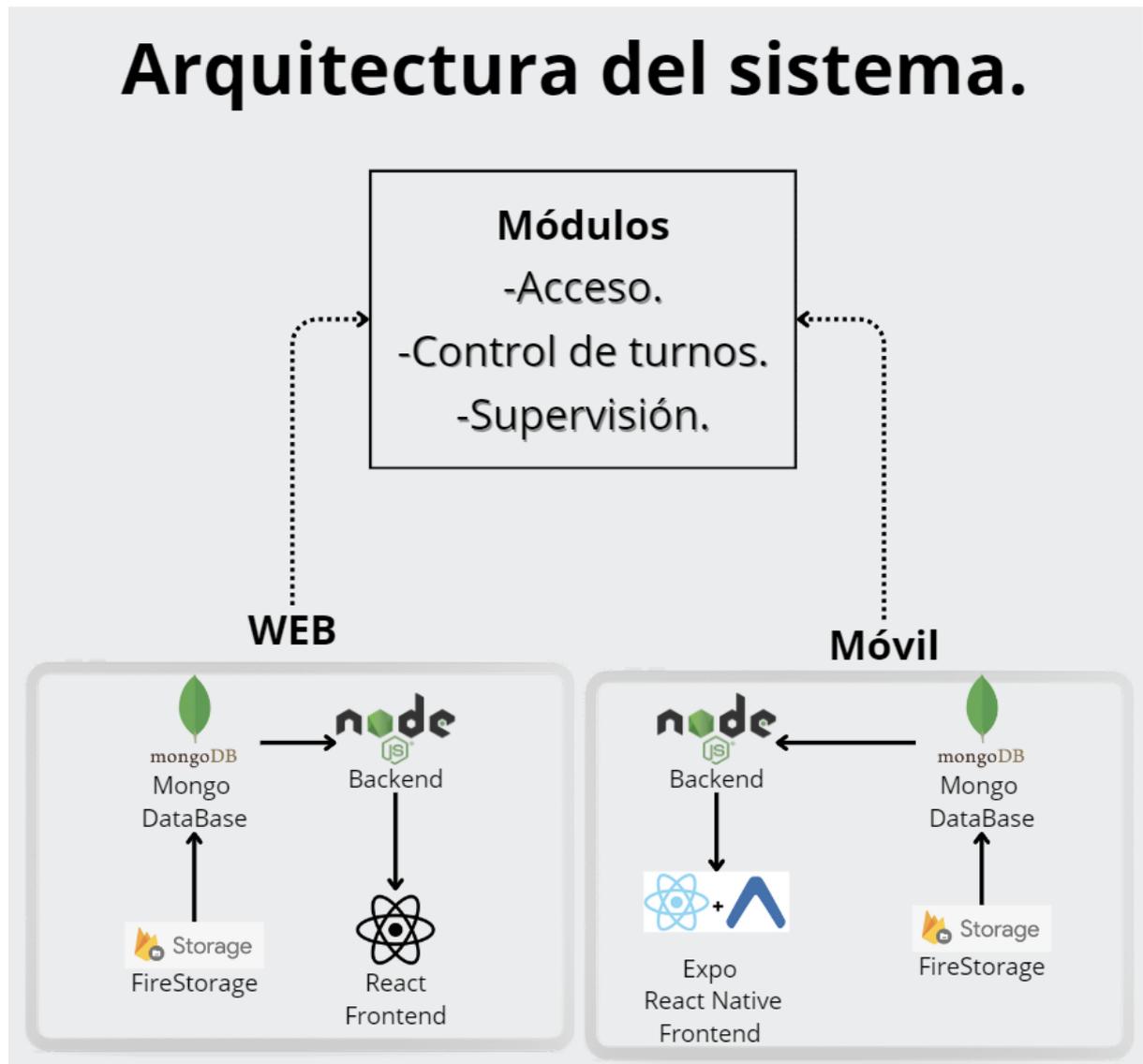
2.2. Diseño.

2.2.1. Arquitectura del sistema.

Se divide en dos partes principales: la versión web y la parte móvil, conectadas a módulos clave del sistema, que incluyen: "Acceso", "Control de turnos" y "Supervisión" como se presenta en la **Figura 12**.

Figura 12

Arquitectura general del sistema.



Nota. Autoría propia.

En la parte de Web, el sistema utiliza la base de datos MongoDB para almacenar datos y los archivos o documentos multimedia se administran a través de FireStorage. El backend se desarrolla utilizando Node.js, interactuando con el frontend integrado en React.

Por otro lado, la parte móvil tiene una estructura similar, donde MongoDB también sirve como base de datos y FireStorage como almacenamiento. El backend se comparte también usando Node.js, pero el frontend está diseñado usando React Native y Expo, lo que permite la creación flexible de aplicaciones móviles.

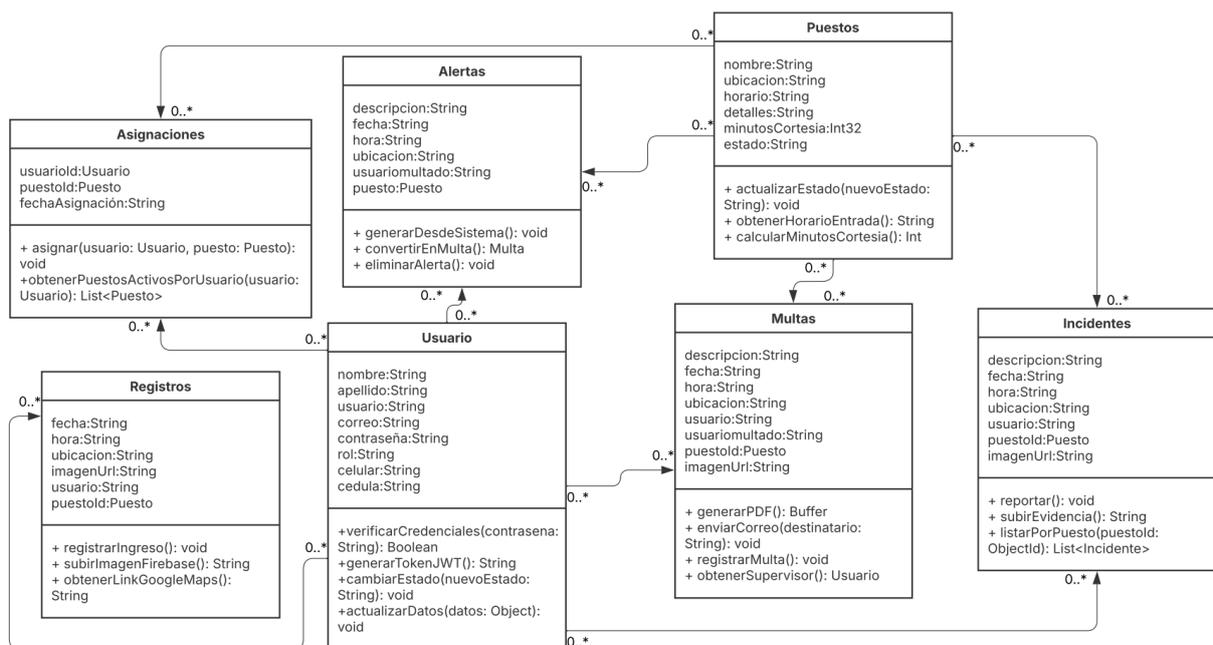
Esta arquitectura permite mejor mantenimiento y permite que se desarrollen las verificaciones de una manera más sencilla al tener herramientas similares.

2.2.2. Diagramas de clases.

Es una herramienta de visualización utilizada en el modelado de sistemas para representar la estructura estática de una aplicación. Estos diagramas detallan clases, propiedades, métodos y relaciones entre ellos, como herencia, asociación y dependencia. Se podría decir que muestran la estructura de cada clase como parte del proyecto, como se muestra en la **Figura 13**.

Figura 13

Diagrama de clases.



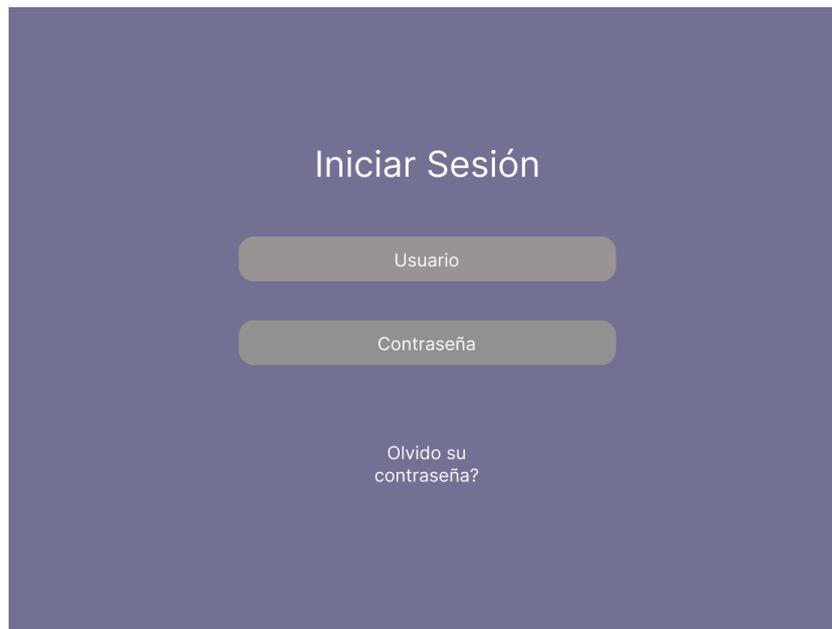
Nota. Autoría propia.

2.2.3. Prototipo del Administrador web

La **Figura 14**, muestra el prototipo del administrador web, la página de inicio de sesión, el cual muestra cómo se plantea realizar el ingreso al administrador web mediante usuario y contraseña, adicionalmente permitirá cambiar la contraseña si el usuario la olvido.

Figura 14

Prototipo "Pantalla de Login".



Nota. Autoría propia.

La **Figura 15**, muestra lo referente a la Pantalla Principal, el cual ilustra la pantalla que podrá permitir navegar al usuario por los diferentes apartados presentes en el administrador web y cerrar la sesión actual.

Figura 15

Prototipo "Pantalla principal".



Nota. Autoría propia.

La **Figura 16**, presenta “Registrar Usuarios”, el cual ilustra la página encargada de permitir al administrador añadir nuevos usuarios los cuales posteriormente podrán ingresar a la aplicación móvil.

Figura 16
Prototipo “Vista de registrar usuarios”.

Registrar Usuario

Nombre.

Apellido.

Correo Electronico.

Rol(Guardia,Supervisor,Administrador).

Celular.

Cedula.

Contraseña.

Confirmar Contraseña.

Crear

Nota. Autoría propia.

La **Figura 17** , presenta la pantalla “Lista de Usuarios”, encargada de mostrar a todos los usuarios y permite Revocar o cambiar de estado al usuario y Editar dicho usuario.

Figura 17
Prototipo “Pantalla de Lista de Usuarios”.

Lista de Usuarios

Acciones. Nombre. Apellido. Rol. Celular. Cedula. Usuario. Correo Electronico. Estado.

Revocar. Editar.	Ulises	Valencia.	Guardia.	098460...	100511...	S100511...	ulisesu@gmail...	Activo.
Revocar. Editar.
Revocar. Editar.
Revocar. Editar.

Ant. 1 Sig.

Nota. Autoría propia.

La **Figura 18**, presenta la pantalla “Registrar Puesto”, permite agregar nuevos puestos para su posterior uso utilizando la ubicación seleccionada en un mapa.

Figura 18
Prototipo “Pantalla de Registrar Puesto”.



Nota. Autoría propia.

La **Figura 19**, presenta la pantalla “Lista de Puestos”, encargada de mostrar todos los puestos creados, además, de Revocar o cambiar de estado y Editar un puesto creado.

Figura 19
Prototipo “Pantalla de Lista de Puestos”.



Nota. Autoría propia.

La **Figura 20**, presenta la pantalla “Asignar Puestos”, que permite al seleccionar un puesto y un usuario, crear una asignación a un puesto.

Figura 20
Prototipo “Pantalla de Asignar Puestos”.



Nota. Autoría propia.

La **Figura 21**, presenta la pantalla “Historial de asignación”, encargada de mostrar el historial de asignaciones a puestos con la fecha en que sucedió la misma.

Figura 21
Prototipo “Pantalla de Historial de asignación”.



Nota. Autoría propia.

La **Figura 22**, presenta la pantalla “Control de turnos”, que permite mostrar toda la evidencia obtenida de un puesto en el aplicativo móvil enfocado en el registro de turno.

Figura 22
Prototipo “Pantalla de Control de turnos”.

Usuario.	Fecha.	Hora	Puesto	Ubicación	Fotografía
Ulises Valencia.	11-11-2024	07:05am	RIOLAB	verUbicacion	verImagen
.....
.....
.....

Nota. Autoría propia.

La **Figura 23**, presenta la pantalla “Multas”, similar a la anterior permite mostrar toda la evidencia obtenida de un puesto en el aplicativo móvil enfocado en el registro de multas.

Figura 23
Prototipo “Pantalla de Multas”.

Supervisor	Multado.	Descripción	Fecha	Hora	Puesto	Ubicación	Evidencia
Ulises Valencia.	Ivan Cevallos	Llego tarde.	11-11-2024	07:20am	RIOLAB	VerUbicación	VerImagen
.....
.....
.....

Nota. Autoría propia.

La **Figura 24**, presenta la pantalla “Incidentes”, similar a las anteriores permite mostrar toda la evidencia obtenida de un puesto en el aplicativo móvil enfocado en el registro de incidentes.

Figura 24
Prototipo “Pantalla de Incidentes”.

Usuario	Descripción	Fecha	Hora	Puesto	Ubicación	Evidencia
Ulises Valencia.	Llego tarde.	11-11-2024	07:20am	RIOLAB	VerUbicación	VerImagen
.....
.....
.....

Nota. Autoría propia.

2.2.4. Prototipo de la aplicación móvil

A continuación, la **Figura 25**, muestra la primera pantalla, en esta se realizara el respectivo Login, que permitirá al usuario ya sea guardia o supervisor, ingresar a la aplicación con su usuario y contraseña.

Figura 25
Prototipo móvil - Pantalla de Login

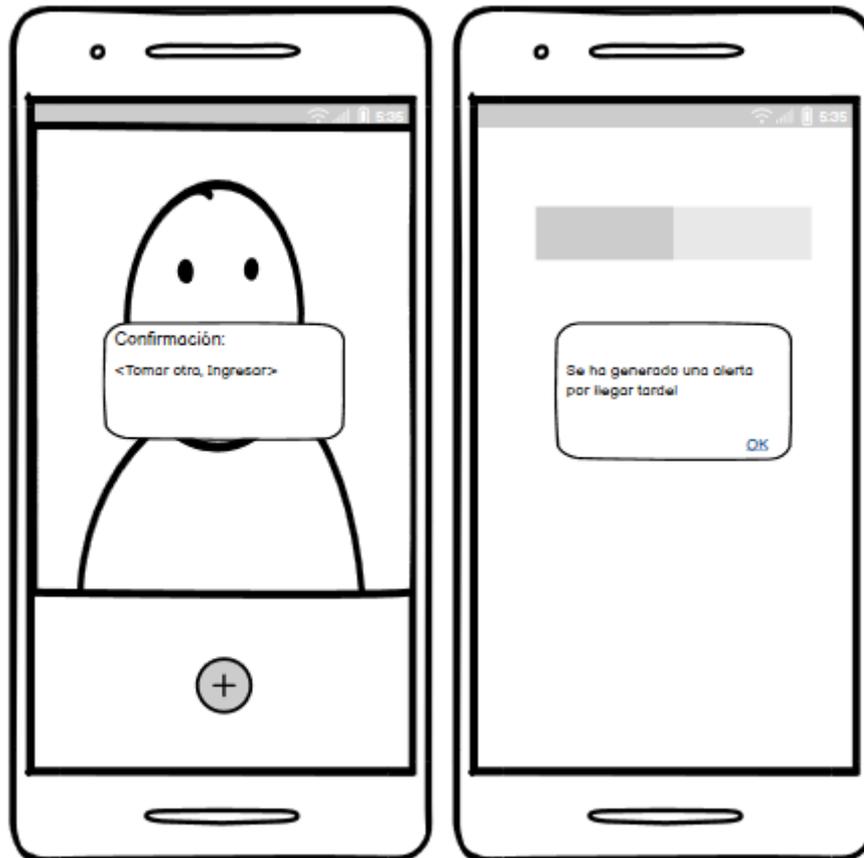


Nota. Autoría propia.

La **Figura 26**, muestra el ingreso del personal mediante una fotografía como evidencia, además de mostrar una alerta si es necesario por llegar tarde.

Figura 26

Prototipo móvil - Pantalla de Ingreso del personal.

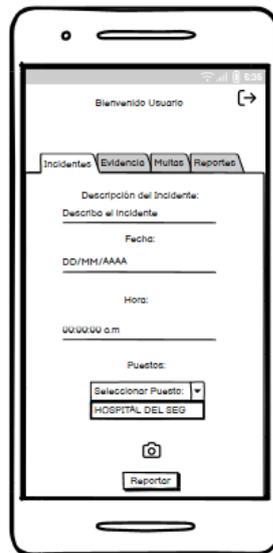


Nota. Autoría propia.

La **Figura 27**, presenta el apartado inicial con todas las sub-pantallas, que, a su vez, permiten la administración de estas, en este caso la pantalla de incidentes permite el ingreso de un nuevo incidente dando una corta descripción, la fecha, la hora, el puesto donde ocurrió y finalmente la evidencia de esta con una fotografía.

Figura 27

Prototipo móvil - Pantalla principal en Administración de Incidentes.

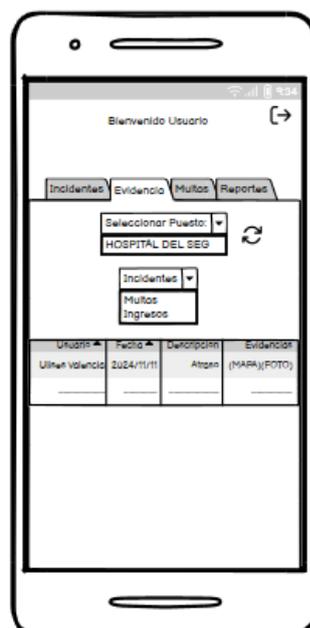


Nota. Autoría propia.

A continuación, se presenta como se mostrará la evidencia obtenida, que serán las fotografías y el mapa de la ubicación, de los incidentes, multas e ingresos, todo esto en la **Figura 28**.

Figura 28

Prototipo móvil - Pantalla principal en Evidencia del Puesto.

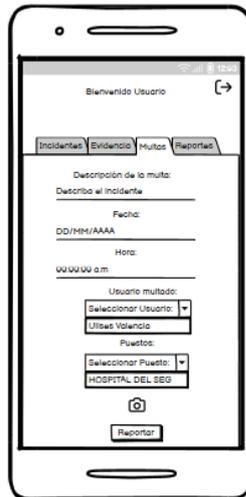


Nota. Autoría propia.

En la **Figura 29**, se muestra algo muy parecido al apartado de incidentes, pero en este caso para multas las cuales se harán de la misma manera, pero agregando la selección del usuario multado, el cual será el afectado por la multa creada.

Figura 29

Prototipo móvil - Pantalla principal en Administración de multas.

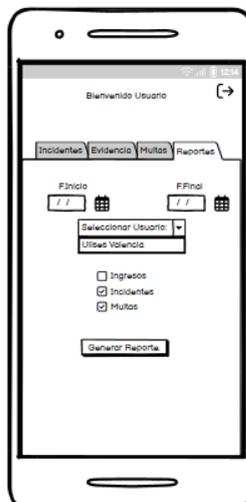


Nota. Autoría propia.

La **Figura 30**, muestra el proceso de creación de reportes, se pedirá algunos datos para la filtración y posterior generación del reporte solicitado, datos como la fecha relativa, el usuario requerido y las tablas que se quieren generar.

