



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

***“Sistema Web y Móvil de gestión de incidencias vía PHP  
y Flutter para la mejora de las infraestructuras públicas  
del distrito Gregorio Albarracín”***

*Curso: Construcción de Software I*

*Docente: Ing. Flor Rodríguez, Alberto Jonathan*

Integrantes:

***Hurtado Ortiz, Leandro***

***(2015052384)***

***Castañeda Centurion, Jorge Enrique***

***(2021069822)***

**Tacna – Perú**

**2025**



# **Sistema Web y Móvil de gestión de incidencias vía PHP y Flutter para la mejora de las infraestructuras públicas del distrito Gregorio Albarracín**

## **Documento de Especificación de Requerimientos de Software**

**Versión 3.0**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	JECC	JECC	JECC	19/03/2025	Versión Original
2.0	LHO	LHO	LHO	24/03/2025	Versión 2.0
3.0	JECC	JECC	JECC	31/03/2025	Versión 3.0
4.0	JECC	JECC	JECC	07/04/2025	Versión 4.0



## ÍNDICE GENERAL

### Contenido

Introducción	3
I. Generalidades de la Empresa	4
1. Nombre de la Empresa:	4
2. Visión	4
3. Misión	4
4. Organigrama	5
II. Visionamiento de la Empresa:	5
1. Descripción del Problema	5
2. Objetivos de Negocios	5
3. Objetivos de Diseño	6
4. Alcance del proyecto	6
5. Viabilidad del sistema	7
III. Análisis de Procesos	9
a) Diagrama del Proceso Propuesto - Diagrama de actividades Inicial	9
IV. Especificación de Requerimientos de Software	10
a) Cuadro de Requerimientos No Funcionales	10
b) Cuadro de Requerimientos Funcionales Final	10
c) Reglas de Negocio	13
V. Fases de Desarrollo	15
1. Perfiles de Usuario	15
2. Modelo Conceptual	16
a) Diagrama de Paquetes	16
b) Diagrama de Casos de Uso	16
c) Escenarios de Caso de Uso (narrativa)	16
3. Modelo Lógico	17
a) Diagrama de Actividades con Objetos	17
b) Diagrama de Secuencia	17
c) Diagrama de Clases	18
Conclusiones	18
Recomendaciones	18



## **Documento de Especificación de Requerimientos de Software**

### **Introducción**

Este documento de especificación de requerimientos de software tiene como objetivo detallar los aspectos fundamentales para el desarrollo e implementación de un sistema tecnológico vía Móvil que permita a los usuarios poder registrarse, ubicar el lugar de la incidencia y realizar el reporte junto a una descripción; mientras que los empleados pueden marcar su asistencia y recibir las tareas encargadas por el administrador para responder rápidamente a las incidencias. En este contexto, se busca modernizar el modo de detección y gestión de las incidencias en infraestructuras, además de que también busca fomentar una mayor participación ciudadana en la vigilancia mejorando el entorno urbano, todo gracias al crowdsourcing.

El sistema propuesto deberá satisfacer las necesidades de los usuarios finales, integrándose con las plataformas existentes y asegurando la escalabilidad, seguridad y confiabilidad de la solución. Además, se brindará un enfoque integral al diseño y desarrollo del sistema, teniendo en cuenta tanto los aspectos técnicos como los operativos y económicos, para garantizar su viabilidad a largo plazo. Este documento está estructurado para proporcionar una visión clara de los objetivos, los requisitos y las soluciones necesarias, permitiendo una correcta implementación y alineación con los intereses del negocio y las expectativas de los usuarios.



## I. Generalidades de la Empresa

### **1. Nombre de la Empresa:**

Sistema de gestión de incidentes en infraestructuras basado en Crowdsourcing

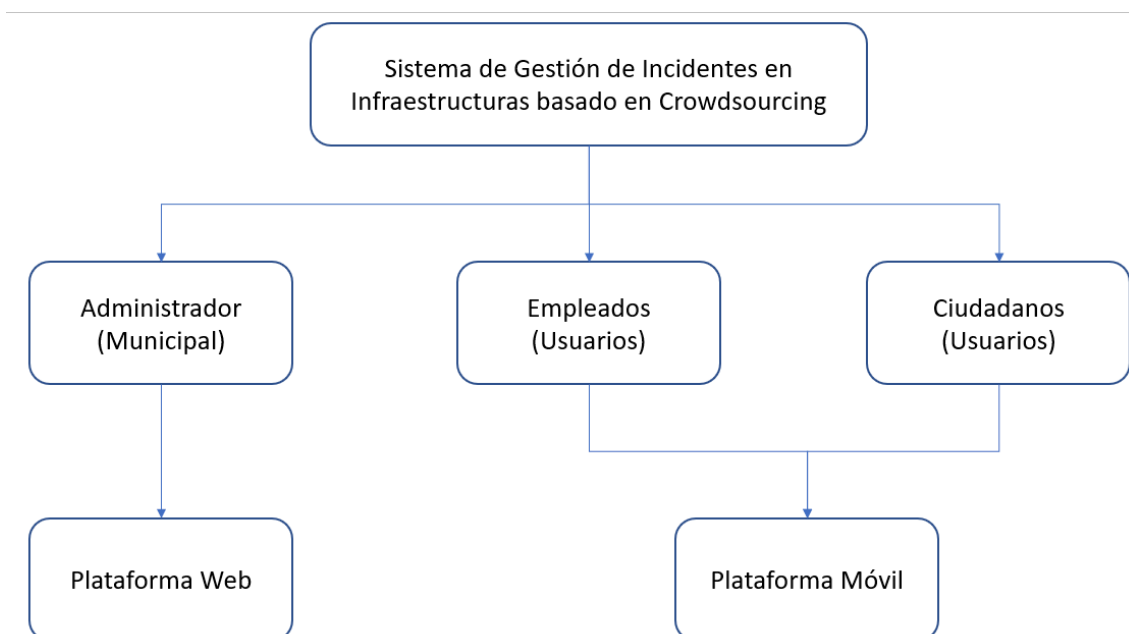
### **2. Visión**

Ser la plataforma digital líder en la gestión de incidentes urbanos mediante la participación ciudadana, optimizando la comunicación entre la comunidad y las autoridades para mejorar la infraestructura del distrito de Gregorio Albarracín Lanchipa.