Figura 30

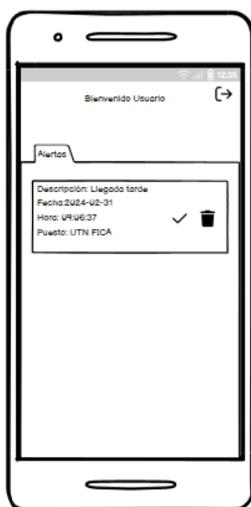
Prototipo móvil - Pantalla principal en Generación de reportes.



Nota. Autoría propia.

Finalmente, la **Figura 31**, presenta el control de alertas, que muestra las alertas generadas cuando un usuario llega tarde, las cuales el Supervisor, decidirá en base a las circunstancias se creará una multa o se descartará la alerta.

Figura 31
Prototipo móvil - Pantalla principal en Control de Alertas.



Nota. Autoría propia.

2.3. Desarrollo del sistema

2.3.1. Módulos del proyecto

Como cualquier proyecto de software, este se compone de varios módulos, los cuales mediante su interconexión hacen que el proyecto funcione de manera correcta. A continuación **Tabla 9**, en la se describen los módulos que constaran en el proyecto.

Tabla 9
Módulos del proyecto

Módulo	Descripción	Funcionalidades
Administrador Web	Interfaz para encargarse de usuarios y datos del proyecto.	<ul style="list-style-type: none"> • Crear Usuarios • Mostrar Usuarios • Modificar Usuarios • Crear Puestos • Mostrar Puestos • Asignar Puestos a Usuarios • Mostrar historial de asignación • Mostrar evidencias

Web services (API)	Interfaz de programación que permite conectarse entre frontend y backend.	<ul style="list-style-type: none"> • Administración de Usuarios, Registros, Puestos, Incidentes, Multas, Asignaciones y Alertas • Establecer la lógica del administrador web y aplicación móvil
Aplicación Móvil	Plataforma móvil que utiliza Expo- React native para la gestión de actividades operativas.	<ul style="list-style-type: none"> • Generar evidencia (fotografías, coordenadas de geolocalización) del puesto • Generación de ingresos, incidentes, multas, alertas y reportes • Mostrar evidencias obtenidas

Nota. Autoría propia.

2.3.2. Sprint 0

En la **Tabla 10**, se presenta lo referente al Sprint 0, describiendo las historias y tareas del usuario asignado a este sprint en detalle, así como la estimación de tiempo adecuada. Este es necesario para administrar y monitorear el proceso del proyecto.

Tabla 10

Sprint 0

SPRINT BACKLOG				
Sprint: 0				
Total horas: 138				
Fecha Inicio SP0: 23/09/2024				
Fecha Final SP0: 02/12/2024				
Actividad / Historia de usuario	Fase Desarrollo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO REAL (Horas)
Inicialización del proyecto.	Desarrollo	Creación y configuración del proyecto.	5	6
HU-01 Control de turnos de guardias de seguridad.	Análisis	Estudio de MongoDB y Cloud Storage for Firebase.	6	8

	Análisis	Revisar la implementación de la API de Google Maps.	5	7
	Análisis	Revisión de requerimientos de control de turnos.	6	6
	Desarrollo	Creación de base de datos MongoDB y colecciones iniciales.	8	9
	Desarrollo	Implementación de almacenamiento de evidencias en Firestore.	10	11
	Desarrollo	Implementación del API de Google Maps.	8	9
	Desarrollo	Desarrollo del módulo de Administración-Puestos.	12	14
	Desarrollo	Desarrollo del módulo de Administración-Evidencias.	10	11
	Desarrollo	Pruebas y Ajustes.	8	9
HU-02 Gestión del registro de personal activo.	Análisis	Revisión de requisitos de usuarios y roles.	5	6
	Desarrollo	Desarrollo del módulo de Administración-Usuarios.	10	12
	Desarrollo	Desarrollo del módulo de Administración-Asignación-Registro.	8	9
	Desarrollo	Desarrollo del módulo de Administración-Asignación-Historial.	8	9
	Desarrollo	Pruebas y ajustes	6	7
	Reuniones	Gestión	Planificación	1
Gestión		Diarias	2	2
Gestión		Revisión	1	1
Gestión		Retrospectiva	1	1
TOTAL			138	149

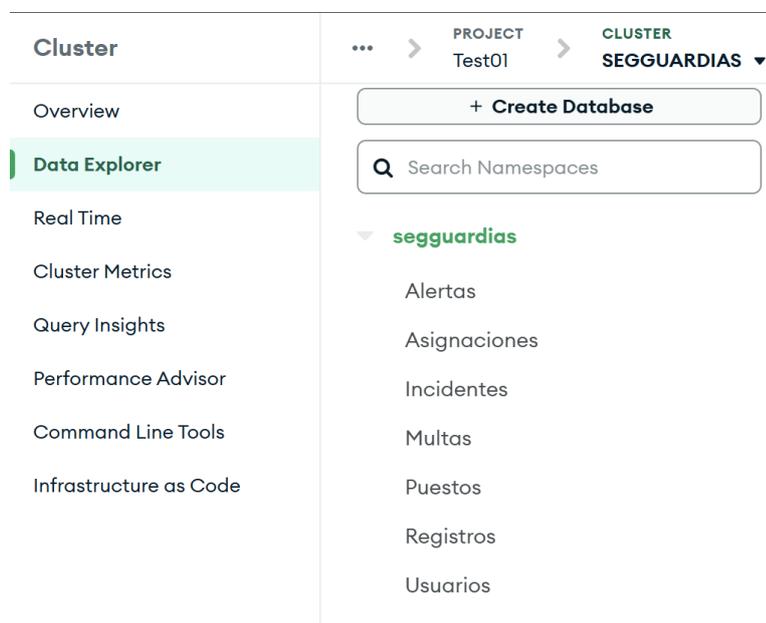
Nota. Autoría propia.

Como etapa inicial, se procedió a la creación del proyecto base utilizando el entorno de desarrollo Visual Studio Code. En este espacio se instalaron todas las

herramientas definidas previamente en la arquitectura, distribuidas en tres proyectos independientes.

El primero corresponde al desarrollo del API REST, el cual tiene como función principal gestionar las operaciones del sistema y facilitar la comunicación entre los módulos, incluyendo la integración con FirebaseStorage como servidor de archivos y MongoDB como base de datos.

Figura 32
Base de datos con sus colecciones.



Nota. Autoría propia.

La **Figura 32** presenta la base de datos denominada “seguardias”, implementada en MongoDB. Esta base de datos contiene un total de siete colecciones, cuyas estructuras se detallan en la **Tabla 11**. Los atributos definidos en dichas colecciones permiten almacenar la información requerida para el funcionamiento de cada uno de los módulos desarrollados.

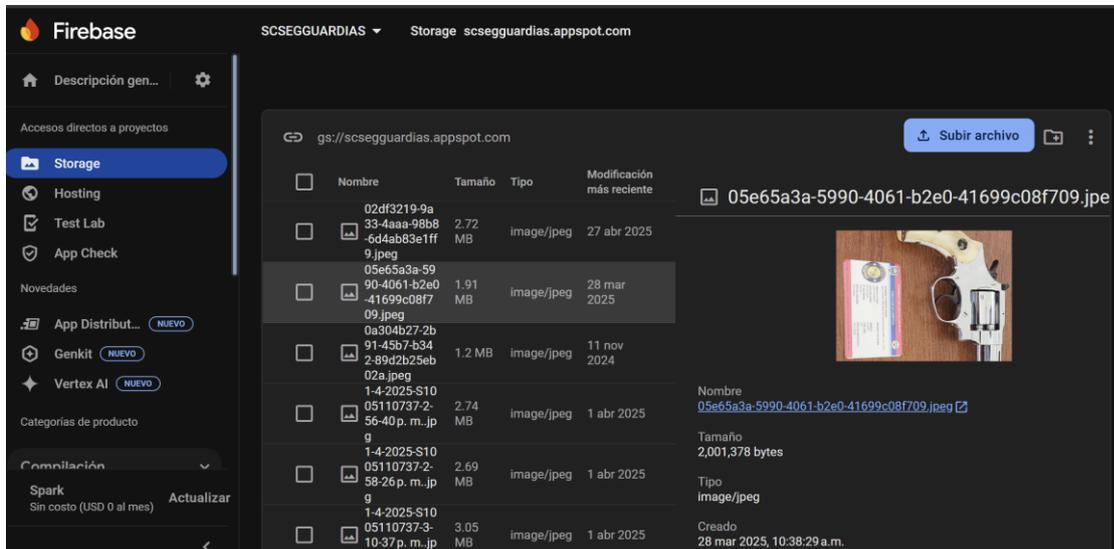
Tabla 11*Atributos de las colecciones de segguardias.*

Colección	Atributos
Usuarios	_id, nombre, apellido, usuario, correo, contraseña, rol, celular, cédula, estado
Registros	_id, fecha, hora, ubicación, imagenUrl, usuario, puestold
Puestos	_id, nombre, ubicación, horario, detalles, minutosCortesía, estado
Multas	_id, descripción, fecha, hora, ubicación, usuario, usuariomultado, puesto, imagenUrl
Incidentes	_id, descripción, fecha, hora, ubicación, usuario, puestold, imagenUrl
Asignaciones	_id, usuariold, puestold, fechaAsignacion
Alertas	_id, descripción, fecha, hora, ubicación, usuariomultado, puesto

Nota. Autoría propia.

Tal como se observa en la **Tabla 11**, entre los atributos registrados se encuentra “imagenUrl”, el cual hace referencia a la ruta donde se almacena la imagen dentro del servidor de contenidos. Esta estructura puede apreciarse con mayor detalle en la **Figura 33**.

Figura 33
Evidencia guardada en FireStorage.



Nota. Autoría propia.

Por consiguiente, es importante señalar que ambos servicios fueron integrados en el proyecto base correspondiente al API REST. Para ello, se desarrollaron archivos de configuración que permiten la conexión con la base de datos, como se muestra en la **Figura 34**.

Figura 34
Configuración para la conexión a la BDD.

```
1 // db.js
2 const { MongoClient, ServerApiVersion } = require('mongodb');
3
4 const uri = "mongodb+srv://[redacted]:[redacted]@segguardias.mongodb.net/?retryWrites=true&appName=segguardias";
5 let client;
6 let database;
7
8 async function connectToDatabase() {
9   if (!client) {
10    client = new MongoClient(uri, {
11      serverApi: { version: ServerApiVersion.v1, strict: true, deprecationErrors: true }
12    });
13    await client.connect();
14    database = client.db('segguardias');
15    console.log("Conexión establecida a MongoDB");
16   }
17   return database;
18 }
19
20 module.exports = connectToDatabase;
21
```

Nota. Autoría propia.

Adicionalmente, se incluyó la configuración necesaria para establecer la conexión con Firebase, permitiendo así la carga de imágenes al servicio, como se evidencia en la **Figura 35**.

Figura 35
Configuración para la conexión y subida a FireStorage.

```
JS firebase.js > uploadImageToFirebase > <function> > stream.on('finish') callback
1 // firebase.js
2 const admin = require('firebase-admin');
3 const serviceAccount = require('./scseguardias-firebase-adminsdk-9s9ir-8faf9c0a7b.json');
4
5 admin.initializeApp({
6   credential: admin.credential.cert(serviceAccount),
7   storageBucket: "scseguardias.appspot.com"
8 });
9
10 const bucket = admin.storage().bucket();
11
12 async function uploadImageToFirebase(file) {
13   const blob = bucket.file(file.originalname);
14   const stream = blob.createWriteStream({ metadata: { contentType: file.mimetype } });
15
16   return new Promise((resolve, reject) => {
17     stream.on('error', reject);
18     stream.on('finish', () => {
19       const publicUrl = `https://firebasestorage.googleapis.com/v0/b/${bucket.name}/o/${encodeURIComponent(file.originalname)}`;
20       resolve(publicUrl);
21     });
22     stream.end(file.buffer);
23   });
24 }
25
26 module.exports = { uploadImageToFirebase };
27
```

Nota. Autoría propia.

Posteriormente, tal como se detalló en la **Tabla 10**, se llevó a cabo el desarrollo del módulo de administración, incorporando funcionalidades como el registro de nuevos usuarios, visualización de usuarios registrados, edición de datos de usuario, desactivación de usuarios, registro y listado de puestos, asignación de puestos, así como la gestión de evidencias (ingresos, multas e incidentes). Estas funcionalidades son esenciales para garantizar el flujo adecuado de información en la aplicación móvil.

Figura 36
Configuración para la implementación de la API de Google Maps.

```
JS googleMapsConfig.js
src > components > JS googleMapsConfig.js > googleMapsConfig
1 // googleMapsConfig.js
2 export const googleMapsConfig = {
3   googleMapsApiKey: '...', // Clave de google maps
4   libraries: ['places', 'maps'], // Incluye todas las librerías necesarias
5   version: 'weekly', // Asegura que la versión sea consistente
6 };
7
```

Nota. Autoría propia.

Para llevar a cabo dicho proceso, fue necesaria la integración de la API de Google Maps, cuya implementación se puede observar en la **Figura 36**.

La primera funcionalidad relevante dentro del módulo de Administración corresponde al control de turnos. Para su implementación, fue necesario habilitar el registro de nuevos usuarios, la visualización de los usuarios existentes, así como la opción de revocar el acceso cuando la situación lo requiera.

Figura 37
API Rest método register.

```
app.post('/register', verificarToken, async (req, res) => {
  const { nombre, apellido, contrasena, correo, rol, celular, cedula }
  = req.body;
  if (!nombre || !apellido || !contrasena || !correo || !rol ||
  !celular || !cedula) {
    return res.status(400).json({ message: 'Nombre, Apellido, Correo,
  Contraseña, Rol, Celular y Cédula son requeridos' });
  }
  const usuario = `${rol.charAt(0).toUpperCase()}${cedula}`;
  try {const db = await connectToDatabase();
    const users = db.collection('Usuarios');
    const existingUser = await users.findOne({ $or: [{ correo }, {
  cedula }] });
    if (existingUser) {return res.status(400).json({ message: 'El
  correo o la cédula ya están registrados' });
    }
    const hashedPassword = await bcrypt.hash(contrasena, 10);
    await users.insertOne({nombre,apellido,usuario,correo,contrasena:
  hashedPassword,rol,celular,cedula,estado: "Activo"});
    return res.status(201).json({ message: 'Usuario registrado
  exitosamente', usuario });
  } catch (error) {
    console.error("Error en el registro:", error);
    return res.status(500).json({ message: 'Error Interno del
  Servidor', error });
  }
});
```

Nota. Autoría propia.

Para llevar a cabo el registro de nuevos usuarios, se implementó un método POST dentro del API REST, tal como se muestra en la **Figura 37**. Este método requiere los datos ingresados en el formulario ilustrado en la **Figura 38**. Una vez completado el formulario en el frontend, el sistema genera automáticamente un campo

de usuario combinando el rol y la cédula del individuo. Por ejemplo, si se trata de un guardia con cédula 12345678, el nombre de usuario generado será G12345678. Este identificador es esencial para que guardias y supervisores puedan acceder a la aplicación móvil. Además, el sistema realiza el proceso de hash sobre la contraseña, como medida de seguridad.

Figura 38
Registro de nuevo usuario en el módulo de Administración.

The screenshot shows a web form titled "Registrar Usuario" with a home icon. The form contains the following fields and values:

- Correo Electrónico *: xroberthrg@gmail.com
- Nombre *: Roberth
- Apellido *: Lima
- Número de Celular *: 0967732818
- Cédula *: 2100572755
- Rol: Supervisor (dropdown menu)
- Contraseña *: [masked]
- Confirmar Contraseña *: [masked]

At the bottom of the form is a blue button labeled "REGISTRAR".

Nota. Autoría propia.

A continuación, se presenta la funcionalidad encargada de obtener y mostrar el listado completo de usuarios almacenados en la base de datos, implementada a través del método mostrado en la **Figura 39**.

Figura 39

API Rest método usuario-consulta.

```
app.get('/usuarios-consulta', verificarToken, async (req, res) => {
  const { rol, estado, limite = 5, pagina = 1 } = req.query;
  try {
    const database = await connectToDatabase();
    const usuariosCollection = database.collection('Usuarios');
    let filtro = {};
    if (rol) filtro.rol = rol;
    if (estado) filtro.estado = estado;
    const totalDocumentos = await usuariosCollection.countDocuments(filtro);
    const usuarios = await usuariosCollection.find(filtro)
      .skip((parseInt(pagina) - 1) * parseInt(limite))
      .limit(parseInt(limite))
      .toArray();
    res.status(200).json({
      usuarios,
      total: totalDocumentos,
      pagina: parseInt(pagina),
      paginasTotales: Math.ceil(totalDocumentos / parseInt(limite)),
    });
  } catch (error) {
    console.error('Error al obtener usuarios:', error);
    res.status(500).json({ message: 'Error interno del servidor' });
  }
});
```

Nota. Autoría propia.

Dado que el sistema permite realizar filtrado y paginación, estas funcionalidades se aplican cuando son requeridas, tal como se ilustra en la **Figura 40**.

Figura 40

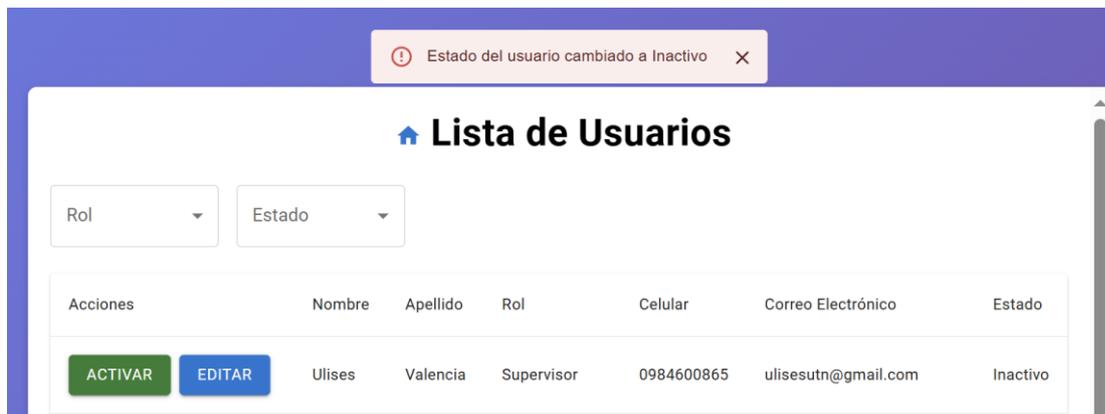
Listado de los usuarios registrados en el Front.

Acciones	Nombre	Apellido	Rol	Celular	Correo Electrónico	Estado
REVOCAR EDITAR	Ulises	Valencia	Supervisor	0984600865	ulisesutn@gmail.com	Activo
REVOCAR EDITAR	Ivan	Valencia	Guardia	0984600865	darktiger_11@hotmail.com	Activo
REVOCAR EDITAR	Alex	Lopez	Administrador	0984600865	uivalencial@utn.edu.ec	Activo
REVOCAR EDITAR	Cesar	Valencia	Guardia	0984600865	ulicesutn@gmail.com	Activo
REVOCAR EDITAR	Roberth	Lima	Guardia	0967732818	xroberthrg@gmail.com	Activo

Nota. Autoría propia.

Asimismo, el sistema permite deshabilitar el acceso de un usuario, es decir, cambiar su estado de activo a inactivo, como se muestra en la **Figura 41**. Esta acción impide que el usuario vuelva a acceder a la aplicación.

Figura 41
Revocar permisos a un usuario.



Nota. Autoría propia.

Finalmente, el sistema también ofrece la opción de editar los datos del usuario en caso de ser necesario, tal como se observa en la **Figura 42**. Esta funcionalidad permite corregir posibles errores en campos como nombre, apellido, número de celular, correo electrónico y estado.

Figura 42
Pantalla de edición de un usuario.

The screenshot shows a form titled "Editar Usuario" with a purple border. It contains several input fields: "Nombre" (Ulises), "Apellido" (Valencia), "Celular" (0984600865), "Correo Electrónico" (ulisesutn@gmail.com), and "Estado" (Activo). A blue button at the bottom is labeled "GUARDAR CAMBIOS".

Nota. Autoría propia.

Por otra parte, se implementa una funcionalidad similar a la de los usuarios, pero orientada a la gestión de puestos. Estos representan los lugares de trabajo que, posteriormente, serán asignados a los usuarios y permitirán registrar su ingreso a través de la aplicación móvil.

Figura 43
Registro de nuevos puestos.

🏠 Crear Puesto

Nombre del Puesto *

SEGPRIC CIA. LTDA

Hora de Entrada *

08:00 🕒

Hora de Salida *

16:00 🕒

Detalles

EMPRESA DE SEGURIDAD PRIVADA

Minutos de Cortesía *

19

Buscar ubicación



USAR MI UBICACIÓN ACTUAL

CREAR PUESTO

Nota. Autoría propia.

En primer lugar, se desarrolló el método correspondiente para el registro de un nuevo puesto, como se muestra en la **Figura 44**. En este proceso se integra la API de Google Maps, con el fin de capturar las coordenadas de geolocalización, las cuales serán utilizadas posteriormente.

Figura 44
API Rest método POST puestos.

```
app.post('/puestos', verificarToken, async (req, res) => {
  const { nombre, ubicacion, horario, detalles, minutosCortesia } =
  req.body;
  if (!nombre || !ubicacion || minutosCortesia === undefined) {
    return res.status(400).json({ message: 'Nombre, ubicación y
minutos de cortesía son requeridos' });
  }
  try {
    const db = await connectToDatabase();
    const puestosCollection = db.collection('Puestos');
    const nuevoPuesto = {
      nombre,
      ubicacion,
      horario,
      detalles,
      minutosCortesia,
      estado: "Activo"
    };
    await puestosCollection.insertOne(nuevoPuesto);
    res.status(201).json({ message: 'Puesto creado exitosamente' });
  } catch (error) {
    console.error("Error al crear el puesto:", error);
    res.status(500).json({ message: 'Error Interno del Servidor' });
  }
});
```

Nota. Autoría propia.

A continuación, se implementó la interfaz para el registro de nuevos puestos, la cual incluye los campos mostrados en la **Figura 43**. Tal como se mencionó previamente, se utiliza la API de Google Maps, esta vez con el propósito de mostrar un mapa interactivo. A través de una bandera ubicada en el mapa, el usuario puede seleccionar fácilmente la ubicación del puesto. También se dispone de un botón que permite obtener automáticamente la ubicación actual, lo cual resulta útil si el registro se realiza directamente en el lugar correspondiente. Un campo adicional relevante es el de los minutos de cortesía, que indica el tiempo permitido antes de considerar un ingreso como tardío.

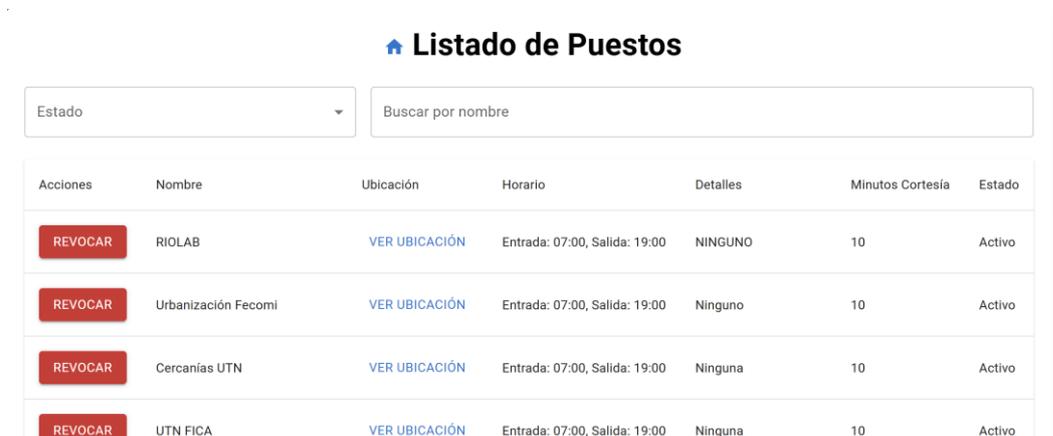
Figura 45
API Rest método GET puestos.

```
app.get('/puestos', verificarToken, async (req, res) => {
  const { estado, limite = 5, pagina = 1 } = req.query;
  try {
    const db = await connectToDatabase();
    const puestosCollection = db.collection('Puestos');
    let filtro = {};
    if (estado) filtro.estado = estado;
    const totalDocumentos = await
puestosCollection.countDocuments(filtro);
    const puestos = await puestosCollection.find(filtro)
      .skip((pagina - 1) * parseInt(limite))
      .limit(parseInt(limite))
      .toArray();
    res.status(200).json({
      puestos,
      total: totalDocumentos,
      pagina: parseInt(pagina),
      paginasTotales: Math.ceil(totalDocumentos / parseInt(limite))
    });
  } catch (error) {
    console.error("Error al obtener los puestos:", error);
    res.status(500).json({ message: 'Error Interno del Servidor' });
  }
});
```

Nota. Autoría propia.

Otra funcionalidad implementada es la visualización del listado de puestos de trabajo, que permite consultar todos los puestos previamente registrados, tal como se muestra en la **Figura 45**. Además, brinda la opción de revocar puestos, lo que impide que puedan ser utilizados desde la aplicación móvil. Finalmente, se incluye la posibilidad de visualizar la ubicación geográfica de cada puesto, como se evidencia en la **Figura 46**.

Figura 46
Pantalla de listado de puestos.

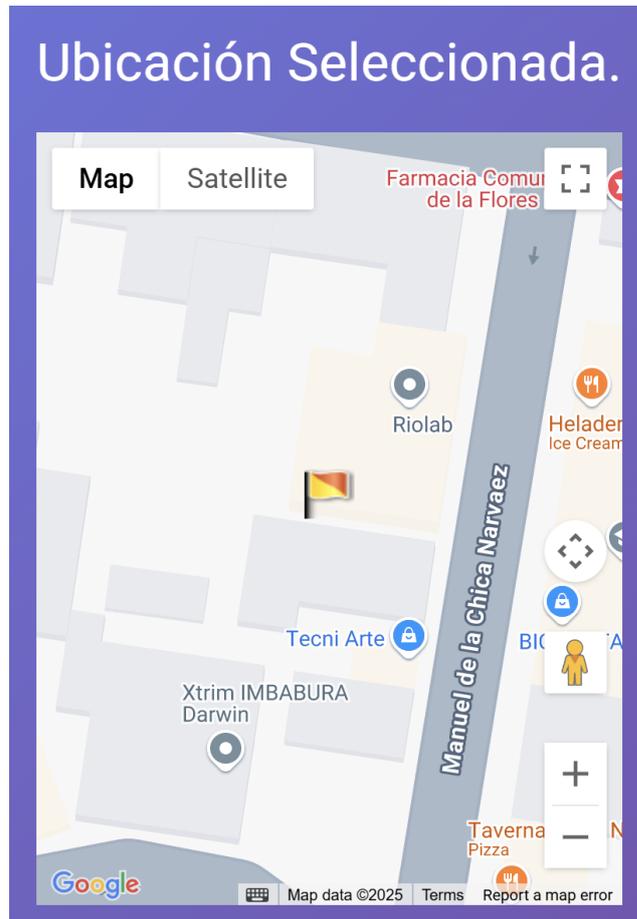


Acciones	Nombre	Ubicación	Horario	Detalles	Minutos Cortesía	Estado
REVOCAR	RIOLAB	VER UBICACIÓN	Entrada: 07:00, Salida: 19:00	NINGUNO	10	Activo
REVOCAR	Urbanización Fecomi	VER UBICACIÓN	Entrada: 07:00, Salida: 19:00	Ninguno	10	Activo
REVOCAR	Cercanías UTN	VER UBICACIÓN	Entrada: 07:00, Salida: 19:00	Ninguna	10	Activo
REVOCAR	UTN FICA	VER UBICACIÓN	Entrada: 07:00, Salida: 19:00	Ninguna	10	Activo

Nota. Autoría propia.

Es importante mencionar que la opción de ver ubicación despliega una pantalla en la cual se presenta la localización del puesto utilizando la API de Google Maps, tal como se observa en la **Figura 47**.

Figura 47
Pantalla de ver ubicación.



Nota. Autoría propia.

A continuación, se presenta la funcionalidad de asignación, la cual permite vincular un puesto previamente registrado con un usuario del sistema. Esta operación se lleva a cabo mediante el método mostrado en la **Figura 48**.

Figura 48

API Rest método GET asignaciones.

```
app.get('/asignaciones', verificarToken, async (req, res) => {
  const { limite = 10, pagina = 1, fechaInicio, fechaFin } = req.query;
  try {const database = await connectToDatabase();
  const asignacionesCollection = database.collection('Asignaciones');
  let filtro = {};
  if (fechaInicio && fechaFin) {filtro.fechaAsignacion = { $gte:
  fechaInicio, $lte: fechaFin }};
  const pipeline = [{ $match: filtro },
  {$lookup: {from: 'Usuarios',localField: 'usuarioId',foreignField:
  '_id',as: 'usuarioDatos',}},},
  {$lookup: {from: 'Puestos',localField: 'puestoId',foreignField: '_id',as:
  'puestoDatos',}},},
  {$project: {_id: 1,fechaAsignacion: 1,usuario: {id: { $arrayElemAt:
  ['$usuarioDatos._id', 0 ]},nombre: { $arrayElemAt:
  ['$usuarioDatos.nombre', 0 ]},apellido: { $arrayElemAt:
  ['$usuarioDatos.apellido', 0 ]},},},puesto: {id: { $arrayElemAt:
  ['$puestoDatos._id', 0 ]},nombre: { $arrayElemAt: ['$puestoDatos.nombre',
  0 ] },},},},},{ $sort: { fechaAsignacion: -1 } },{ $skip: (parseInt(pagina)
  - 1) * parseInt(limite) },{ $limit: parseInt(limite) },,];
  const totalDocumentos = await
  asignacionesCollection.countDocuments(filtro);
  const asignaciones = await
  asignacionesCollection.aggregate(pipeline).toArray();
  res.status(200).json({asignaciones,total: totalDocumentos,pagina:
  parseInt(pagina),paginasTotales: Math.ceil(totalDocumentos /
  parseInt(limite)),});
  } catch (error) {console.error('Error al obtener asignaciones:', error);
  res.status(500).json({ message: 'Error interno del servidor' });}});
```

Nota. Autoría propia.

Tal como se muestra en la **Figura 49**, el proceso de asignación consiste en seleccionar un usuario activo y un puesto igualmente habilitado. Esta asignación permitirá que el usuario designado tenga acceso a la aplicación móvil y pueda registrar su ingreso al puesto correspondiente.

Figura 49
Pantalla de la gestión de asignaciones.

The screenshot shows a web interface for 'Gestión de Asignaciones'. At the top, there is a dark blue header with a home icon, the text 'Gestión de Asignaciones', and two links: 'ASIGNAR' and 'HISTORIAL'. Below the header, the main content area is titled 'Asignar Puesto a Usuario'. It features two dropdown menus: 'Usuario' with the selected value 'Ulises Valencia' and 'Puesto' with the selected value 'Urbanización Fe...'. Below these is a prominent blue button labeled 'ASIGNAR PUESTO'.

Nota. Autoría propia.

Asimismo, se dispone de una sección destinada a consultar el historial de asignaciones, la cual permite verificar la fecha en que se realizó la asignación de un puesto a un usuario, como se ilustra en la **Figura 50**.

Figura 50
Pantalla de la historial de asignaciones.

The screenshot shows the 'Historial de Asignación' page. It has the same header as Figure 49. Below the header, there is a title 'Historial de Asignación' and four filter boxes: 'Usuario', 'Puesto', 'Fecha Inicio' (with a date format 'dd/mm/aaaa' and a calendar icon), and 'Fecha Fin' (with a date format 'dd/mm/aaaa' and a calendar icon). Below the filters is a table with the following data:

Usuario	Puesto	Fecha	Detalles
Ivan Valencia	Cercanías UTN	2025-04-28	VER
Ivan Valencia	UTN FICA	2025-04-28	VER
Robert Lima	Alpreseg Guayaquil	2025-04-23	VER
Alex Lopez	Cercanías UTN	2025-03-31	VER
Ulises Valencia	Cercanías UTN	2025-03-31	VER

Nota. Autoría propia.

Adicionalmente, tal como se evidencia en la **Figura 51**, se incluye una funcionalidad que permite visualizar detalles específicos de cada registro, facilitando así la verificación de información adicional relevante.

Figura 51
Pantalla de detalles de asignación.

Usuario		Puesto	
Estado:	Activo	Nombre:	Cercanías UTN
Nombre:	Ivan Valencia	Horario:	Entrada: 07:00, Salida: 19:00
Rol:	Guardia	Detalles:	Ninguna
		Ubicación:	VER UBICACIÓN

Nota. Autoría propia.

Finalmente, se incorpora la funcionalidad de visualización de evidencias, la cual permite al administrador acceder a todos los datos recopilados desde la aplicación móvil. Esta opción incluye el registro de ingresos, multas e incidentes generados por los usuarios.

Figura 52

API Rest método GET registro-consulta.

```
app.get('/registros-consulta', verificarToken, async (req, res) => {
  const {usuario, fechaInicio, fechaFin, puesto, limite = 5, pagina = 1,} =
  req.query;
  try {const database = await connectToDatabase();
  const registrosCollection = database.collection('Registros');
  const { ObjectId } = require('mongodb');
  let filtro = {};
  if (usuario) {filtro['usuario'] = { $regex: usuario, $options: 'i' }};
  if (fechaInicio && fechaFin) {filtro['fecha'] = { $gte: fechaInicio,
  $lte: fechaFin }};
  if (puesto) {try {filtro['puestoId'] = new ObjectId(puesto);} catch
  (error) {console.error('PuestoId no es un ObjectId válido:', error);
  return res.status(400).json({ message: 'Puesto inválido'}});}
  const pipeline = [{ $match: filtro },
  {$lookup: {from: 'Usuarios', localField: 'usuario', foreignField:
  'usuario', as: 'datosUsuario',}},,
  {$lookup: {from: 'Puestos', localField: 'puestoId', foreignField: '_id', as:
  'datosPuesto',}},,
  {$project: {_id: 1, fecha: 1, hora: 1, ubicacion: 1, imagenUrl: { $ifNull:
  ['$imagenUrl', 'No agregada'] }, usuario: {$concat: [
  { $arrayElemAt: ['$datosUsuario.nombre', 0] }, ' ',
  { $arrayElemAt: ['$datosUsuario.apellido', 0] }, ],},,
  puesto: { $arrayElemAt: ['$datosPuesto.nombre', 0] },},},,
  { $sort: { fecha: -1, hora: -1 } }, { $skip: (parseInt(pagina) - 1) *
  parseInt(limite) }, { $limit: parseInt(limite) },];
  const totalDocumentos = await
  registrosCollection.countDocuments(filtro);
  const registros = await
  registrosCollection.aggregate(pipeline).toArray();
  res.status(200).json({registros, total: totalDocumentos, pagina:
  parseInt(pagina), paginasTotales: Math.ceil(totalDocumentos /
  parseInt(limite)),});} catch (error) {console.error('Error al obtener los
  registros:', error);res.status(500).json({ message: 'Error Interno del
  Servidor' });
  }});
```

Nota. Autoría propia.

En primer lugar, como se observa en la **Figura 53**, se presenta la funcionalidad de control de turnos o ingresos, la cual muestra toda la evidencia de registro del personal en sus respectivos puestos de trabajo, a través del método representado en la **Figura 52**. Además, el administrador tiene la posibilidad de visualizar la ubicación, como se visualiza en la **Figura 47**, y acceder a la imagen asociada, la cual se abre directamente desde el almacenamiento en FireStorage.

Figura 53
Pantalla de evidencias-control turnos.

Usuario	Fecha	Hora	Puesto	Ubicación	Imagen
Ulises Valencia	2025-04-28	11:50:02	UTN FICA	VER UBICACIÓN	VER IMAGEN
Ivan Valencia	2025-04-28	11:49:08	UTN FICA	VER UBICACIÓN	VER IMAGEN
Ulises Valencia	2025-04-28	11:42:50	UTN FICA	VER UBICACIÓN	VER IMAGEN
Ulises Valencia	2025-04-25	18:56:50	Urbanización Fecomi	VER UBICACIÓN	VER IMAGEN
Ulises Valencia	2025-04-25	18:22:36	Urbanización Fecomi	VER UBICACIÓN	VER IMAGEN

Nota. Autoría propia.

De igual manera, se realiza el listado de multas e incidentes, pero teniendo métodos y pantallas diferentes.

Figura 54
API Rest método GET multas-consulta.

```
app.get('/multas-consulta', verificarToken, async (req, res) => {
  const { supervisor, multado, fechaInicio, fechaFin, puesto, limite = 5,
  pagina = 1 } = req.query;
  try {const db = await connectToDatabase();
  const col = db.collection('Multas');
  const { ObjectId } = require('mongodb');
  let filtro = {};
  if (supervisor) filtro.usuario = { $regex: supervisor, $options: 'i' };
  if (multado) filtro.usuariomultado = { $regex: multado, $options: 'i' };
  if (fechaInicio && fechaFin) filtro.fecha = { $gte: fechaInicio, $lte:
  fechaFin };
  if (puesto) try { filtro.puesto = new ObjectId(puesto); } catch { return
  res.status(400).json({ message: 'Puesto inválido' }); }
  const pipeline = [
  { $match: filtro },
  { $lookup: { from: 'Usuarios', localField: 'usuario', foreignField:
  'usuario', as: 'datosSupervisor' } },
  { $lookup: { from: 'Usuarios', localField: 'usuariomultado',
  foreignField: 'usuario', as: 'datosMultado' } },
  { $lookup: { from: 'Puestos', localField: 'puesto', foreignField: '_id',
  as: 'datosPuesto' } },
  { $project: { _id: 1, descripcion: 1, fecha: 1, hora: 1, ubicacion: 1,
  imagenUrl: { $ifNull: ['$imagenUrl', 'No agregada'] },
  supervisor: { $concat: [ { $arrayElemAt: ['$datosSupervisor.nombre', 0]
  }, ' ', { $arrayElemAt: ['$datosSupervisor.apellido', 0] } ] },
  multado: { $concat: [ { $arrayElemAt: ['$datosMultado.nombre', 0] }, '
  ', { $arrayElemAt: ['$datosMultado.apellido', 0] } ] },
  puesto: { $arrayElemAt: ['$datosPuesto.nombre', 0] } } },
  { $sort: { fecha: -1, hora: -1 } },
  { $skip: (parseInt(pagina) - 1) * parseInt(limite) },
  { $limit: parseInt(limite) } ];
  const total = await col.countDocuments(filtro);
  const multas = await col.aggregate(pipeline).toArray();
  res.status(200).json({ multas, total, pagina: parseInt(pagina),
  paginasTotales: Math.ceil(total / parseInt(limite)) });} catch (e)
  {console.error('Error al obtener las multas:', e);
  res.status(500).json({ message: 'Error Interno del Servidor' });}};
```

Nota. Autoría propia.

En el caso de multas, para el método se utilizó lo ilustrado en la **Figura 54** y como pantalla, lo visualizado en la **Figura 55**.

Figura 55
Pantalla de evidencias-multas

Supervisor	Multado	Descripción	Fecha	Hora	Puesto	Ubicación	Imagen
Ulises Valencia	Ivan Valencia	Llegada tarde	2025-04-28	11:49:08	UTN FICA	VER UBICACIÓN	No agregada
Ulises Valencia	Ivan Valencia	Prueba	2025-04-28	11:43:27	UTN FICA	VER UBICACIÓN	VER IMAGEN
Ulises Valencia	Ivan Valencia	Prueba ubicación	2025-04-25	17:02:26	Urbanización Fecomi	VER UBICACIÓN	VER IMAGEN
Ulises Valencia	Ivan Valencia	Prueba incidente	2025-04-25	16:07:10	Hospital Del Seguro	VER UBICACIÓN	VER IMAGEN
Ulises Valencia	Robert Lima	Prueba backend	2025-04-23	11:47:36	Urbanización Fecomi	VER UBICACIÓN	VER IMAGEN

Nota. Autoría propia.