### **3. Misión**

Facilitar el reporte y gestión de incidentes urbanos a través de una plataforma intuitiva y accesible, permitiendo a los ciudadanos informar problemas en tiempo real y a las autoridades priorizar y resolver incidencias de manera eficiente.

### **4. Organigrama**





## II. Visionamiento de la Empresa:

### 1. Descripción del Problema

Las fallas en infraestructura (baches, aceras deterioradas, mala señalización) no se reportan oportunamente, afectando la calidad de vida y seguridad. La ausencia de un sistema centralizado dificulta la toma de decisiones rápidas y efectivas.

### 2. Objetivos de Negocios

El **Sistema de Gestión de Incidentes en Infraestructuras basado en Crowdsourcing** tiene como objetivos principales:

- Facilitar el proceso de creación de los reportes de incidencias por parte de los ciudadanos.
- Construir un módulo de administrador con funcionalidades para gestionar las incidencias reportadas.
- Permitir el monitoreo en tiempo real de la data de las incidencias.
- Implementar una asignación de recursos por parte del administrador para la reparación de las infraestructuras públicas.

### 3. Objetivos de Diseño

Para cumplir con los objetivos de negocio, el sistema se diseñará bajo los siguientes principios:

- **Usabilidad:** Interfaz intuitiva y accesible tanto para ciudadanos como para autoridades municipales.
- **Escalabilidad:** Posibilidad de ampliar las funcionalidades y cobertura del sistema en el futuro.
- **Interoperabilidad:** Integración con APIs de mapas y bases de datos existentes.
- **Seguridad:** Protección de los datos personales de los usuarios mediante encriptación y autenticación segura.
- **Disponibilidad:** Accesibilidad desde navegadores web y dispositivos móviles.



#### **4. Alcance del proyecto**

##### **Inclusiones:**

- Desarrollo de un sistema de reportes de incidencias para la versión móvil y un sistema de gestión de los reportes para la versión Web, ambos conectados a una misma base de datos.
- Interfaz de usuario accesible para dispositivos móviles.
- Interfaz de usuario accesible para administradores en web php.
- Funcionalidades de gestión y reportes.
- Integración con APIS como intermediarios.
- Módulo de autenticación de usuarios.

##### **Exclusiones:**

- Desarrollo de aplicaciones móviles específicas para cada plataforma (iOS y Android), aunque la interfaz será accesible desde dispositivos móviles.
- Soporte para idiomas adicionales fuera del español e inglés en una fase inicial.

#### **5. Viabilidad del sistema**

##### **1.1. Viabilidad Técnica**

##### **Hardware Disponible**

- Dispositivos móviles para los usuarios y de escritorio para los administradores
- Infraestructura para alojar bases de datos y aplicaciones (servidores locales)

##### **Equipos de desarrollo**

- Desarrolladores en Php, Flutter, y PostgreSQL
- Administradores de base de datos para gestionar la infraestructura de datos.



### Software

- Herramientas de desarrollo: Visual Studio Code, Php, Android Studio, Flutter, PostgreSQL.
- Sistemas operativos: Windows.

### Aplicaciones y Herramientas de Desarrollo

- Php para el backend, la conexión a bases de datos y el sistema del administrador.
- Flutter para la creación del sistema para los usuarios (ciudadanos y empleados).
- Herramientas de integración de API para la conexión entre los datos de Flutter y Php.
- Composer para la gestión de dependencias del backend en Php, lo que facilita la instalación y actualización de librerías necesarias.

### Navegadores Web

- Compatible con los navegadores más utilizados: Google Chrome, Mozilla Firefox, Microsoft Edge.

## 1.2. Viabilidad Operativa

- **Beneficios para los Usuarios:**

El proyecto ofrecerá una plataforma segura y eficiente para generar los reportes de las incidencias y sus respectivas gestiones gracias al administrador en la versión Web, y con una API backend, los usuarios podrán acceder de manera segura al sistema móvil, gracias al API de Google Maps podrán encontrar más fácilmente el lugar donde visualizaron la incidencia para así comenzar con el proceso de reporte.

### **Beneficios clave para los usuarios incluyen:**

1. **Acceso seguro y centralizado:** Los usuarios podrán acceder a los sistemas internos de manera más segura mediante la autenticación con el backend realizado en PHP.