Figura 56
API Rest método GET incidentes-consulta.

```

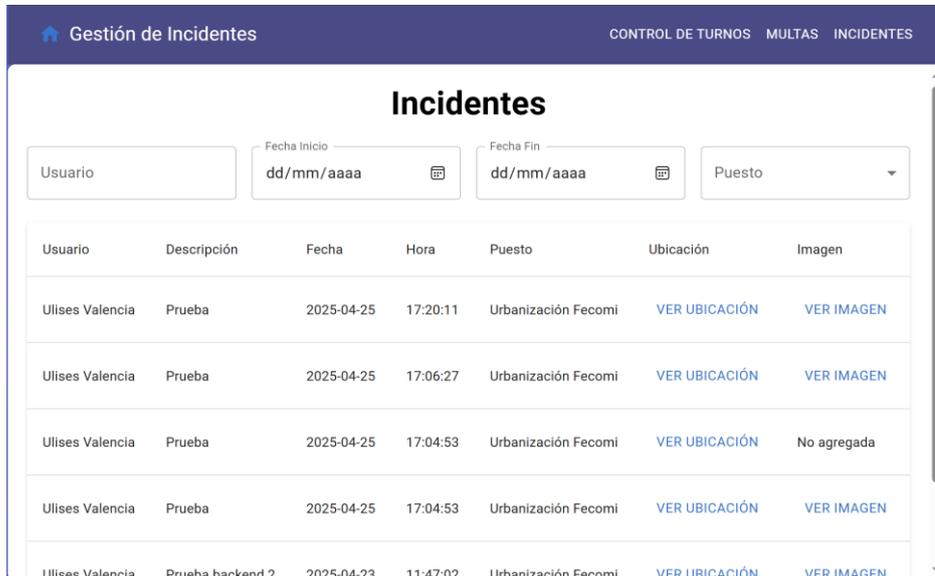
app.get('/incidentes-consulta', verificarToken, async (req, res) => {
  const { usuario, fechaInicio, fechaFin, puesto, limite = 5, pagina = 1 }
  = req.query;
  try {const db = await connectToDatabase();
  const col = db.collection('Incidentes');
  const { ObjectId } = require('mongodb');
  let filtro = {};
  if (usuario) filtro.usuario = { $regex: usuario, $options: 'i' };
  if (fechaInicio && fechaFin) filtro.fecha = { $gte: fechaInicio, $lte:
  fechaFin };
  if (puesto) try { filtro.puestoId = new ObjectId(puesto); } catch {
  return res.status(400).json({ message: 'Puesto inválido' }); }
  const pipeline = [
  { $match: filtro },
  { $lookup: { from: 'Usuarios', localField: 'usuario', foreignField:
  'usuario', as: 'datosUsuario' } },
  { $lookup: { from: 'Puestos', localField: 'puestoId', foreignField:
  '_id', as: 'datosPuesto' } },
  { $project: { _id: 1, descripcion: 1, fecha: 1, hora: 1, ubicacion:
  1, imagenUrl: { $ifNull: ['$imagenUrl', 'No agregada'] },
  usuario: { $concat: [ { $arrayElemAt: ['$datosUsuario.nombre', 0] }, ' ',
  { $arrayElemAt: ['$datosUsuario.apellido', 0] } ] }, puesto: {
  $arrayElemAt: ['$datosPuesto.nombre', 0] } } },
  { $sort: { fecha: -1, hora: -1 } },
  { $skip: (parseInt(pagina) - 1) * parseInt(limite) },
  { $limit: parseInt(limite) }];
  const total = await col.countDocuments(filtro);
  const incidentes = await col.aggregate(pipeline).toArray();
  res.status(200).json({ incidentes, total, pagina: parseInt(pagina),
  paginasTotales: Math.ceil(total / parseInt(limite)) });} catch (e) {
  console.error('Error al obtener los incidentes:', e);
  res.status(500).json({ message: 'Error Interno del Servidor' });}});

```

Nota. Autoría propia.

Y en el caso de incidentes, para el método se utilizó lo ilustrado en la **Figura 56** y como pantalla, lo visualizado en la **Figura 57**.

Figura 57
Pantalla de evidencias-incidentes.



Nota. Autoría propia.

La **Tabla 12**, muestra la reunión de retrospectiva, que ilustra las lecciones aprendidas, las acciones propuestas, además de los aspectos positivos y aspectos negativos. Este documento es indispensable para la mejora continua y el proceso de desarrollo del proyecto.

Tabla 12
Retrospectiva del Sprint 0

Reunión de retrospectiva				
Sprint: 0				
Fecha: 02/12/2024				
Actividad / Historia de usuario	Aspectos positivos	Aspectos negativos	Lecciones aprendidas	Acciones propuestas
HU-01 Control de turnos de guardias de seguridad	Se logró implementar correctamente la API de Google Maps. El backend y la base de datos funcionaron de forma estable.	Se presentaron dificultades al integrar Google Maps y Firebase Storage. Hubo demoras en la prueba de geolocalización.	La integración de APIs externas requiere mayor planificación y pruebas anticipadas.	Probar módulos críticos desde el inicio del Sprint. Documentar mejor los pasos para configurar Firebase.

HU-02 Gestión del registro de personal activo. El módulo de creación y edición de usuarios quedó completamente funcional. Hubo confusión inicial con los campos requeridos y validaciones. Es fundamental validar desde el análisis todos los datos obligatorios y reglas de negocio. Hacer prototipos funcionales más temprano y validarlos con los requerimientos de reales antes de desarrollar.

Nota. Autoría propia.

2.3.3. Sprint 1

En la **Tabla 13** muestra la planificación correspondiente al Sprint 1, en la cual se detallan las historias de usuario asignadas, las tareas asociadas y las respectivas estimaciones de tiempo. Este registro resulta fundamental para dar seguimiento al proyecto.

Tabla 13

Sprint 1

SPRINT BACKLOG				
Sprint: 1				
Total horas: 144				
Fecha Inicio SP1: 03/12/2024				
Fecha Final SP1: 27/04/2025				
Actividad / Historia de usuario	Fase Desarrollo	TAREA	TIEMPO ESTIMADO (Horas)	TIEMPO REAL (Horas)
HU -03 Gestión del puesto de guardia.	Análisis	Análisis de metodología para manejar el perímetro.	4	5
	Análisis	Revisión de requerimientos para guardias	3	3
	Análisis	Análisis de evidencias: imagen, hora, ubicación.	3	4
	Desarrollo	Desarrollo de pantalla de inicio y autenticación.	5	5
	Desarrollo	Implementación de captura de ingreso con foto y geolocalización.	8	9
	Desarrollo	Desarrollar la lógica de tiempo de llegada.	6	7

	Desarrollo	Implementación del método de Haversine, para el perímetro.	7	8
	Desarrollo	Desarrollo del módulo de incidentes para guardias.	6	7
	Desarrollo	Pruebas funcionales y ajustes del módulo de guardias.	5	6
HU- 04 Gestión de los puestos a supervisar.	Análisis	Revisión de flujos de supervisores y asignaciones.	3	3
	Análisis	Definición de lógica para multas y reportes diarios.	3	4
	Desarrollo	Implementación de inicio de sesión de supervisor.	4	4
	Desarrollo	Implementación del módulo de incidentes.	5	5
	Desarrollo	Desarrollo del módulo de creación de multas.	6	7
	Desarrollo	Generación y envío automático de memo al correo.	6	6
	Desarrollo	Visualización de evidencias (foto, ubicación, hora).	5	6
	Desarrollo	Desarrollo del control de alertas.	6	6
	Desarrollo	Desarrollo de la generación de reportes y envío al correo formato.xlsx.	8	9
	Reuniones	Gestión	Planificación	1
Gestión		Diarias	2	2
Gestión		Revisión	1	1
Gestión		Retrospectiva	1	1
TOTAL			144	156

Nota. Autoría propia.

Como una de las primeras prioridades del desarrollo, se implementó la funcionalidad de inicio de sesión. Para ello, se reutilizó el mismo método empleado en el módulo del administrador, con la diferencia de que se validó el tipo de usuario,

verificando si corresponde a un guardia o a un supervisor. Esta implementación se muestra en la **Figura 58**.

Figura 58

API Rest método POST login.

```
app.post('/login', async (req, res) => {
  const { usuario, contraseña } = req.body;
  if (!usuario || !contraseña)
    return res.status(400).json({ message: 'usuario y contraseña son requeridos' });
  try {const db = await connectToDatabase();
    const users = db.collection('Usuarios');
    const user = await users.findOne({$or: [{ usuario }, { correo: usuario }]});
    if (user && await bcrypt.compare(contraseña, user.contraseña)) {
      const token = jwt.sign(
        { id: user._id, usuario: user.usuario, rol: user.rol
        },process.env.JWT_SECRET,{ expiresIn: process.env.JWT_EXPIRES || '1h' });
      return res.status(200).json({message: 'Login exitoso',rol:
      user.rol,token});}
    res.status(401).json({ message: 'Usuario o contraseña inválidos' });}
  catch (error) {console.error("Error en login:", error);
    res.status(500).json({ message: 'Error interno del servidor' });}});
```

Nota. Autoría propia.

Para establecer el área de validación que permitirá el ingreso al sistema, se utilizó el método de Haversine, el cual permite calcular la distancia entre dos coordenadas geográficas. En este caso, se definió un perímetro de 50 metros como rango permitido. Esta lógica fue implementada en el frontend de la aplicación móvil, como se visualiza en la **Figura 59**.

Figura 59

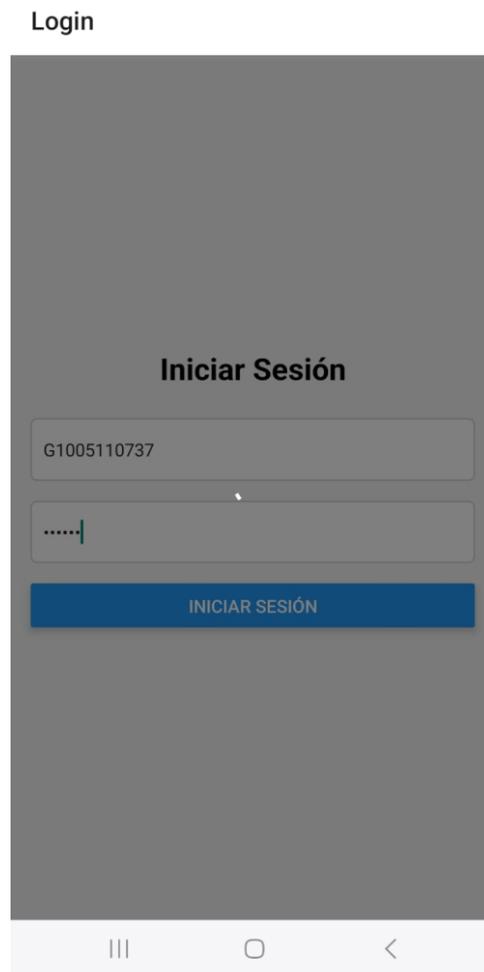
Método para calcular el perímetro para el ingreso.

```
const calcularDistancia = (ubicacion1, ubicacion2) => {
  const R = 6371e3; // Radio de la Tierra en metros
  const  $\phi_1$  = ubicacion1.latitude * Math.PI / 180;
  const  $\phi_2$  = ubicacion2.latitude * Math.PI / 180;
  const  $\Delta\phi$  = (ubicacion2.latitude - ubicacion1.latitude) * Math.PI / 180;
  const  $\Delta\lambda$  = (ubicacion2.longitude - ubicacion1.longitude) * Math.PI / 180;
  const a = Math.sin( $\Delta\phi$  / 2) * Math.sin( $\Delta\phi$  / 2) +Math.cos( $\phi_1$ ) *
  Math.cos( $\phi_2$ ) *Math.sin( $\Delta\lambda$  / 2) * Math.sin( $\Delta\lambda$  / 2);
  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  return R * c;};
```

Nota. Autoría propia.

Una vez completado este proceso, tal como lo visto en la **Figura 60**, el sistema permite el acceso a la aplicación móvil, cuya interfaz de inicio de sesión se demuestra en la **Figura 61**.

Figura 60
Pantalla de login en el dispositivo móvil.



Nota. Autoría propia.

Figura 61
Pantalla de registro de turno.



Nota. Autoría propia.

Una vez dentro de la aplicación, como se muestra en la **Figura 61**, el usuario puede registrar su ingreso al puesto mediante la captura de una fotografía. Este proceso se realiza a través del método representado en la **Figura 62**, el cual almacena la evidencia fotográfica junto con la fecha, la hora y las coordenadas de geolocalización.

Figura 62

Método para subir datos de registro de turno.

```
app.post('/upload-data', upload.single('image'), verificarToken, async (req, res) => {
  const { fecha, hora, ubicacion, usuario, puestoId } = req.body;
  if (!fecha || !hora || !ubicacion || !req.file || !usuario || !puestoId) {
    return res.status(400).json({ message: 'Fecha, Hora, Ubicación, Imagen, Usuario y Puesto son requeridos' });
  }
  try {const database = await connectToDatabase();
    const usuariosCollection = database.collection('Usuarios');
    const puestosCollection = database.collection('Puestos');
    const [usuarioExistente, puestoExistente] = await Promise.all([
      usuariosCollection.findOne({ usuario }),
      puestosCollection.findOne({ _id: new ObjectId(puestoId) })]);
    if (!usuarioExistente) {
      return res.status(404).json({ message: 'Usuario no encontrado' });
    }
    if (!puestoExistente) {
      return res.status(404).json({ message: 'Puesto no encontrado' });
    }
    const publicUrl = await uploadImageToFirebase(req.file);
    const collection = database.collection('Registros');
    await collection.insertOne({fecha, hora, ubicacion, imageUrl: publicUrl, usuario, puestoId: new ObjectId(puestoId)});
    return res.status(201).json({ message: 'Datos subidos con éxito', url: publicUrl });} catch (error) {
    console.error("Error al subir datos:", error);
    return res.status(500).json({ message: 'Error Interno del Servidor', error });}});
```

Nota. Autoría propia.

Es importante señalar que también se implementa la verificación de los minutos de cortesía. En el caso de los usuarios con rol de guardia, si se supera el tiempo permitido, el sistema genera automáticamente una alerta por retraso. Esta lógica se encuentra representada en la **Figura 63**, y adicionalmente, se muestra una notificación en pantalla indicando la tardanza, como se observa en la **Figura 64**.

Figura 63

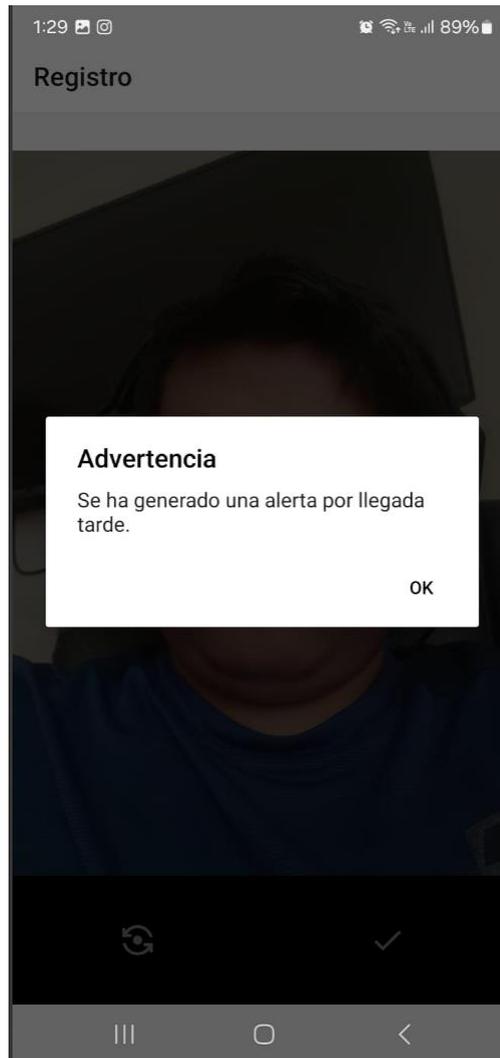
Método para crear alertas en caso de llegar tarde.

```
app.post('/alertas/crear', verificarToken, async (req, res) => {
  const { descripcion, fecha, hora, ubicacion, usuariomultado, puesto } = req.body;
  try {const db = await connectToDatabase();
    const alertasCollection = db.collection('Alertas');
    const puestoId = puesto ? new ObjectId(puesto) : null;
    const nuevaAlerta = {
      descripcion, fecha, hora, ubicacion, usuariomultado, puesto: puestoId, };
    const result = await alertasCollection.insertOne(nuevaAlerta);
    res.status(201).json({ message: 'Alerta creada exitosamente', alertaId: result.insertedId });} catch (error) {
    console.error('Error al crear la alerta:', error);
    res.status(500).json({ message: 'Error en el servidor', error });}});
```

Nota. Autoría propia.

Figura 64

Generación de alertas por llegar tarde.



Nota. Autoría propia.

Para la gestión de alertas, el supervisor dispone de una sección específica en la aplicación. Desde allí, puede confirmar una alerta lo que genera automáticamente una multa o eliminarla en caso de que exista una justificación válida por parte del guardia. Esta funcionalidad se representa en la **Figura 65**.

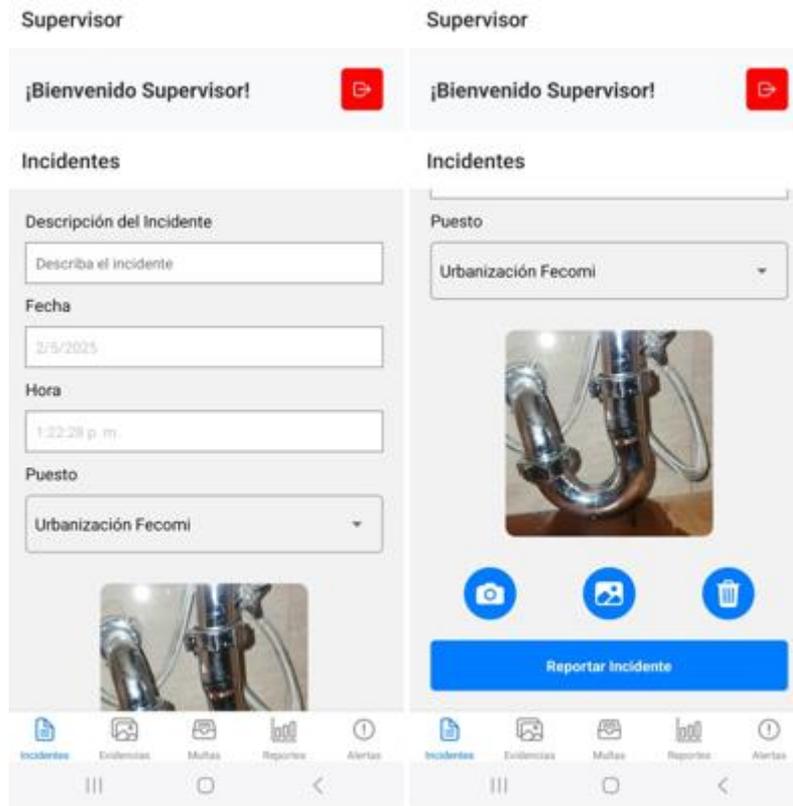
Figura 65
Pantalla de administración de alertas por llegar tarde.



Nota. Autoría propia.

Una vez superado el proceso de registro de turnos, se habilitan nuevos apartados dentro de la aplicación. Por defecto, el primer módulo visible es el de incidentes, el cual permite registrar nuevos eventos incluyendo una descripción, la fecha, la hora y el puesto en el que ocurrieron. De manera opcional, el usuario puede capturar una fotografía o adjuntar una imagen desde la galería. Esta funcionalidad se encuentra ilustrada en la **Figura 66**, mientras que su lógica de funcionamiento se presenta en la **Figura 67**.

Figura 66
Pantalla de incidentes.



Nota. Autoría propia.

Figura 67
Método para crear incidentes.

```

app.post('/report-incident', upload.single('imagen'), verificarToken,
async (req, res) => {
const { descripcion, fecha, hora, ubicacion, usuario, puestoId } =
req.body;
if (!descripcion || !fecha || !hora || !ubicacion || !usuario ||
!puestoId) {
return res.status(400).json({ message: 'Descripción, fecha, hora,
ubicación, usuario y puestoId son requeridos' });}
try {const database = await connectToDatabase();
const usuariosCollection = database.collection('Usuarios');
const puestosCollection = database.collection('Puestos');
const incidentes = database.collection('Incidentes');
const [usuarioExistente, puestoExistente] = await
Promise.all([usuariosCollection.findOne({ usuario
}),puestosCollection.findOne({ _id: new ObjectId(puestoId) })]);
if (!usuarioExistente) {
return res.status(404).json({ message: 'Usuario no encontrado' });}
if (!puestoExistente) {return res.status(404).json({ message: 'Puesto no
encontrado' });}
let imagenUrl = null;
if (req.file) {imagenUrl = await uploadImageToFirebase(req.file); }
const nuevoIncidente =
{descripcion, fecha, hora, ubicacion, usuario, puestoId: new
ObjectId(puestoId), imagenUrl,};
await incidentes.insertOne(nuevoIncidente);
return res.status(201).json({ message: 'Incidente registrado con
éxito' });} catch (error) {console.error('Error al registrar el
incidente:', error);
return res.status(500).json({ message: 'Error Interno del Servidor', error
});}});

```

Nota. Autoría propia.

El módulo siguiente corresponde al listado de evidencias, el cual agrupa los tres tipos registrados: multas, incidentes e ingresos. Al igual que en la versión web, permite visualizar toda la información recopilada, aplicando filtros por tipo de evidencia, puesto, usuario, fecha de inicio y fecha de finalización. Esta funcionalidad se muestra en la **Figura 68**.

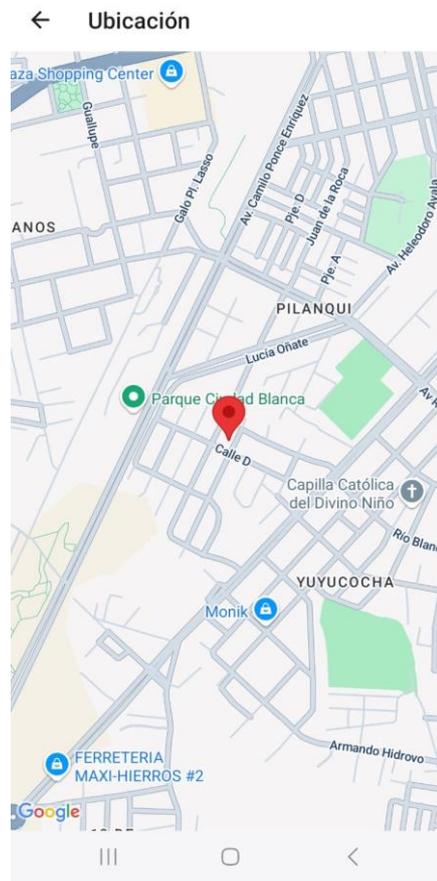
Figura 68
Pantalla evidencias.



Nota. Autoría propia.

Según el tipo de evidencia seleccionado, se emplea un método específico para la obtención de los datos. Estos métodos son los mismos que se utilizan en el frontend web: para incidentes, el método se encuentra representado en la **Figura 56**; para multas, en la **Figura 54**; y para ingresos, en la **Figura 52**.

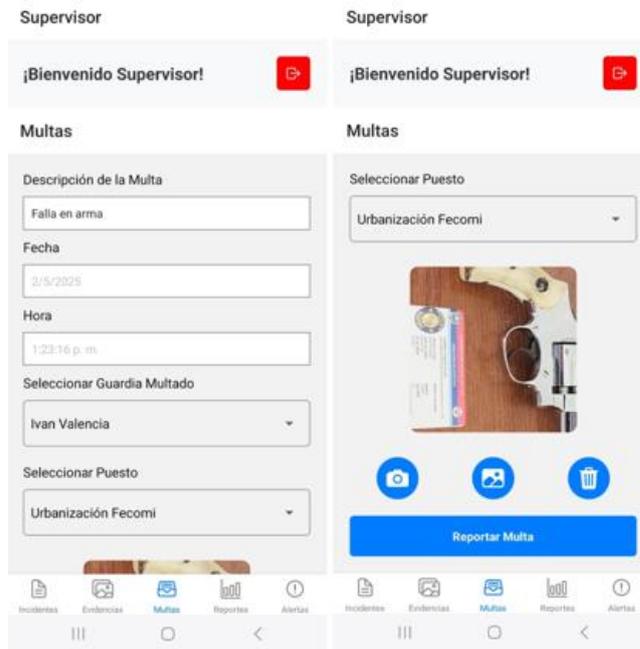
Figura 69
Pantalla de visualización del mapa de la evidencia.



Nota. Autoría propia.

Al igual que en el módulo de incidentes, la pantalla destinada al registro de multas permite ingresar información relacionada con el evento. Sin embargo, a diferencia de los incidentes, se requiere adicionalmente especificar el usuario sancionado. Esta funcionalidad se representa en la **Figura 70**.

Figura 70
Pantalla de creación de multas.



Nota. Autoría propia.

El procedimiento para generar una nueva multa se encuentra detallado en el método representado en la **Figura 71**, el cual utiliza los datos ingresados para registrar correctamente la sanción correspondiente.

Figura 71
Método para crear multas.

```
app.post('/report-multa', upload.single('imagen'), verificarToken, async
(req, res) => {const { descripcion, fecha, hora, ubicacion, usuario,
usuarioMultado, puesto } = req.body;
if (!descripcion || !fecha || !hora || !ubicacion || !usuario ||
!usuarioMultado || !puesto) {
return res.status(400).json({ message: 'Descripción, fecha, hora,
ubicación, usuario, usuarioMultado y puesto son requeridos' });}
try {const database = await connectToDatabase();
const usuariosCollection = database.collection('Usuarios');
const puestosCollection = database.collection('Puestos');
const multasCollection = database.collection('Multas');
const [usuarioExistente, usuarioMultado, puestoExistente] = await
Promise.all([usuariosCollection.findOne({ usuario
}),usuariosCollection.findOne({ usuario: usuarioMultado
}),puestosCollection.findOne({ _id: new ObjectId(puesto) })]);
if (!usuarioExistente) {
return res.status(404).json({ message: 'Usuario no encontrado' });}if
(!usuarioMultado) {
return res.status(404).json({ message: 'Usuario multado no encontrado'
});}if (!puestoExistente) {
return res.status(404).json({ message: 'Puesto no encontrado' });}
let imagenUrl = null;
if (req.file) {imagenUrl = await uploadImageToFirebase(req.file);}
const nuevaMulta =
{descripcion,fecha,hora,ubicacion,usuario,usuarioMultado,puesto: new
ObjectId(puesto),imagenUrl};
await multasCollection.insertOne(nuevaMulta);
await enviarCorreoConPDF({ descripcion, fecha, hora, ubicacion,
imagenUrl, usuarioMultado, puesto },usuarioExistente,'*****@gmail.com');
return res.status(201).json({ message: 'Multa registrada y correo enviado
con éxito' });}
} catch (error) {console.error('Error al registrar la multa:', error);
return res.status(500).json({ message: 'Error Interno del Servidor',
error });}});
```

Nota. Autoría propia.

Además de registrar la multa, como se muestra en la **Figura 71**, el sistema envía automáticamente un archivo tipo memo a un correo electrónico predeterminado. Esto permite al jefe operativo revisar la información contenida en la solicitud y determinar si es necesario emitir un memo oficial. Un ejemplo de dicho archivo se presenta en la **Figura 72**.

Figura 72
Ejemplo de una solicitud de memo enviada al correo.



Nota. Autoría propia.

De manera similar, en el apartado de reportes se solicita al usuario ingresar una fecha de inicio y de fin, además de seleccionar un guardia, un puesto y las tablas que desea incluir en el informe. Esta funcionalidad se encuentra ilustrada en la **Figura 73**.

Figura 73
Pantalla de reportes.

Supervisor

¡Bienvenido Supervisor!

Reportes

Fecha Fin

2/5/2025

Seleccionar Guardia

Ulises Valencia

Seleccionar Puesto

Urbanización Fecomi

Tablas

Registros

Incidentes

Multas

Generar Reporte

Incidentes Evidencias Multas Reportes Alertas

Nota. Autoría propia.

El método empleado para la generación del reporte se muestra en la **Figura 74**. Dado que el informe puede incluir múltiples tablas, como registros de incidentes y multas, se crea un archivo en formato Excel que consolida la información seleccionada según los parámetros definidos por el usuario.

Figura 74
Método para crear reportes.

```
app.post('/generar-reporte-final', verificarToken, async (req, res) => {
  try {const db = await connectToDatabase();
  const Registro = db.collection('Registros'), Incidente =
  db.collection('Incidentes'), Multa = db.collection('Multas'), Puesto =
  db.collection('Puestos'), Usuario = db.collection('Usuarios');
  const { fechaInicio, fechaFin, puestoId, usuario, tablas } = req.body;
  const fechaInicioDate = new Date(fechaInicio), fechaFinDate = new
  Date(fechaFin); fechaFinDate.setHours(23, 59, 59);
  const workbook = new ExcelJS.Workbook();
  const getNombrePuesto = async (id) => (await Puesto.findOne({ _id: id
  }))?.nombre || 'Sin nombre';
  const getNombreUsuario = async (userId) => {
  const u = await Usuario.findOne({ usuario: userId });
  return u ? { nombre: u.nombre, apellido: u.apellido } : { nombre: '',
  apellido: '' }; };
  if (tablas.includes('Registros')) {
  const f = { fecha: { $gte: fechaInicio, $lte: fechaFin } };
  if (puestoId) f.puestoId = new ObjectId(puestoId);
  if (usuario) f.usuario = usuario;
  const registros = await Registro.find(f).toArray();
  const s = workbook.addWorksheet('INGRESOS AL PUESTO');s.columns = [
  { header: 'NOMBRE PUESTO', key: 'puesto' }, { header: 'NOMBRE', key:
  'nombre' }, { header: 'APELLIDO', key: 'apellido' },
  { header: 'FECHA', key: 'fecha' }, { header: 'HORA', key: 'hora' }, {
  header: 'UBICACION', key: 'ubicacion' }, { header: 'EVIDENCIA', key:
  'evidencia' }];
  for (const r of registros) {const u = await getNombreUsuario(r.usuario),
  p = await getNombrePuesto(r.puestoId);s.addRow({ puesto: p, nombre:
  u.nombre, apellido: u.apellido, fecha: r.fecha, hora: r.hora,ubicacion:
  r.ubicacion ? {
  text:`https://www.google.com/maps/search/?api=1&query=${r.ubicacion.trim(
  )}`,
  hyperlink:`https://www.google.com/maps/search/?api=1&query=${r.ubicacion.
  trim()}` } : 'N/A',evidencia: r.imagenUrl ? { text: r.imagenUrl,
  hyperlink: r.imagenUrl } : 'N/A' });}
  // LO MISMO PARA INCIDENTES Y MULTAS
  *****
  const buffer = await workbook.xlsx.writeBuffer();
  res.setHeader('Content-Type',
  'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
  res.setHeader('Content-Disposition', 'attachment;filename=reporte.xlsx');
  res.end(buffer);} catch (error) {console.error(error);
  res.status(500).send('Error generando el reporte');}});
```

Nota. Autoría propia.

Además, el reporte puede ser enviado al correo electrónico, tal como se observa en la **Figura 75**. Además, el sistema ofrece la opción de remitirlo a una dirección distinta a la predeterminada o compartirlo mediante la aplicación de WhatsApp, brindando mayor flexibilidad al usuario.

Figura 75
Reporte en formato xlsx.

	A	B	C	D	E	F	G
1	NOMBRE PUI	NOMBRE	APELLIDO	FECHA	HORA	UBICACION	EVIDENCIA
2	Urbanizació	Ulises	Valencia	2025-04-14	09:16:13	https://www.https://fireb	
3	Urbanizació	Ulises	Valencia	2025-04-14	11:51:40	https://www.https://fireb	
4	Urbanizació	Ulises	Valencia	2025-04-15	10:45:43	https://www.https://fireb	
5	Urbanizació	Ulises	Valencia	2025-04-15	11:11:52	https://www.https://fireb	
6	Urbanizació	Ulises	Valencia	2025-04-15	17:09:01	https://www.https://fireb	
7	Urbanizació	Ulises	Valencia	2025-04-15	17:12:42	https://www.https://fireb	
8	Urbanizació	Ulises	Valencia	2025-04-15	17:21:36	https://www.https://fireb	
9	Urbanizació	Ulises	Valencia	2025-04-15	17:27:23	https://www.https://fireb	
10	Urbanizació	Ulises	Valencia	2025-04-15	17:39:39	https://www.https://fireb	
11	Urbanizació	Ulises	Valencia	2025-04-15	17:42:58	https://www.https://fireb	
12	Urbanizació	Ulises	Valencia	2025-04-15	17:48:59	https://www.https://fireb	
13	Urbanizació	Ulises	Valencia	2025-04-15	17:56:16	https://www.https://fireb	
14	Urbanizació	Ulises	Valencia	2025-04-17	13:22:16	https://www.https://fireb	
15	Urbanizació	Ulises	Valencia	2025-04-17	13:27:56	https://www.https://fireb	
16	Urbanizació	Ulises	Valencia	2025-04-20	08:02:01	https://www.https://fireb	
17	Urbanizació	Ulises	Valencia	2025-04-20	08:04:31	https://www.https://fireb	
18	Urbanizació	Ulises	Valencia	2025-04-20	08:17:57	https://www.https://fireb	
19	Urbanizació	Ulises	Valencia	2025-04-20	08:34:22	https://www.https://fireb	
20	Urbanizació	Ulises	Valencia	2025-04-21	10:52:29	https://www.https://fireb	
21	Urbanizació	Ulises	Valencia	2025-04-21	11:01:59	https://www.https://fireb	
22	Urbanizació	Ulises	Valencia	2025-04-21	11:05:32	https://www.https://fireb	
23	Urbanizació	Ulises	Valencia	2025-04-21	11:11:24	https://www.https://fireb	
24	Urbanizació	Ulises	Valencia	2025-04-21	11:15:30	https://www.https://fireb	
25	Urbanizació	Ulises	Valencia	2025-04-21	11:20:42	https://www.https://fireb	
26	Urbanizació	Ulises	Valencia	2025-04-21	11:25:03	https://www.https://fireb	
27	Urbanizació	Ulises	Valencia	2025-04-21	11:36:14	https://www.https://fireb	
28	Urbanizació	Ulises	Valencia	2025-04-21	11:47:21	https://www.https://fireb	
29	Urbanizació	Ulises	Valencia	2025-04-21	11:52:01	https://www.https://fireb	
30	Urbanizació	Ulises	Valencia	2025-04-21	11:57:28	https://www.https://fireb	
31	Urbanizació	Ulises	Valencia	2025-04-21	12:00:10	https://www.https://fireb	
32	Urbanizació	Ulises	Valencia	2025-04-21	12:08:18	https://www.https://fireb	
33	Urbanizació	Ulises	Valencia	2025-04-21	12:11:53	https://www.https://fireb	
34	Urbanizació	Ulises	Valencia	2025-04-21	12:25:16	https://www.https://fireb	
35	Urbanizació	Ulises	Valencia	2025-04-21	12:27:53	https://www.https://fireb	
36	Urbanizació	Ulises	Valencia	2025-04-21	12:37:10	https://www.https://fireb	
37	Urbanizació	Ivan	Valencia	2025-04-21	12:47:11	https://www.https://fireb	
38	Urbanizació	Ulises	Valencia	2025-04-21	12:48:57	https://www.https://fireb	

Nota. Autoría propia.

Finalmente, es importante mencionar que los usuarios con rol de guardia únicamente tienen acceso a la pantalla de registro de incidentes, como se comprueba en la **Figura 76**.

Figura 76
Pantalla de inicio de guardias.



Nota. Autoría propia.

Tabla 14
Retrospectiva del Sprint 1

Reunión de retrospectiva				
Sprint: 1				
Fecha: 27/04/2025				
Actividad / Historia de usuario	Aspectos positivos	Aspectos negativos	Lecciones aprendidas	Acciones propuestas
HU-03 Gestión del puesto de guardia.	Se logró implementar de forma efectiva el registro de turnos con evidencia fotográfica y geolocalización.	La configuración de la lógica de tiempo de llegada y la integración del método Haversine tomó más tiempo del previsto.	La validación geográfica requiere pruebas anticipadas en distintos dispositivos móviles y entornos reales.	Optimizar la tolerancia del sistema ante variaciones leves de ubicación.
HU-04 Gestión de los puestos a supervisar.	Se completó exitosamente el módulo de supervisión con funciones como registro de incidentes, creación de multas y generación de reportes automáticos.	La interfaz para visualización de evidencias y alertas resultó más compleja de integrar en dispositivos móviles.	Es necesario diseñar desde el inicio un flujo claro para usuarios con múltiples privilegios y pantallas activas.	Dividir módulos complejos en subcomponentes reutilizables.

Nota. Autoría propia.

CAPÍTULO 3

Validación de resultados

3.1 Metodología de Evaluación

Se procedió a hacer la validación del sistema, mediante la subcaracterística de la disponibilidad definida en la ISO/IEC 25023, la cual se enfoca en evaluar la capacidad del sistema para ejecutarse sin interrupciones en los momentos en los que se requiera de su uso (Perleche et al., 2022).

Para cumplir con el objetivo planteado se tomó en cuenta dos enfoques los cuales permiten comprobar la disponibilidad del sistema: una evaluación cuantitativa mediante bitácora manual, además de una evaluación de tolerancia y estabilidad mediante pruebas de estrés.

3.1.1 Evaluación mediante bitácora manual.

Se optó por una estrategia de registro continuo del comportamiento del sistema durante un periodo específico, simulando así el uso normal del aplicativo, en distintos horarios y fechas, probando cada funcionalidad del aplicativo y al registrar errores e incidentes poder sacar datos cuantitativos para su posterior análisis y a su vez calcular las métricas propuestas, se solicitaron las métricas ilustradas en la **Figura 77**.

Figura 77

Datos requeridos en la bitácora.

Datos requeridos	Explicación
Sesión.	Numero de sesión en la que se realizó la prueba.
Fecha.	Fecha en la que se realizó la prueba formato DD/MM/YYYY.
Hora de Inicio.	Hora en que comenzó la prueba formato 24h.

Hora de Fin.	Hora en que termino la prueba formato 24h.
¿Hubo fallo?	Si, si existió algún tipo de fallo; No, si no existió ningún fallo.
Hora de fallo.	Si hubo fallo se pone la hora, si no se deja en blanco.
Hora de recuperación.	Si hubo fallo se pone la hora, si no se deja en blanco.
Duración del fallo.	La duración en minutos del fallo ocurrido en la sesión.
Tiempo total de la prueba.	La duración total de la sesión en minutos.
Observaciones.	Cualquier observación o tipo de error obtenido durante la sesión.

Nota. Autoría propia.

Como se visualiza en la **Figura 78**, se obtienen todos los datos para su posterior utilización con el fin de calcular las métricas que aportaran a la validación del aplicativo.

Finalmente con los datos obtenidos en la primera hoja de Excel, se crea una segunda la cual permitirá obtener las métricas requeridas en base a los datos como se muestra en la **Figura 79**.

Figura 78

Visualización de los datos de la bitácora manual.