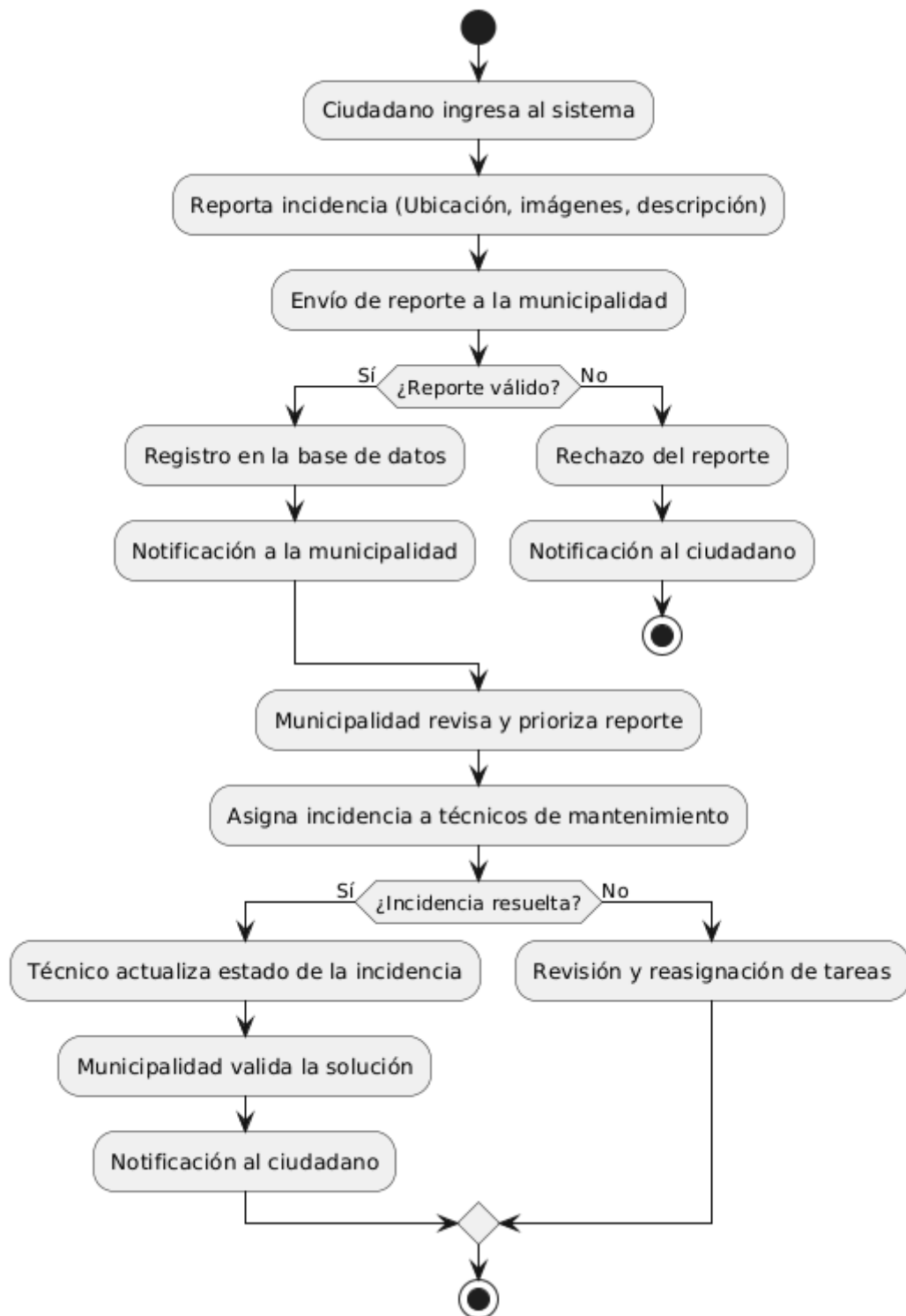


2. **Gestión eficiente de permisos:** Los usuarios tendrán acceso solo a los recursos permitidos según su rol, lo que facilita la gestión de permisos y mejora la seguridad al evitar accesos no autorizados.
3. **Rendimiento mejorado:** Con la integración de PostgreSQL como base de datos, las consultas y operaciones relacionadas con los usuarios y roles serán rápidas y eficientes, garantizando tiempos de respuesta bajos y un sistema más ágil.
4. **Cumplimiento de normativas:** El sistema también garantizará que los datos personales estén protegidos y se cumpla con normativas internacionales como la ley 29733, lo que generará mayor confianza entre los usuarios.



### III. Análisis de Procesos

#### a) Diagrama del Proceso Propuesto - Diagrama de actividades Inicial





*Diagrama 01: Este diagrama proporciona una vista clara del flujo del sistema, destacando las responsabilidades y las interacciones entre los usuarios y el sistema. Esto permite identificar puntos clave de interacción y garantizar que cada funcionalidad esté correctamente implementada en el sistema.*

#### IV. Especificación de Requerimientos de Software

##### a) **Cuadro de Requerimientos No Funcionales**

<b>Código</b>	<b>Requerimiento</b>	<b>Descripción</b>
RNF-01	Seguridad	El sistema debe encriptar las contraseñas para las sesiones con el uso de token JWT.
RNF-02	Rendimiento	El sistema debe tener un buen rendimiento gracias a operaciones relacionadas con la base de datos usando buenas prácticas como autoloader y namespaces, además de minimizar el uso de funciones globales y trabajar con frameworks, evitando en todo momento el uso de las consultas SQL en posibles loops del código general.
RNF-03	Usabilidad	El sistema debe tener interfaces sencillas de usar tanto para administradores como para usuarios generales.



## b) Cuadro de Requerimientos Funcionales Final

Código	Requerimiento	Descripción
RF-01	Reporte de incidencias	Permitir a los ciudadanos registrar incidentes a través de la aplicación, proporcionando una descripción, imágenes y geolocalización.
RF-02	Autenticación de Usuarios	Permitir a los empleados registrarse e iniciar sesión en la aplicación.
RF-03	Monitoreo y Actualización de Incidencias	Permitir a los empleados visualizar una lista de las incidencias asignadas y actualizar su estado.
RF-04	Autenticación de Administradores	Permitir a los administradores registrarse e iniciar sesión en la plataforma.
RF-05	Gestión del Dashboard	Permitir a los administradores asignar incidencias a empleados y supervisar su estado a través de un dashboard.
RF-06	Generación de Reportes de Incidencias	Permitir a los administradores generar reportes consolidados sobre las incidencias registradas.