Sesión	Fecha	Hora de Inicio	Hora de Fin	¿Hubo fallo? (Sí/No)	Hora de fallo	Hora de recuperación	Duración del fallo (min)	Tiempo total de prueba (min)	Observaciones
1	10/5/2025	8:00	9:00	Sí	10:15	10:20	5	60	Fallo al subir evidencia con imagen en multas
2	11/5/2025	15:00	16:00	No			0	60	Todo sin novedad
3	12/5/2025	8:28	9:02	No	8:43	8:50	7	34	Fallo al crear datos, se crea con fecha 13/05/2025
4	12/5/2025	17:31	18:22	No			0	51	Todo sin novedad
5	13/5/2025	6:30	6:47	Sí	6:37	6:38	1	17	Reporte se creo vacio
6	13/5/2025	16:03	16:55	No			0	52	Todo sin novedad
7	14/5/2025	10:11	10:38	No			0	27	Todo sin novedad
8	14/5/2025	20:11	21:00	No			0	49	Todo sin novedad
9	15/5/2025	5:27	5:58	No			0	31	Todo sin novedad
10	15/5/2025	14:29	15:01	Sí	14:30	14:32	2	32	No inicio sesión, Error en el servidor
11	16/5/2025	8:53	9:34	No			0	41	Todo sin novedad
12	16/5/2025	20:43	21:12	No			0	29	Todo sin novedad
13	17/5/2025	13:06	13:39	No			0	33	Todo sin novedad
14	17/5/2025	17:49	18:37	No			0	48	Todo sin novedad
15	18/5/2025	7:38	8:01	No			0	23	Todo sin novedad
16	18/5/2025	14:01	14:57	Sí	14:36	14:40	4	56	Se quedo cargando en la pantalla de registro
17	19/5/2025	9:31	12:01	Sí	9:38	9:48	10	150	Problemas con el servidor, Error en el servidor
18	19/5/2025	14:31	17:01	Sí	15:02	15:04	2	150	Metodo /Uploud-data marca , Error en el servidor
19	20/5/2025	11:33	11:52	Sí	11:39	11:48	9	19	Error al crear el reporte, con mas de 300 campos, no genera
20	20/5/2025	18:35	19:01	Sí	18:36	18:47	11	26	Error al crear el reporte, con mas de 300 campos, no genera
21	21/5/2025	7:27	7:58	No			0	31	Todo sin novedad
22	21/5/2025	19:26	20:13	No			0	47	Todo sin novedad
23	22/5/2025	10:03	10:33	No			0	30	Todo sin novedad
24	22/5/2025	18:05	19:13	Sí	18:16	18:19	3	68	Error en el servidor, al ingresar
25	23/5/2025	4:37	5:08	No			0	31	Todo sin novedad
26	23/5/2025	13:49	14:12	No			0	23	Todo sin novedad
27	24/5/2025	11:12	11:42	No			0	30	Todo sin novedad
28	24/5/2025	19:53	20:22	No			0	30	Todo sin novedad
29	25/5/2025	7:01	7:22	Sí	7:01	7:04	3	21	No cargaron los puestos en incidentes
30	25/5/2025	12:06	12:35	No			0	29	Todo sin novedad
31	26/5/2025	9:32	9:52	Sí	9:42	9:50	8	20	Reporte se creo vacio
32	26/5/2025	15:17	15:49	No			0	32	Todo sin novedad
33	27/5/2025	6:57	7:30	No			0	33	Todo sin novedad
34	27/5/2025	15:23	15:49	No			0	26	Todo sin novedad
35	28/5/2025	8:16	9:01	No			0	45	Todo sin novedad
36	28/5/2025	17:07	17:34	No			0	27	Todo sin novedad
37	29/5/2025	9:11	9:37	No			0	26	Todo sin novedad
38	29/5/2025	16:48	17:23	No			0	35	Todo sin novedad

Nota. Autoría propia.

Figura 79

Resultados obtenidos de las métricas.

Métrica	Valor
Total de sesiones.	38
Total de fallos.	11
Tiempo total de prueba(min).	1572
Tiempo total de inactividad(min).	65
MTBF(min)	142,9090909
MTTR(min)	5,909090909
Disponibilidad(%)	96,02932193

Nota. Autoría propia.

Se utilizo varias fórmulas para obtener los resultados de la **Figura 79** como la fórmula del MTBF y el MTTR, además del porcentaje de Disponibilidad. A continuación, se detalla la fórmula utilizada para obtener el tiempo medio entre fallos(MTBF):

$$MTBF = \frac{T}{N}$$

Donde

T es el Tiempo total de prueba

N es el Total de fallos

Por lo que para este cálculo se asignaron los siguientes valores a las variables:

$$T = 1572$$

$$N = 11$$

Reemplazando en la fórmula y resolviendo el cálculo:

$$MTBF = \frac{1572}{11} = \mathbf{142,9090909}$$

Por lo que ya tendríamos el resultado del MTBF el cual nos ayudara a calcular posteriormente el % de Disponibilidad. A continuación , se detalla la otra métrica necesaria el tiempo medio de reparación (MTTR).

$$MTTR = \frac{T_R}{N}$$

Donde

T_R es el Tiempo total de inactividad

N es el Total de fallos

Por lo que para este cálculo se asignaron los siguientes valores a las variables:

$$T_R = \sum \text{Duración del fallo} = 65$$

$$N = 11$$

Reemplazando en la fórmula y resolviendo el cálculo:

$$MTTR = \frac{65}{11} = \mathbf{5,909090909}$$

Con esto ya se tiene los valores del MTBF Y MTTR con esto ya se tiene lo necesario para sacar el % de disponibilidad, como se muestra a continuación:

$$\% \text{ Disponibilidad} = \frac{MTBF}{(MTBF + MTTR)} \times 100$$

Para este cálculo se asignaron los siguientes valores a las variables:

$$MTBF = 142,9090909$$

$$MTTR = 5,909090909$$

Reemplazando en la fórmula y resolviendo el cálculo:

$$\% \text{ Disponibilidad} = \frac{142,9090909}{(142,9090909 + 5,909090909)} \times 100$$

$$\% \text{ Disponibilidad} = \mathbf{96,02932193}$$

Finalmente, ya se tiene todas las métricas necesarias para poder interpretar si el aplicativo satisface la disponibilidad de la ISO 25023.

3.1.2 Evaluación mediante pruebas de estrés.

Se llevaron a cabo pruebas de estrés utilizando la herramienta de Postman, que ayuda a probar los principales endpoints del aplicativo, mediante el apartado de runner permite a su vez ejercer solicitudes intensas.

Estos runners permiten que se puedan probar varias iteraciones a un endpoint y con un delay parametrizado, esto permite realizar las pruebas de estrés además de pruebas de concurrencia simulando varios usuarios, con varias sesiones de runners abiertas.

Se evaluaron los endpoints utilizados en el aplicativo móvil, los cuales serán los más utilizados en la empresa y necesitan esta evaluación, llegando a los endpoint visualizados en la **Figura 80**.

Figura 80

Endpoints a evaluar en las pruebas de estrés.

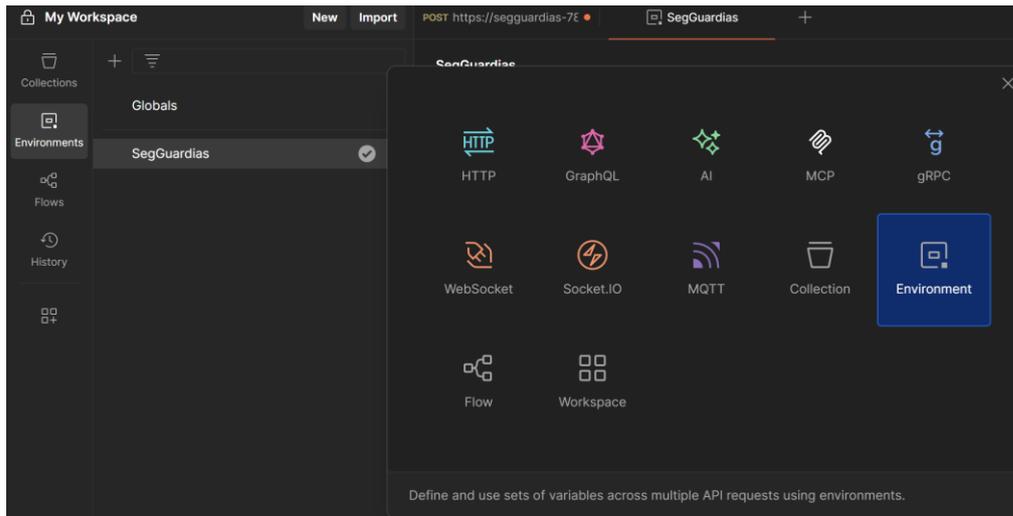
Endpoint	Descripción breve	Solicitudes
/login	Inicio de sesión, clave para autenticación.	100
/upload-data	Subida de registros (imagen + datos).	30
/report-incident	Registro de incidentes.	30
/report-multa	Registro de multas.	30
/alertas/crear	Generación automática/manual de alertas.	30
/alertas/:usuario	Alertas activas según el guardia.	100
/multas-formato-tabla	Consulta de multas para reporte/vista.	100
/incidentes-formato-tabla	Consulta de incidentes en formato tabla.	100
/registros-formato-tabla	Consulta de ingresos en tabla.	100
/puestos-asignados/:usuario	Ver puestos asignados al usuario.	100
/puesto/obtener-id/:ubicacion	Para identificar el puesto desde ubicación GPS.	100

Nota. Autoría propia.

Como se visualiza en la **Figura 80**, en la mayoría de las solicitudes se tomó como 100 iteraciones como un predeterminado, y la mayoría de estas a su vez son solicitudes GET, a excepción del endpoint /login el cual es tipo POST, debido a que esta no tiene gran carga.

En cambio, las solicitudes con 30 iteraciones son todas de tipo POST y se tomó solo 30, debido a que estas a su vez realizan otros subprocesos de mayor carga, como subir una imagen o crear y enviar un .pdf mediante correo, teniendo todo esto en cuenta se procedió con la configuración para realizar las pruebas usando Postman.

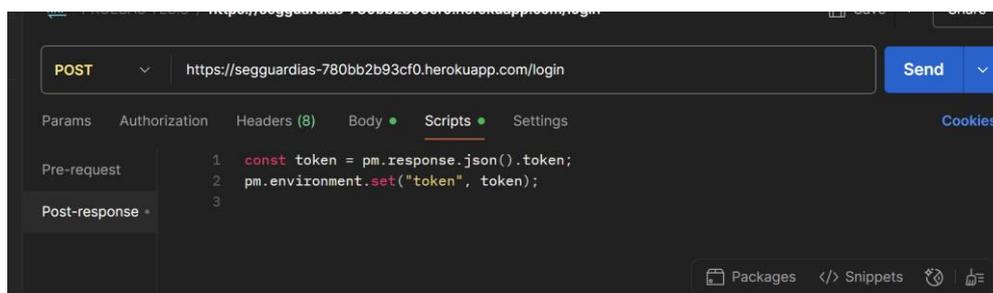
Figura 81
Creación del Environment SegGuardias en Postman.



Nota. Autoría propia.

Primero, se realizó la creación del environment como se ilustra en la **Figura 81**, que servirá como configuración inicial y guardará el token que permitirá, realizar todas las iteraciones solicitadas, para eso se debe agregar un pequeño script, el cual guardará el token en el environment al realizar una solicitud de login como se visualiza en la **Figura 82**.

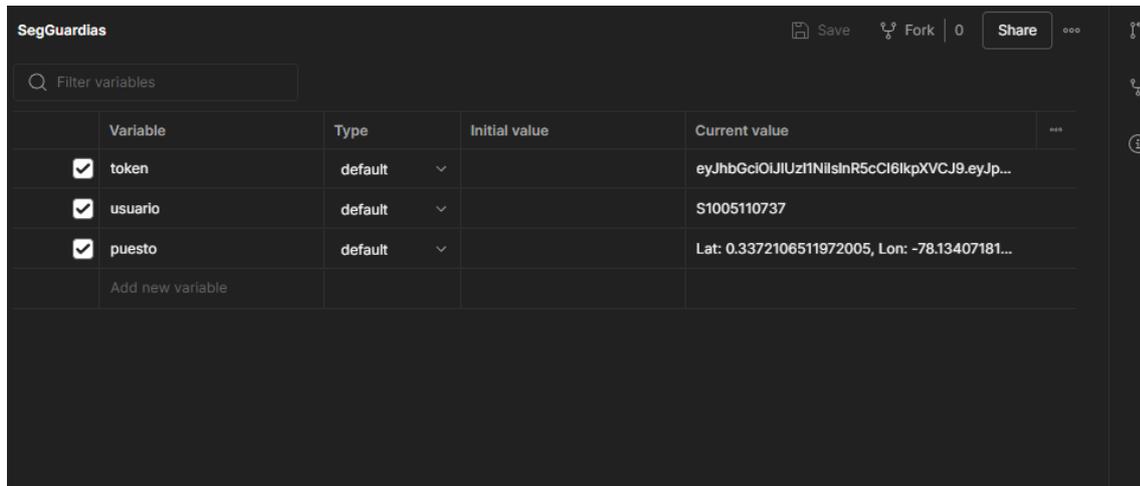
Figura 82
Creación del Script de token.



Nota. Autoría propia.

Al ejecutar normalmente la solicitud de login se guardará en el environment, a su vez se puede agregar más variables para utilizarlas en los endpoint, resultando este archivo con los siguientes datos ilustrados en la **Figura 83**.

Figura 83
Environment SegGuardias.

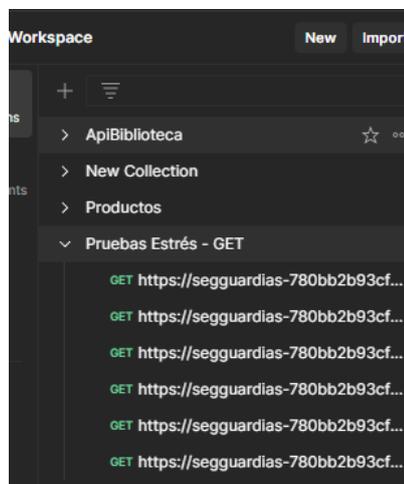


Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/> token	default		eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJp...
<input checked="" type="checkbox"/> usuario	default		S1005110737
<input checked="" type="checkbox"/> puesto	default		Lat: 0.3372106511972005, Lon: -78.13407181...
Add new variable			

Nota. Autoría propia.

Posteriormente, se crea las colecciones con los endpoint a probar, en este caso se dividió en dos colecciones por tipo de solicitud y una para solamente el endpoint login, para el primer aparato se creó la colección Pruebas Estrés – GET, como se visualiza en la **Figura 84**.

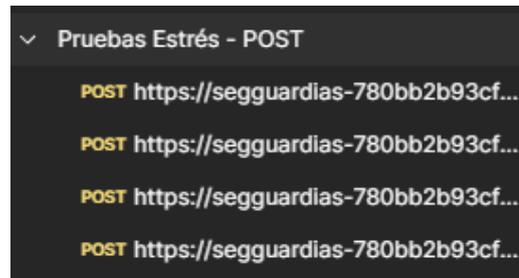
Figura 84
Colección Pruebas Estrés – GET.



Nota. Autoría propia.

A continuación se ilustra la colección en la **Figura 85** para el tipo de solicitud POST, llamado Pruebas Estrés - POST, que contiene los endpoint tipo POST, a excepción del login como se mencionó anteriormente.

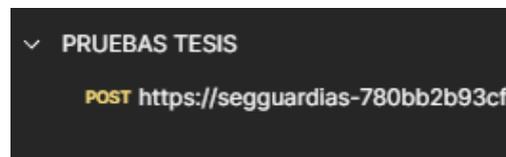
Figura 85
Colección Pruebas Estrés – POST.



Nota. Autoría propia.

Finalmente, se ilustra la colección para la solicitud POST, llamada PRUEBAS TESIS la cual solo contiene el endpoint login, como se ilustra en la **Figura 86**.

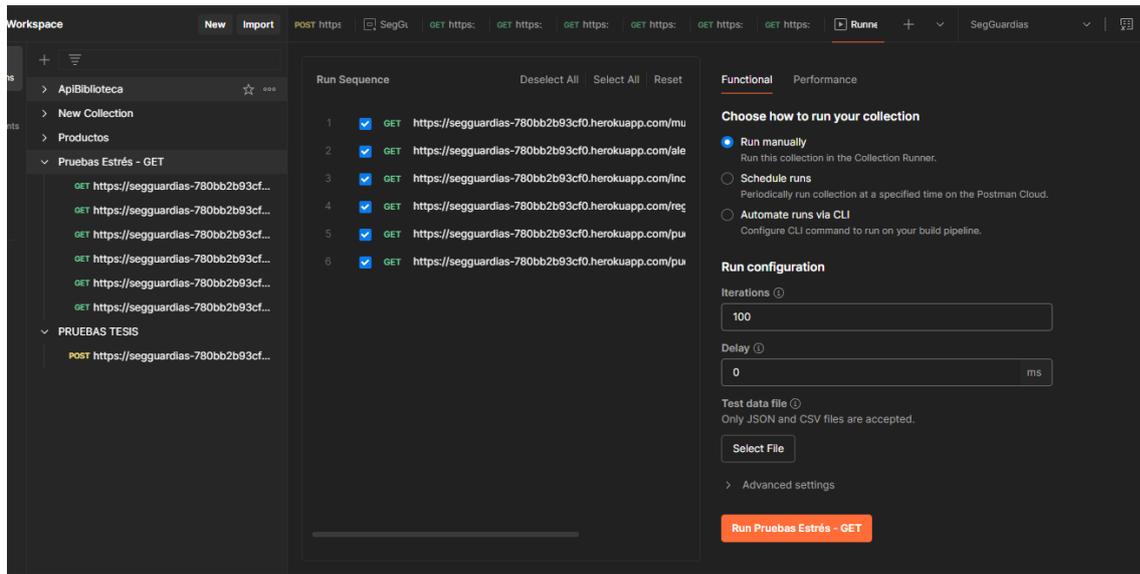
Figura 86
Colección PRUEBAS TESIS.



Nota. Autoría propia.

Ya con la configuración y entorno requerido, creamos el Runner que nos ayudara a realizar la prueba de estrés en este caso para los de tipo GET, como se ilustra en la **Figura 87**.

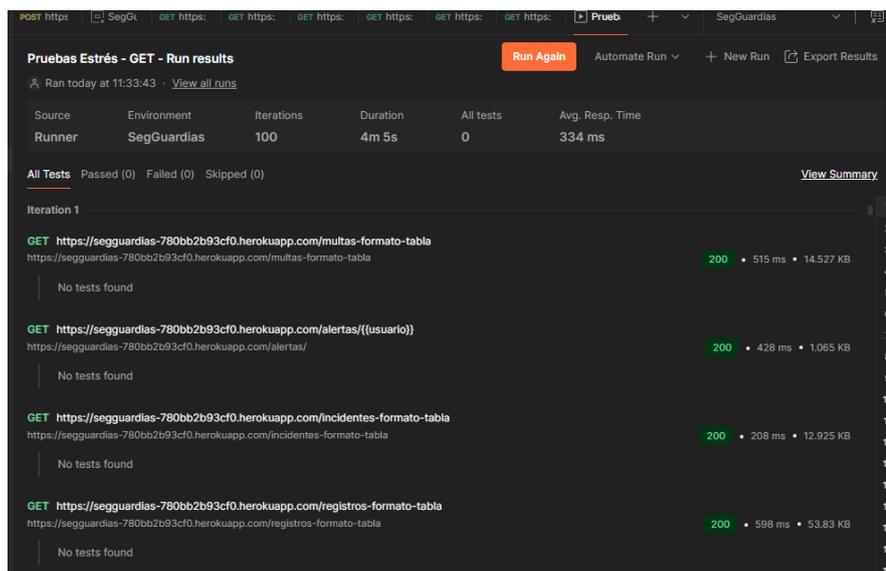
Figura 87
Configuración de Runner para la colección de tipo GET.



Nota. Autoría propia.

Se hizo una prueba intensa enviando 100 solicitudes seguidas a los principales endpoints tipo GET del sistema, sin ningún tipo de espera entre una y otra (0 ms de delay) como se mostró en la **Figura 87**. Todos respondieron correctamente, sin errores ni caídas, y el tiempo promedio de respuesta fue de 334 milisegundos como se ilustra en la **Figura 88**.