### c) Reglas de Negocio

Nombre	Descripción	Autoridad
Autenticación	Solo los empleados registrados y con cuenta verificada pueden recibir las incidencias asignadas por los administradores	Sistema
Asignación de Tareas	Solo el administrador puede asignar incidencias a empleados registrados.	Administrador
Estados de Incidencia	Una incidencia debe pasar por los estados: Pendiente, En proceso y Resuelta. No puede retroceder a un estado anterior.	Administrador
Subida de Evidencias	Los reportes de incidencias deben incluir al menos una foto y una descripción detallada.	Usuario
Ubicación Georreferenciada	Toda incidencia debe registrarse con coordenadas geográficas obtenidas del mapa.	API
Reportes y Estadísticas	Se deben generar informes sobre incidencias atendidas	Administrador
Integridad de Datos	No se pueden eliminar reportes de incidencias una vez creados, solo se pueden marcar como resueltos.	Administrador
Cumplimiento Legal	El sistema debe cumplir con la normativa de protección de datos personales vigente en el Perú.	Departamento legal



## V. Fases de Desarrollo

### 1. **Perfiles de Usuario**

- Ciudadano (Usuario)

El usuario principal de la aplicación móvil es el ciudadano. Su función consiste en reportar problemas relacionados con infraestructuras públicas de manera eficiente y sencilla. Después de ingresar anónimamente, el ciudadano tiene acceso a un mapa interactivo donde puede señalar la ubicación exacta del problema. Luego, debe llenar un formulario ofreciendo una descripción detallada del inconveniente y subir una foto como prueba.

- Empleado (Usuario)

El trabajador del municipio es un individuo que se ha registrado en la aplicación móvil, encargado de gestionar los problemas reportados por los ciudadanos. Para ingresar al sistema, necesita registrarse e iniciar sesión. Después de autenticarse, puede recibir alertas cuando le asignan una nueva tarea. Su labor implica ir al sitio del problema, analizar la situación y realizar las acciones necesarias para solucionarlo. Cuando termine con la tarea, debe actualizar el estado del inconveniente.

- Administrador

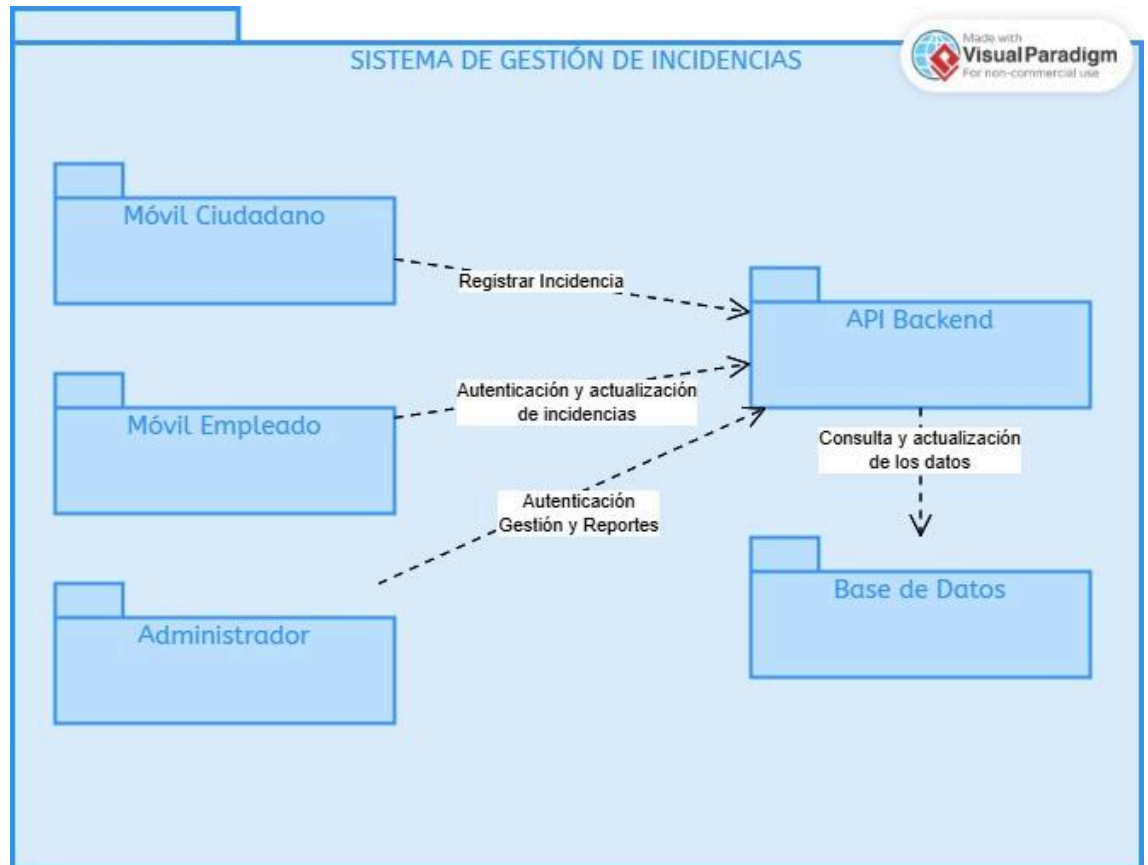
El administrador es el usuario con los niveles más altos de acceso en el sistema y entra a la plataforma mediante la aplicación en línea. Su rol principal consiste en manejar las incidencias que se han reportado, verificando la información entregada por los ciudadanos y asignándole a los trabajadores municipales. Además, el administrador está en posición de observar el rendimiento de los empleados y elaborar reportes estadísticos sobre las incidencias resueltas. También tiene la capacidad de acceder a un calendario para programar actividades y mejorar la administración de recursos. Así mismo, es responsable de la seguridad y mantenimiento del sistema, asegurándose de que solo los usuarios autorizados puedan acceder a las diversas funcionalidades.



## 2. Modelo Conceptual

### a) Diagrama de Paquetes

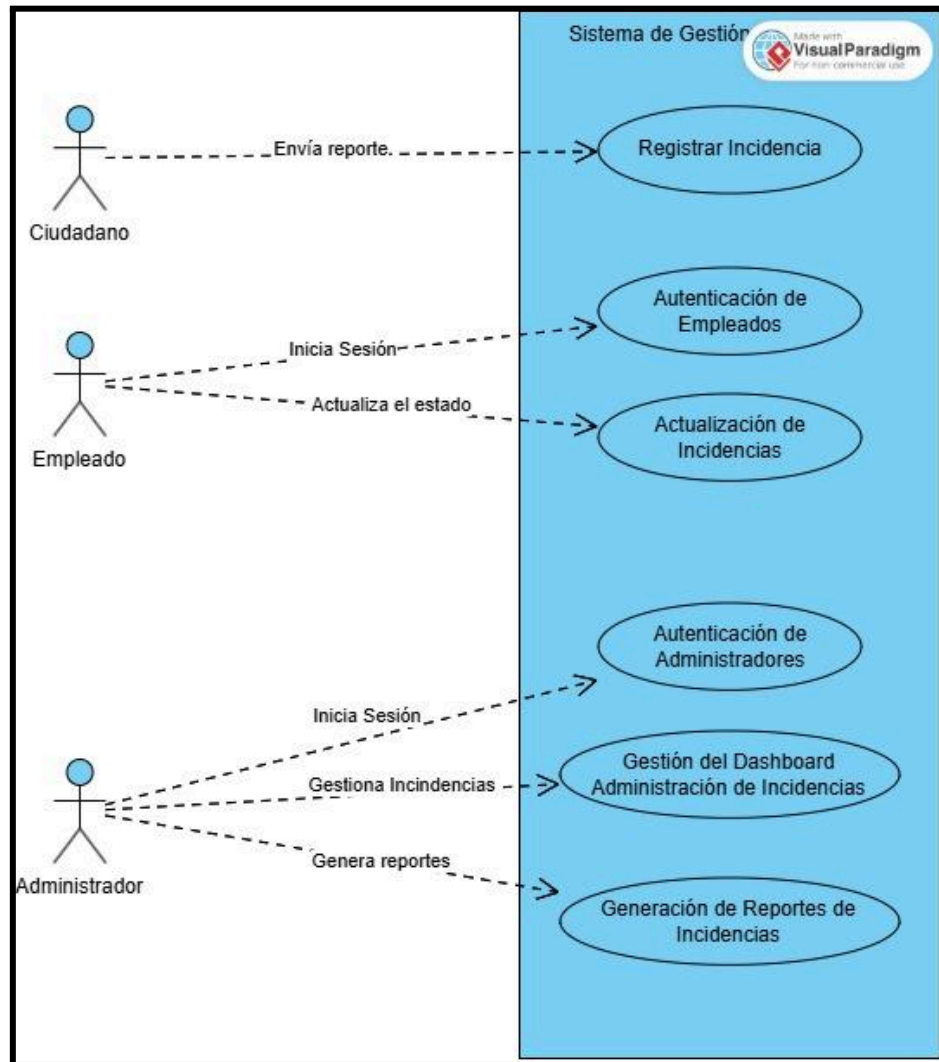
#### Diagrama de Paquetes del API



*El diagrama muestra la estructura del proyecto StackBoard con los subpaquetes principales: Gestión de Cambio, Gestión de Proyectos, Gestión de Metodologías y Gestión de Equipos/Usuarios.*

b) Diagrama de Casos de Uso

Diagrama de Casos de Uso de la Plataforma







c) Escenarios de Caso de Uso (narrativa)

**Narrativa de CU01 - Reporte de Incidencias**

<b>Caso de Uso</b>	<b>Reporte de Incidencias</b>	<b>CU01</b>
<b>Actores</b>	Ciudadano	
<b>Descripción</b>	Permite a los ciudadanos ingresar en la plataforma móvil para reportar una incidencia	
<b>Precondiciones</b>	- El dispositivo debe estar conectado a internet.	
<b>FLUJO NORMAL</b>	<b>Acción:</b>	
	<b>ACCIÓN DEL ACTOR</b>	<b>CURSO DEL SISTEMA</b>
	1. El usuario ingresa a la app.	2. El sistema muestra el menú de opciones para el ciudadano
	3. Selecciona la opción "Reportar Incidencia" en la interfaz.	4. El sistema despliega el formulario para reportar incidencias, incluyendo un mapa integrado, un campo para la descripción y una opción para adjuntar una foto.
	5. El ciudadano ubica el punto de la incidencia en el mapa, confirma la ubicación, adjunta una foto y escribe la descripción del problema.	
	6. Al finalizar, presiona el botón "Enviar".	7. La aplicación empaqueta los datos y los envía al API Backend mediante una solicitud REST o similar.



		8. El API Backend recibe la solicitud, valida la información.
		9. Una vez validado y transformado, el API Backend inserta los datos en la base de datos PostgreSQL.
		10. La aplicación móvil notifica al ciudadano que su incidencia ha sido registrada correctamente, mostrando un mensaje de confirmación.