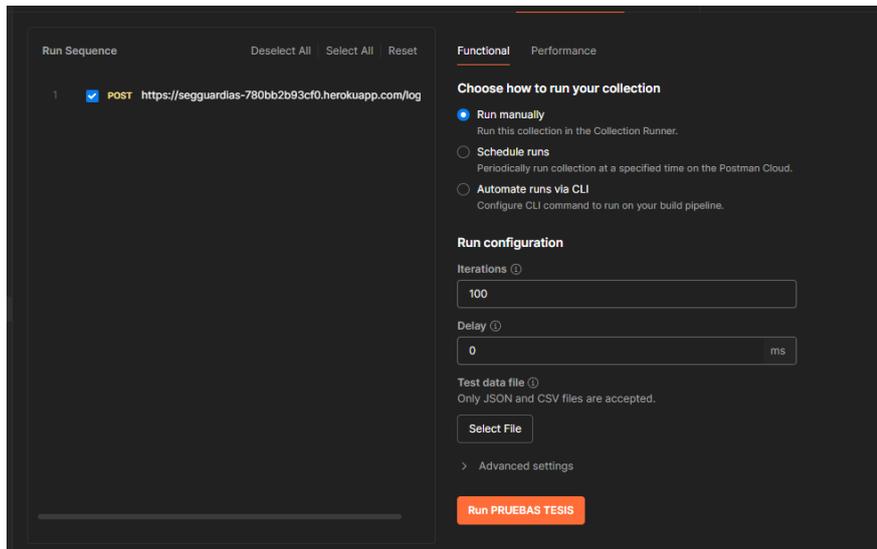
Figura 88
Resultado obtenido del Runner para la colección tipo GET.



Nota. Autoría propia.

Posteriormente se realizó el mismo proceso pero para el endpoint login con la misma configuración utilizada en la colección de tipo GET, como se ilustra en la **Figura 89**.

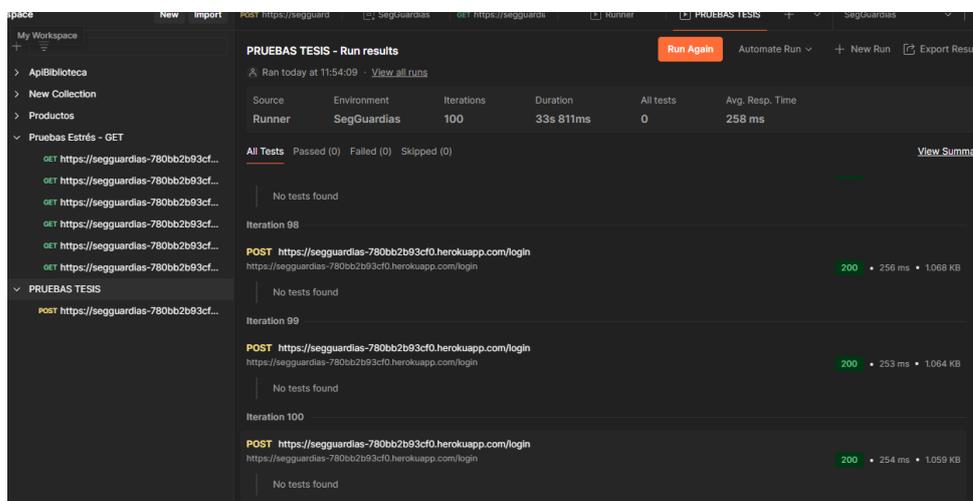
Figura 89
Configuración de Runner para la colección de tipo POST Login.



Nota. Autoría propia.

En este caso, todas las peticiones fueron exitosas. El sistema no presentó errores ni bloqueos, y el tiempo promedio de respuesta fue de 258 milisegundos, como se visualiza en la **Figura 90**.

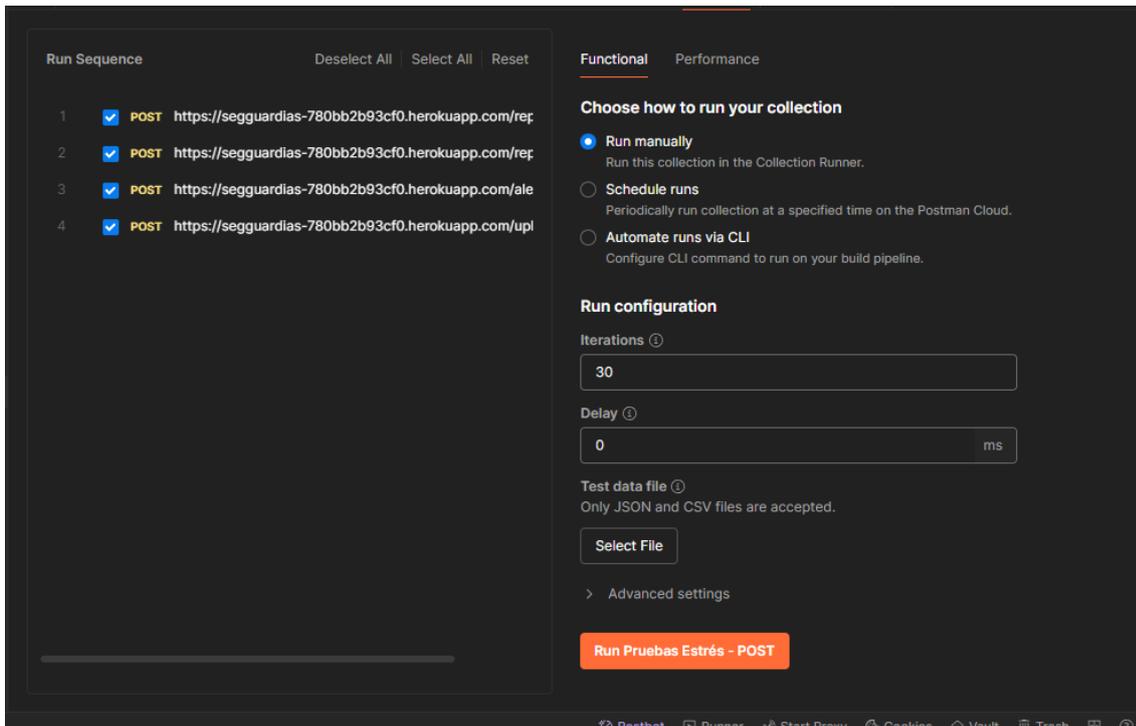
Figura 90
Resultado obtenido del Runner para la colección tipo POST Login.



Nota. Autoría propia.

Finalmente, se realizó la configuración y evaluación para la colección de tipo POST, en este caso como se concluyó anteriormente con 30 interacciones, como se visualiza en la **Figura 91**.

Figura 91
Configuración de Runner para la colección de tipo POST.

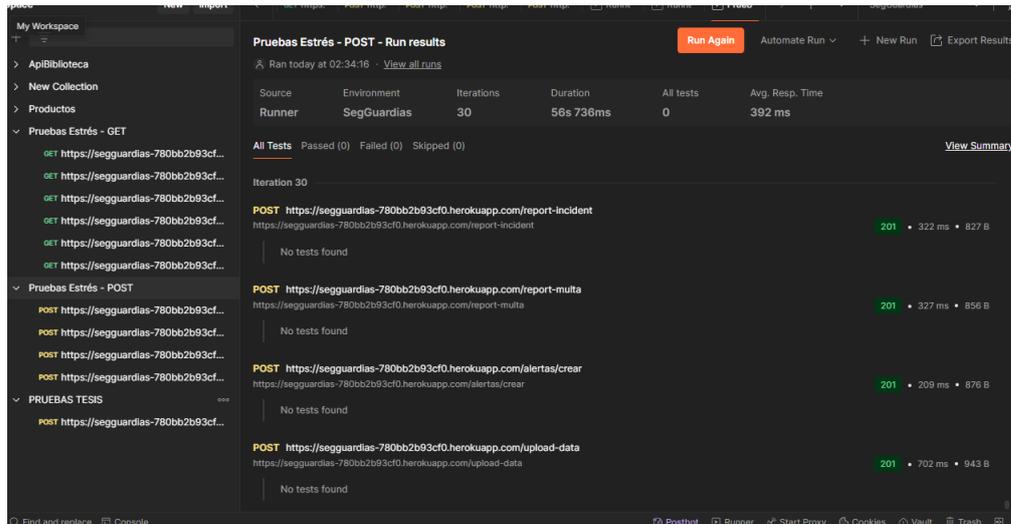


Nota. Autoría propia.

Así como en las pruebas anteriores, se ejecutó esta colección con un delay de 0 ms, pero en este caso con solo 30 iteraciones, ya que estos endpoints implican la subida de imágenes, lo cual agrega mayor peso a la carga. Aun así, el tiempo promedio de respuesta fue de 392 ms, lo cual es un valor aceptable, esto se comprueba en la **Figura 92**.

Figura 92

Resultado obtenido del Runner para la colección tipo POST.

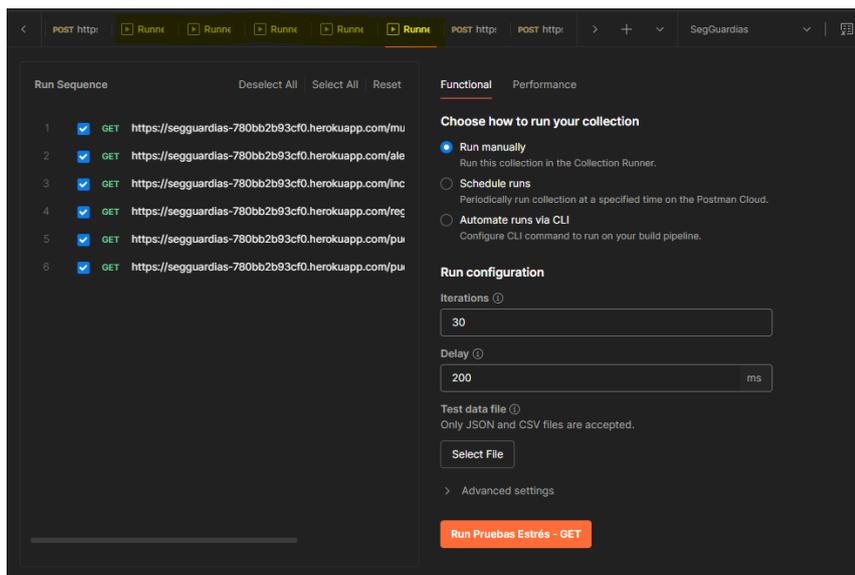


Nota. Autoría propia.

Por otro lado, se simularon múltiples usuarios concurrentes mediante el uso de instancias paralelas de Postman Runner, configurando delay y volumen de iteraciones para replicar el comportamiento de múltiples usuarios activos en la aplicación móvil, en este caso con 5 instancias de Runner, con cada instancia con 30 iteraciones y un delay de 200 ms que simula un uso más realista, como se ilustra en la .

Figura 93

Pruebas de concurrencia con cinco usuarios en endpoints GET.



Nota. Autoría propia.

Todas las respuestas fueron exitosas, sin errores ni demoras notables. El tiempo de respuesta se mantuvo entre 392 y 397 milisegundos, como se ilustra en la **Figura 94**.

Figura 94
Resultados de las iteraciones del tipo GET.

N° de sesión	Tiempo de respuesta
1	397 ms
2	395 ms
3	392 ms
4	395 ms
5	397 ms

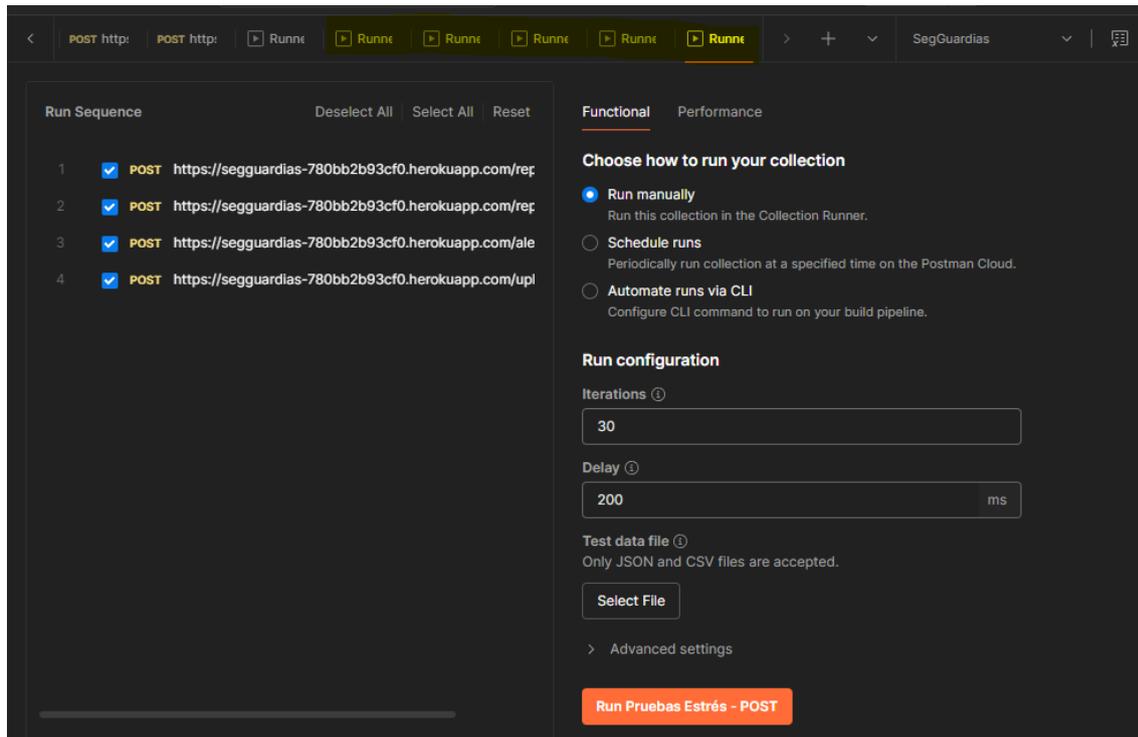
Nota. Autoría propia.

Asimismo, se realizó lo mismo pero con la colección de tipo POST usando el mismo número de instancias, iteraciones y delay, como se visualiza en la **Figura 95**. A diferencia de los endpoints GET, esta colección incluye métodos que suben imágenes (/upload-data, /report-multa, /report-incident), lo cual representa una mayor carga tanto en datos como en procesamiento.

Durante las pruebas se presentaron errores en el endpoint /upload-data, sobre todo al inicio de cada ejecución. En cada una hubo entre 4 y 9 errores, pero solo al comienzo; el resto de las peticiones funcionaron sin problemas, como se ilustra en la **Figura 96**.

Figura 95

Pruebas de concurrencia con cinco usuarios en endpoints POST.



Nota. Autoría propia.

Figura 96

Resultados de las iteraciones del tipo POST.

Nº de sesión	Tiempo de respuesta	Numero de errores
1	394 ms	9
2	400 ms	4
3	391 ms	8
4	390 ms	6
5	377 ms	9

Nota. Autoría propia.

Esto muestra que cuando varias solicitudes con imágenes llegan al mismo tiempo, el sistema puede fallar un poco al principio, tal vez por sobrecarga del servidor o de Firebase. Aun así, el sistema se recupera rápido y sigue respondiendo bien, con tiempos de respuesta entre 377 y 400 milisegundos

Para demostrar que paso las pruebas de estrés podemos calcular la tolerancia a fallos utilizando las pruebas de concurrencia realizadas. Aunque en literatura técnica se utiliza con frecuencia la métrica tasa de error para representar los fallos ocurridos en este tipo de pruebas, en este proyecto se decidió tomar en cuenta el porcentaje de solicitudes que fueron procesadas exitosamente, o denominado tolerancia al fallo (Sachin, 2023). A continuación, se detalla la fórmula utilizada para obtener el porcentaje de tolerancia al fallo del aplicativo:

$$\text{Tolerancia al fallo (\%)} = \left(\frac{T - E}{T} \right) \times 100$$

Donde

T es el Total de solicitudes

E es el Numero de errores

Por lo que para este cálculo se asignaron los siguientes valores a las variables:

$$T = 300$$

$$E = 36$$

Reemplazando en la fórmula y resolviendo el cálculo:

$$\text{Tolerancia al fallo (\%)} = \left(\frac{300 - 36}{300} \right) \times 100$$

$$\text{Tolerancia al fallo (\%)} = \mathbf{88}$$

Este resultado indica que el sistema fue capaz de manejar correctamente el 88 % de las solicitudes bajo condiciones exigentes.

3.2 Análisis e interpretación de resultados.

Ya con los resultados se puede comprobar que el sistema alcanza una disponibilidad del 96,03 %, un MTBF de 142,9 minutos y un MTTR de 5,9 minutos, como se ilustra en la **Figura 97**, métricas que se encuentran dentro de los parámetros aceptables establecidos por la norma ISO/IEC 25023, la cual considera que una disponibilidad superior al 85 % es adecuada para sistemas no críticos (International Organization for Standardization, 2016).

Figura 97

Resultados de las métricas obtenidos.

Métrica	Resultado obtenido
MTBF(min)	142,9 minutos
MTTR(min)	5,9 minutos
Disponibilidad(%)	96,03%
Tolerancia al fallo (%)	88%

Nota. Autoría propia.

Además, los resultados obtenidos en las pruebas de estrés reflejan la capacidad del sistema para mantener un rendimiento estable bajo condiciones de alta concurrencia, lo que es fundamental para aplicaciones que operan en entornos reales con múltiples usuarios simultáneos. La tolerancia al fallo del 88 %, mostrada en la **Figura 97** y calculada a partir de las pruebas de concurrencia realizadas, demuestra que el sistema fue capaz de mantener su operatividad ante múltiples solicitudes simultáneas. Esta métrica, basada en la proporción de solicitudes exitosas frente al total enviado, es comúnmente utilizada en pruebas de rendimiento para evaluar la estabilidad de aplicaciones web y móviles (Sachin, 2023).

Por tanto, el sistema superó satisfactoriamente la validación de la subcaracterística de disponibilidad, y cumple con el objetivo planteado en este proyecto, demostrando su preparación técnica para operar de forma confiable en entornos reales.

CONCLUSIONES

El desarrollo de la aplicación móvil junto con su administrador web, utilizando tecnologías como React Native con Expo, Node.js, MongoDB y FireStorage, permitió construir una solución robusta y escalable para la gestión del personal de seguridad en la empresa Alpriseg Cía Ltda, que mediante herramientas como expo-location permite conseguir información relevante del personal, que servirá como evidencia para sustentar las bitácoras del puesto de trabajo. Además, la adopción de la metodología ágil Scrum facilitó el control y la adaptación constante durante el desarrollo, asegurando la entrega oportuna de funcionalidades clave.

La validación técnica realizada mediante la subcaracterística de disponibilidad definida por la norma ISO/IEC 25023, apoyada en bitácoras manuales y pruebas de estrés con Postman, evidenció que el sistema mantiene una alta disponibilidad (96,03 %) y una tolerancia al fallo adecuada (88 %) bajo condiciones de carga simultánea. Estos resultados confirman que la solución propuesta cumple con los requisitos de funcionamiento en ambientes operativos reales, apoyando a que pueda ser escalable y mejorable.

La implementación del sistema contribuirá significativamente a mejorar el control y la supervisión del personal de seguridad, facilitando la generación de reportes confiables y la gestión eficiente de incidencias en tiempo real. Esto permitirá un apoyo a las bitácoras por puesto, teniendo evidencia fotográfica, geográfica y horarias, la cual a su vez da apoyo al responder a incidentes los cuales requieran una rendición de cuentas ante entidades contratantes o de control. Este aporte tecnológico representa un avance relevante para la empresa Alpriseg Cía Ltda, la cual beneficiara a sus objetivos de brindar un servicio de calidad.

RECOMENDACIONES

Desarrollar un modo offline para la aplicación móvil que permita al usuario almacenar los registros, capturas de evidencia y demás procesos sin necesidad de una conexión constante a internet. Esto garantizaría la continuidad operativa en zonas con poca o nula cobertura, sincronizando los datos automáticamente cuando se restablezca la conexión, mejorando así la funcionalidad del sistema en entornos más rurales, adaptándose a futuros puestos que requieran las funciones del aplicativo.

Ampliar el módulo de puestos para incluir parámetros configurables, tales como los minutos de cortesía permitidos y el perímetro geográfico asignado a cada puesto en base al contrato de servicio correspondiente, el cual se agregaría como un adjunto al crear el puesto. Esto facilitaría que los administradores cuenten con información precisa para la gestión y cumplimiento de las condiciones establecidas, optimizando el control y la planificación de los recursos.