## Narrativa de CU02 - Autenticación de Usuarios

<b>Caso de Uso</b>	Autenticación de Usuarios	<b>CU02</b>
<b>Actores</b>	Empleado	
<b>Descripción</b>	Permite a los usuarios autenticarse en el sistema ingresando sus credenciales.	
<b>Precondiciones</b>		
<b>FLUJO NORMAL</b>	<b>Acción:</b>	
	<b>ACCIÓN DEL ACTOR</b>	<b>CURSO DEL SISTEMA</b>
	1. El usuario abre la aplicación móvil y selecciona la opción “Registro” en la pantalla de inicio dependiendo si es ciudadano o empleado.	2. El sistema despliega el formulario de registro donde se solicitan datos personales, incluido el DNI, correo electrónico y contraseña.
	3. El empleado ingresa sus datos y confirma el registro.	4. La aplicación envía la información de registro al API Backend.
		5. El API Backend valida los datos.
		6. Encripta la contraseña y registra el nuevo empleado en la base de datos PostgreSQL.
		7. Devuelve una respuesta confirmando que el registro fue exitoso.
	8. Posteriormente, el empleado selecciona “Inicio de Sesión” e ingresa su correo y contraseña.	9. La aplicación envía las credenciales al API Backend para validarlas.
		10. El API Backend verifica que las



		credenciales sean correctas.
		11. La aplicación móvil establece la sesión del empleado y lo redirige a una pantalla para continuar con la visualización de incidencias.



### Narrativa de CU03 - Monitoreo y Actualización de Incidencias

<b>Caso de Uso</b>	Monitoreo y Actualización de Incidencias	<b>CU03</b>
<b>Actores</b>	Empleado	
<b>Descripción</b>	Permite a los empleados recibir las incidencias a las que fueron asignados y actualizar el estado.	
<b>Precondiciones</b>		
<b>FLUJO NORMAL</b>	<b>Acción:</b>	
	<b>ACCIÓN DEL ACTOR</b>	<b>CURSO DEL SISTEMA</b>
	1. Una vez autenticado, el empleado accede a la pantalla donde se muestran las incidencias asignadas.	2. La aplicación solicita a través del API Backend la lista de incidencias asignadas.
		3. El API Backend consulta la base de datos PostgreSQL filtrando las incidencias asociadas al empleado y envía la información de vuelta.
		4. La aplicación muestra la lista de incidencias con detalles básico
	5. El empleado selecciona una incidencia para ver sus detalles y decide actualizar el estado	6. La acción de actualización se envía al API Backend con la nueva información de estado y el identificador de la incidencia.
		7. El API Backend valida el cambio de estado y actualiza el registro correspondiente en la base de datos.
		8. La aplicación móvil actualiza la vista de la



		incidencia para reflejar el nuevo estado.
--	--	---

#### Narrativa de CU04 - Autenticación de Administradores

<b>Caso de Uso</b>	Autenticación de Administradores	<b>CU04</b>
<b>Actores</b>	Administrador	
<b>Descripción</b>	Permite a los administradores iniciar sesión en el sistema.	
<b>Precondiciones</b>		
<b>FLUJO NORMAL</b>	<b>Acción:</b>	
	<b>ACCIÓN DEL ACTOR</b>	<b>CURSO DEL SISTEMA</b>
	1. El administrador accede a la versión web e ingresa sus credenciales en la pantalla de login	2. La aplicación web envía las credenciales al API Backend para validarlas.
		3. Verifica la autenticidad de las credenciales contra la base de datos.
		4. Devuelve una respuesta de éxito
		5. El sistema redirige al administrador al Dashboard del sistema.



### Narrativa de CU05 - Gestión del Dashboard

<b>Caso de Uso</b>	Gestión del Dashboard	<b>CU05</b>
<b>Actores</b>	Administrador	
<b>Descripción</b>	Permite al administrador visualizar y gestionar el estado general de las incidencias reportadas a través de un dashboard.	
<b>Precondiciones</b>		
<b>FLUJO NORMAL</b>	<b>Acción:</b>	
	<b>ACCIÓN DEL ACTOR</b>	<b>CURSO DEL SISTEMA</b>
	1. Desde el Dashboard, el administrador procede a consultar el estado de todas las incidencias.	
	2. Selecciona una incidencia para ver detalles y asignarla a un empleado o modificar su estado	3. La aplicación web solicita la información de incidencias al API Backend, que consulta la base de datos y devuelve los datos actualizados.
		4. El API Backend procesa la solicitud y envía la información necesaria al sistema web.
		5. El Dashboard actualiza la vista para mostrar el estado actual de las incidencias.
		6. Al asignar una incidencia, el sistema envía la acción al API Backend, el cual actualiza la base de datos y confirma la asignación.
	7. El administrador recibe confirmación visual de que la incidencia ha sido asignada o	



	modificada correctamente.	
--	------------------------------	--

### Narrativa de CU06 - Generación de Reportes

<b>Caso de Uso</b>	Generación de Reportes	<b>CU06</b>
<b>Actores</b>	Administrador	
<b>Descripción</b>	Permite al administrador generar reportes generales de las incidencias actualizadas.	
<b>Precondiciones</b>		
<b>FLUJO NORMAL</b>	<b>Acción:</b>	
	<b>ACCIÓN DEL ACTOR</b>	<b>CURSO DEL SISTEMA</b>
	1. Desde el Dashboard, el administrador selecciona la opción "Generar Reporte".	2. Se despliega un formulario para configurar parámetros del reporte.
	3. El administrador introduce los parámetros deseados y confirma la generación del reporte.	4. El sistema web transforma los datos en el formato requerido para el reporte.
		5. El sistema web muestra un mensaje de confirmación y permite al administrador descargar o visualizar el reporte generado.

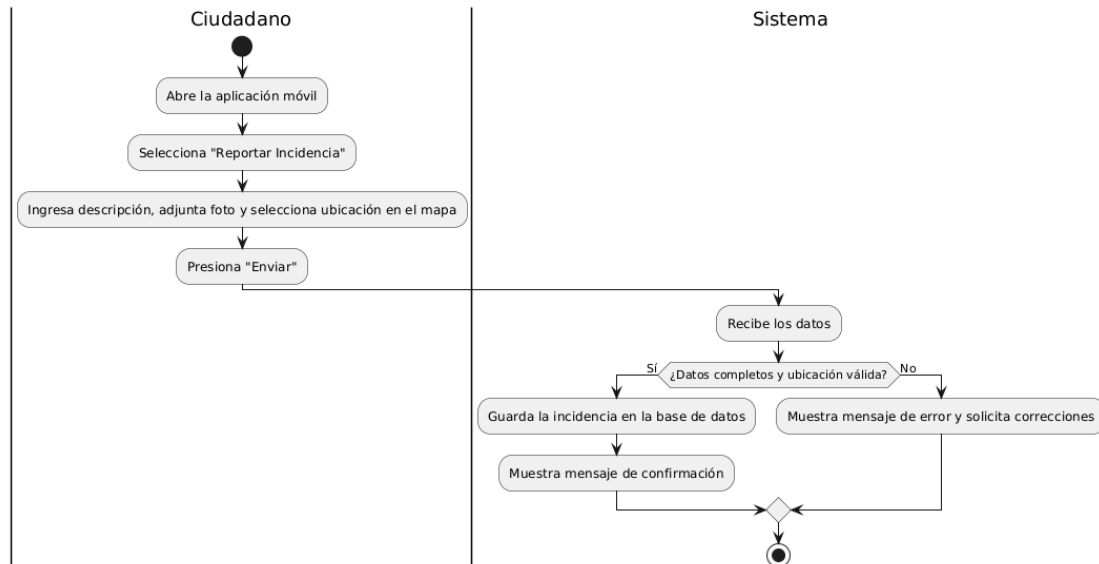




### 3. Modelo Lógico

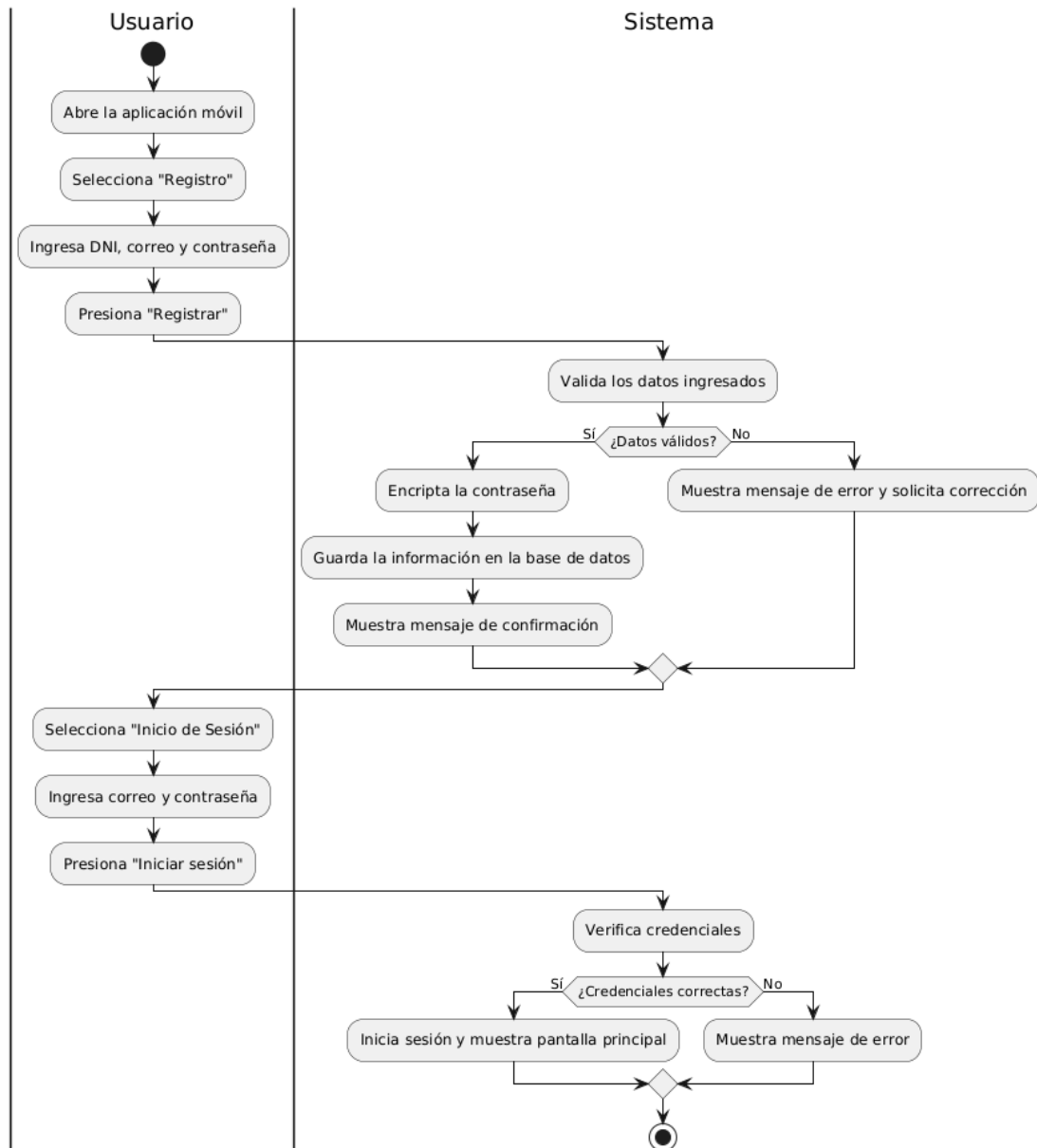
#### a) Diagrama de Actividades con Objetos