Implementar un nuevo apartado en la plataforma móvil que permita visualizar en tiempo real con un pequeño delay, la ubicación de cada guardia en un mapa interactivo. Esta funcionalidad mejoraría la supervisión y seguridad del personal, facilitando la toma de decisiones inmediatas ante incidentes o situaciones de emergencia, y fortaleciendo el seguimiento continuo de las actividades operativas.

BIBLIOGRAFÍA

- Acosta Flores, M. J. (2019). *Sistema de control y supervisión de los servicios de seguridad mediante geolocalización a través de tecnologías web – móvil para Delta Seguridad Cía. Ltda.*
<https://repositorio.puce.edu.ec/handle/123456789/39690>
- Amin Kuhail, M., & Lauesen, S. (2022). *User Story Quality in Practice: A Case Study*. 1, 223–243. <https://doi.org/10.3390/software1030010>
- Appinventiv. (2024). *A Definitive Guide on Agile Mobile App Development*. <https://appinventiv.com/blog/agile-scrum-methodology-in-mobile-app-development/>

- Arévalo Chávez, P., Cárdenas, J. C., Maldonado, G., Fierro, P., Bonilla Bedoya, S., Bastidas, A. E., Lanas, J. G., Zapata Rodríguez, M., Guerrero, J. J., Flores, H. A., Ramos, C., Patricio, G., Chávez, A., & Ramos Galarza, C. (2020). *Actualización en metodología de la investigación científica*. 13–159. <https://repositorio.uti.edu.ec/handle/123456789/1686>
- Autili, M., Malavolta, I., Perucci, A., Scoccia, G. L., & Verdecchia, R. (2021). Software engineering techniques for statically analyzing mobile apps: research trends, characteristics, and potential for industrial adoption. *Journal of Internet Services and Applications*, 12(1), 1–60. <https://doi.org/10.1186/S13174-021-00134-X/TABLES/4>
- Berihun, N. G., Dongmo, C., & Van der Poll, J. A. (2023). The Applicability of Automated Testing Frameworks for Mobile Application Testing: A Systematic Literature Review. *Computers* 2023, Vol. 12, Page 97, 12(5), 97. <https://doi.org/10.3390/COMPUTERS12050097>
- Bhimanapati, V. B., Jain, S., & Aggarwal, A. (2024). Agile Methodologies in Mobile App Development for Real-Time Data Processing. *Universal Research Reports*, 11(4), 211–231. <https://doi.org/10.36676/URR.V11.I4.1350>
- BlazeMeter. (2024). *BlazeMeter* | *Documentation*. <https://help.blazemeter.com/docs/guide/intro.html>
- BORRMART. (2024). *Ley de Vigilancia y Seguridad Privada entra en vigor en Ecuador*. https://www.segurilatam.com/actualidad/ley-de-vigilancia-y-seguridad-privada-ecuador_20240214.html
- Briones, M. (2024). *SICOSEP | Consulta De Guardias De Seguridad Ecuador - certificados.ec*. <https://certificados.ec/sicosep-consulta-de-guardias-de-seguridad-ecuador/>
- Caiza Chacón, G. A. (2019). *Métricas para la calidad interna, externa y en uso ISO/IEC 25022 e*. <https://1library.co/article/m%C3%A9tricas-calidad-interna-externa-uso-iso-iec.y4w4wlvq>
- Calabrese, J., & Muñoz, R. (2018). *ASISTENTE PARA LA EVALUACIÓN DE CALIDAD DE PRODUCTO DE SOFTWARE SEGÚN LA FAMILIA DE NORMAS ISO/IEC 25000 UTILIZANDO EL ENFOQUE GQM*.
- Cando Villa, G. A. (2020). *Estrategias financieras para incrementar la rentabilidad en la empresa Vigelectrom Cia. Ltda en la ciudad de Ambato provincia de Tungurahua*. <http://dspace.esPOCH.edu.ec/handle/123456789/14852>
- Carcelén Gutiérrez, G. E. (2022). *Diseño e implementación de un prototipo que permita desplazamientos autónomos a una cámara de vigilancia fija con servomotores y sensores de movimientos, controlados mediante una aplicación móvil*. <http://repositorio.ug.edu.ec/handle/redug/64029>
- Cartaxo, B., Araújo, A., Barreto, A. S., & Soares, S. (2013). The impact of scrum on customer satisfaction: An empirical study. *Proceedings - 2013 27th Brazilian Symposium on Software Engineering, SBES 2013*, 129–136. <https://doi.org/10.1109/SBES.2013.10>

- Chandarana, N., Gada, T., & Kukkadapu, S. (2024). A Comprehensive Review of Agile Methodologies in Mobile App Development. *International Research Journal of Modernization in Engineering Technology and Science*. <https://doi.org/10.56726/IRJMETS61006>
- Chaudhary, M. (2023). *Test Your API's Performance by Simulating Real-World Traffic with Postman* | Postman Blog. <https://blog.postman.com/postman-api-performance-testing/>
- Chauhan, A. (2019). A Review on Various Aspects of MongoDB Databases. *International Journal of Engineering Research & Technology*, 8(5). <https://doi.org/10.17577/IJERTV8IS050031>
- Contreras Mejia, N. Andrea. (2019). *Metodología para la ejecución de pruebas no funcionales (carga y estrés) de software para KACTUS-HCM en Digital Ware*. <http://repositoriodspace.unipamplona.edu.co/jspui/handle/20.500.12744/658>
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *Una introducción básica a la teoría y práctica de Scrum Versión 2.0*. www.goodagile.com
- DigitalOcean. (2010). *Scrum Team Survey*. <https://scrumteamsurvey.org/>
- Drewbatgit. (2022). *React Native para el desarrollo de aplicaciones de escritorio de Windows* | Microsoft Learn. <https://learn.microsoft.com/es-es/windows/dev-environment/javascript/react-native-for-windows>
- Expo Documentation. (2024). *Location - Expo Documentation*. <https://docs.expo.dev/versions/latest/sdk/location/>
- Fernanda Araujo, M. (2023). *Nueva Ley de Vigilancia y Seguridad Privada: Hacia un Marco Legal Integral y Actualizado*. <https://www.asambleanacional.gob.ec/es/blogs/maria-fernanda-araujo/90628-nueva-ley-de-vigilancia-y-seguridad>
- Goševa-Popstojanova, K., & Trivedi, K. S. (2001). Architecture-based approach to reliability assessment of software systems. *Performance Evaluation*, 45(2–3), 179–204. [https://doi.org/10.1016/S0166-5316\(01\)00034-7](https://doi.org/10.1016/S0166-5316(01)00034-7)
- Hema, V., Thota, S., Naresh Kumar, S., Padmaja, C., Rama Krishna, C. B., & Mahender, K. (2020). Scrum: An Effective Software Development Agile Tool. *IOP Conference Series: Materials Science and Engineering*, 981(2). <https://doi.org/10.1088/1757-899X/981/2/022060>
- Herrera, S. I., Najar, P. J., Palavecino, R., & Goñi, J. L. (2013). *Evaluación de la calidad en aplicaciones móviles*.
- Huerta, O., Ayala, V., Murphy, L., & Portillo, A. O. (2019). DYNAMOJM: A JMeter Tool for Performance Testing Using Dynamic Workload Adaptation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11812 LNCS, 234–241. https://doi.org/10.1007/978-3-030-31280-0_14/FIGURES/6
- HyScaler. (2024). *How to Load Test Your Web App with Locust.io: A Step-by-Step Guide - HyScaler*. <https://hyscaler.com/resources/load-test-web-app-with-locust/>

- IdeaUsher. (2024). *Agile Scrum Methodology In Mobile App Development - IdeaUsher*. <https://ideausher.com/blog/agile-app-development/>
- International Organization for Standardization. (2016). *ISO/IEC 25023:2016(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality*. <https://www.iso.org/obp/ui/es/#iso:std:iso-iec:25023:ed-1:v1:en>
- Iwama, F., & Fukuda, T. (2019). Automated testing of basic recognition capability for speech recognition systems. *Proceedings - 2019 IEEE 12th International Conference on Software Testing, Verification and Validation, ICST 2019*, 13–24. <https://doi.org/10.1109/ICST.2019.00012>
- Japara, E. M., & Arifin, S. (2023). Android application development using flutter framework: Creation of geolocation system module to validate user location coordinates. *AIP Conference Proceedings*, 2734(1). <https://doi.org/10.1063/5.0155337/2917156>
- JustCoded. (2024). *5 key benefits of using scrum for mobile app development | JustCoded*. <https://justcoded.com/blog/5-key-benefits-of-using-scrum-for-mobile-app-development/>
- Khan, U., Martinez, D., & Baeldung. (2024). *Load Testing Rest Endpoint Using Gatling | Baeldung*. <https://www.baeldung.com/gatling-load-testing-rest-endpoint>
- Khandelwal, A. (2020). *12 Best Load Testing tools for mobile Applications | What is Load testing*. <https://testinggenex.com/load-testing-tools-for-mobile-applications/>
- Kravchenko, T., Bogdanova, T., & Shevgunov, T. (2022). Ranking Requirements Using MoSCoW Methodology in Practice. *Lecture Notes in Networks and Systems*, 503 LNNS, 188–199. https://doi.org/10.1007/978-3-031-09073-8_18
- Laor, T., & Galily, Y. (2022). In WAZE we trust? GPS-based navigation application users' behavior and patterns of dependency. *PLOS ONE*, 17(11), e0276449. <https://doi.org/10.1371/JOURNAL.PONE.0276449>
- Lárusdóttir, M. K., Cajander, Á., & Simader, M. (2014). Continuous improvement in agile development practice: The case of value and non-value adding activities. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8742, 57–72. https://doi.org/10.1007/978-3-662-44811-3_4
- LEY ORGÁNICA DE VIGILANCIA Y SEGURIDAD PRIVADA, Pub. L. No. 496 (2024). www.asambleanacional.gob.ec
- Libre Scrum. (2020). *Técnicas que te ayudarán a Priorizar – Libre Scrum*. <https://librescrum.com/2020/09/22/tecnicas-de-priorizacion/>
- Lim, S., Henriksson, A., & Zdravkovic, J. (2021). Data-Driven Requirements Elicitation: A Systematic Literature Review. *SN Computer Science*, 2(1), 1–35. <https://doi.org/10.1007/S42979-020-00416-4/TABLES/10>
- Loaza Granda, J. M. (2019). *TEMA: IMPLEMENTACIÓN DE LA METODOLOGÍA SCRUM, EN EL DESARROLLO DEL SOFTWARE DE LA DIRECCIÓN*

NACIONAL DE COMUNICACIONES, EN LA SECCIÓN DE DESARROLLO DE SOFTWARE DE LA POLICÍA NACIONAL DEL ECUADOR. <http://repositorio.uisrael.edu.ec/handle/47000/2164>

- Maila-Maila, F., Intriago-Pazmiño, M., & Ibarra-Fiallo, J. (2019). Evaluation of Open Source Software for Testing Performance of Web Applications. *Advances in Intelligent Systems and Computing*, 931, 75–82. https://doi.org/10.1007/978-3-030-16184-2_8
- Mallidi, R. K., & Sharma, M. (2021). Study on Agile Story Point Estimation Techniques and Challenges. *International Journal of Computer Applications*, 174(13), 9–14. <https://doi.org/10.5120/IJCA2021921014>
- Martínez, M. Á. Q., Antón, O. A. J., Moran, D. H. P., & Vazquez, M. Y. L. (2020). Priorización de requisitos para una adecuada calidad de software. *Serie Científica de La Universidad de Las Ciencias Informáticas*, 13(6), 135–145. <https://publicaciones.uci.cu/index.php/serie/article/view/597>
- Matam, S., & Jain, J. (2017). Pro Apache JMeter. *Pro Apache JMeter*. <https://doi.org/10.1007/978-1-4842-2961-3>
- Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2020). Requirements specification for developers in agile projects: Evaluation by two industrial case studies. *Information and Software Technology*, 117, 106194. <https://doi.org/10.1016/J.INFSOF.2019.106194>
- Medina-Rojas, F., Nuñez-Santa, J. M., Sánchez-Medina, I. I., & Cabrera-Medina, J. M. (2017). Implementación del ABP, PBL y método SCRUM en cursos académicos para desarrollar sistemas informáticos enfocados en fortalecer la región. *Revista Educación En Ingeniería*, 12(24), 52. <https://doi.org/10.26507/REI.V12N24.758>
- Meeker, W. Q. ., Escobar, L. A. ., & Pascual, F. G. . (2022). *Statistical methods for reliability data*. 659. <https://www.wiley.com/en-us/Statistical+Methods+for+Reliability+Data%2C+2nd+Edition-p-9781118594483>
- Moreseg Cia. Ltda. (2024). *Seguridad Privada en Ecuador: Innovación y Servicios - Moreseg Cia. Ltda.* <https://moreseg.com/seguridad-privada-en-ecuador-innovacion-y-servicios/>
- Naciones Unidas. (2019). *Objetivos y metas de desarrollo sostenible - Desarrollo Sostenible*. <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- Node.js. (n.d.). *Node.js — About Node.js®*. Retrieved September 22, 2024, from <https://nodejs.org/en/about>
- Pecorelli, F., Catolino, G., Ferrucci, F., De Lucia, A., & Palomba, F. (2022). Software testing and Android applications: a large-scale empirical study. *Empirical Software Engineering*, 27(2), 1–41. <https://doi.org/10.1007/S10664-021-10059-5>/METRICS
- Perleche, B. L., Patricia, L., Bravo Ruiz, M., & Arturo, J. (2022). *FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS TESIS EVALUACIÓN DE LA USABILIDAD EN*

ENTORNOS PARA OPTAR EL TITULO PROFESIONAL DE INGENIERA DE SISTEMAS Autor (es). <https://orcid.org/0000-0002-1997-3908>

Ramírez Ramírez, M., Del Consuelo, M., Soto, S., Beatriz, H., Moreno, R., Rojas, E. M., Del Carmen, N., Millán, O., Fernando, R., & Cisneros, R. (2018). *Metodología SCRUM y desarrollo de Repositorio Digital*.

REGLAMENTO A LA LEY DE VIGILANCIA Y SEGURIDAD PRIVADA, Pub. L. No. 1181, Registro Oficial 383, 17 de julio de 2008 (2008). www.lexis.com.ec

Sachin, A. (2023). *Key Performance Test Metrics to Track | Blazemeter*. <https://www.blazemeter.com/blog/key-test-metrics-to-track>

Schmidtke, H. R. (2020). Location-aware systems or location-based services: a survey with applications to CoViD-19 contact tracking. *Journal of Reliable Intelligent Environments*, 6(4), 191–214. <https://doi.org/10.1007/S40860-020-00111-4/FIGURES/2>

Schwaber, K., & Sutherland, J. (2020a). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*.

Schwaber, K., & Sutherland, J. (2020b). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*.

SEBoK. (2024). *Guide to the Systems Engineering Body of Knowledge (SEBoK) Contents Articles Front Matter*. www.sebokwiki.org

Secretaría Nacional de Planificación. (2021). *Plan de Creación de Oportunidades 2021-2025 – Secretaría Nacional de Planificación*. <https://www.planificacion.gob.ec/plan-de-creacion-de-oportunidades-2021-2025/>

Servicio nacional de contratación pública. (2022). *OBJETO DE SELECCIÓN: SELECCIÓN DE PROVEEDORES PARA LA PRESTACIÓN DEL “SERVICIO DE VIGILANCIA Y SEGURIDAD PRIVADA FIJA” QUITO, MAYO 2022*.

Sheeba, A., Vinora, A., Ananth, P., Nithya, K., Nisha Jenipher, V., & Surya, U. (2023). Tracking and Monitoring of Soldiers Using IoT and GPS. *Lecture Notes in Networks and Systems*, 475, 53–63. https://doi.org/10.1007/978-981-19-2840-6_5

Silvertrac Software. (2024). *Silvertrac Software | Security Guard Management Software*. <https://www.silvertracsoftware.com/>

Skhosana, M., & Ezugwu, A. E. (2021). A Real-Time Machine Learning-Based Public Transport Bus-Passenger Information System. *International Journal of Sustainable Development and Planning*, 16(7), 1221–1238. <https://doi.org/10.18280/IJSDP.160703>

Steiner, J. (2024). Guía para pruebas de rendimiento de aplicaciones móviles | Blog | Digital.ai. <https://Digital.Ai/Es/>. <https://digital.ai/es/catalyst-blog/mobile-app-performance-testing/>

Sumner, S., JKirsch1, & Leavitt, S. (2023). *Recomendaciones para pruebas de rendimiento - Microsoft Azure Well-Architected Framework | Microsoft Learn*.

<https://learn.microsoft.com/es-es/azure/well-architected/performance-efficiency/performance-test>

Talreja, A. (2023). *Stress Testing: Pushing Your Application to the Limits*. <https://mastersoftwaretesting.com/testing-fundamentals/types-of-testing/stress-testing>

THERMS. (2023). *Security Guard Management: Everything You Need to Know — THERMS - Security Incident Report Software*. <https://www.therms.io/blog/security-guard-management/>

Trackforce. (2020). *Trackforce: Guard Management and Real-Time Tracking*. <https://www.trackforce.com/blog/improving-guard-management-with-gps-real-time-tracking/>

Velásquez, E. (2018). *¿Cuales son los requisitos para trabajar de guardia de seguridad?* <https://comoestrabajar.com/en-ecuador/cuales-son-los-requisitos-para-trabajar-de-guardia-de-seguridad-en-ecuador/>

Verwijns, C. (2020). *Thinking By Sprinting: What Cognitive Science Tells Us About Why Scrum Works* | *Scrum.org*. <https://www.scrum.org/resources/blog/thinking-sprinting-what-cognitive-science-tells-us-about-why-scrum-works>

Verwijns, C. (2021). *What Makes Scrum Teams Effective? A scientific investigation of 1.200 Scrum teams* | *Scrum.org*. <https://www.scrum.org/resources/blog/what-makes-scrum-teams-effective-scientific-investigation-1200-scrum-teams>

Weichbroth, P. (2024). Usability Testing of Mobile Applications: A Methodological Framework. *Applied Sciences* 2024, Vol. 14, Page 1792, 14(5), 1792. <https://doi.org/10.3390/APP14051792>

Xie, W., Peng, X., Liu, M., Treude, C., Xing, Z., Zhang, X., & Zhao, W. (2020). API method recommendation via explicit matching of functionality verb phrases. *ESEC/FSE 2020 - Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 1015–1026. https://doi.org/10.1145/3368089.3409731/SUPPL_FILE/FSE20MAIN-P445-P-VIDEO.MP4

Zangenehnejad, F., & Gao, Y. (2021). GNSS smartphones positioning: advances, challenges, opportunities, and future perspectives. *Satellite Navigation*, 2(1), 1–23. <https://doi.org/10.1186/S43020-021-00054-Y/TABLES/5>

Zarour, M. (2020). A rigorous user needs experience evaluation method based on software quality standards. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(5), 2787–2799. <https://doi.org/10.12928/TELKOMNIKA.V18I5.16061>

Żyłuk, A., Zieja, M., Grzesik, N., Tomaszewska, J., Kozłowski, G., & Jaształ, M. (2023). Implementation of the Mean Time to Failure Indicator in the Control of the Logistical Support of the Operation Process. *Applied Sciences* 2023, Vol. 13, Page 4608, 13(7), 4608. <https://doi.org/10.3390/APP13074608>

ANEXOS

Link de la hoja de cálculo de la bitácora de pruebas manuales:

https://docs.google.com/spreadsheets/d/1txIS2m9j0gnQ7IFzAZ4iFVI52HOvFIK/edit?usp=drive_link&oid=116973616191311508683&rtpof=true&sd=true

Esquema del prototipo móvil con Balsamiq Wireframes:

https://drive.google.com/file/d/14Pgf4L3eZmIPm4yISLTPpYuKneBcyIDF/view?usp=drive_link

Esquema del prototipo web con Figma:

<https://www.figma.com/design/UMch9t1mAKzBi1IOK0xcUw/Untitled?node-id=0-1&t=giPkeOqAg0igA3yl-1>