##### RF-01: Reporte de Incidencias



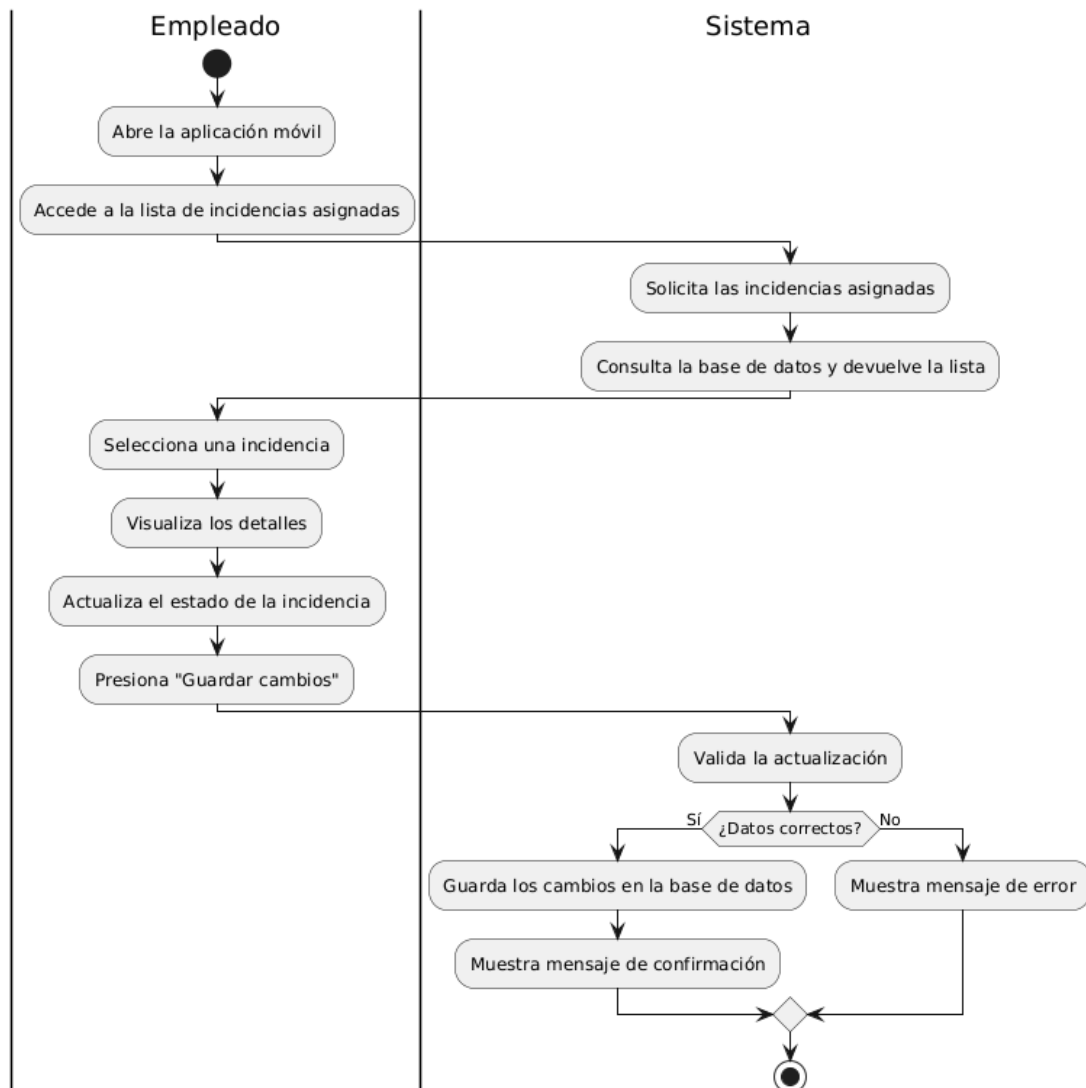


## RF-02: Autenticación de Usuarios



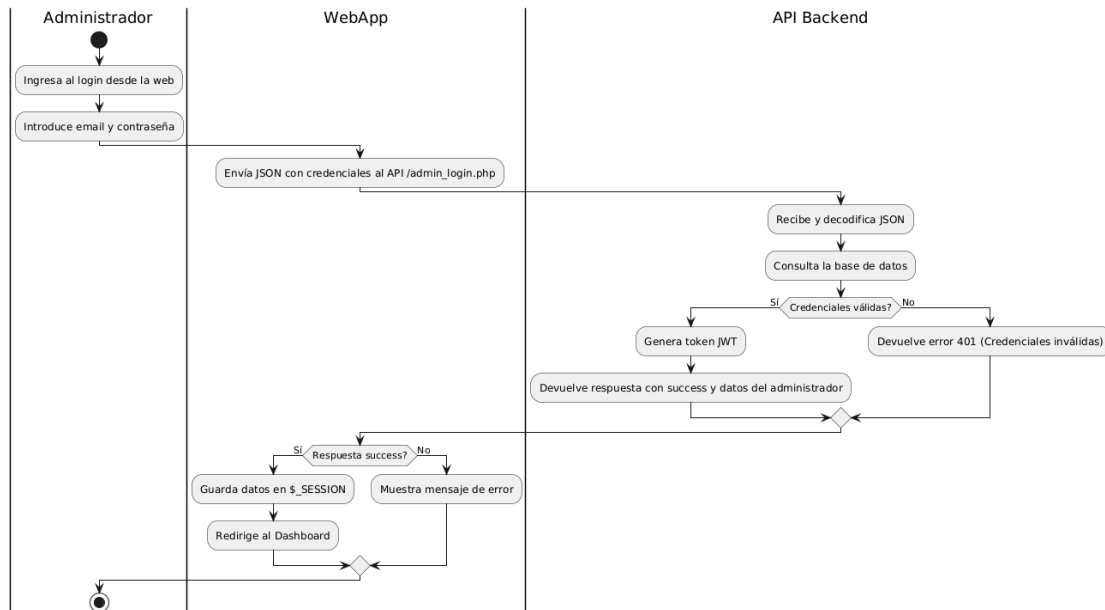


### RF-03: Monitoreo y Actualización de Incidencias

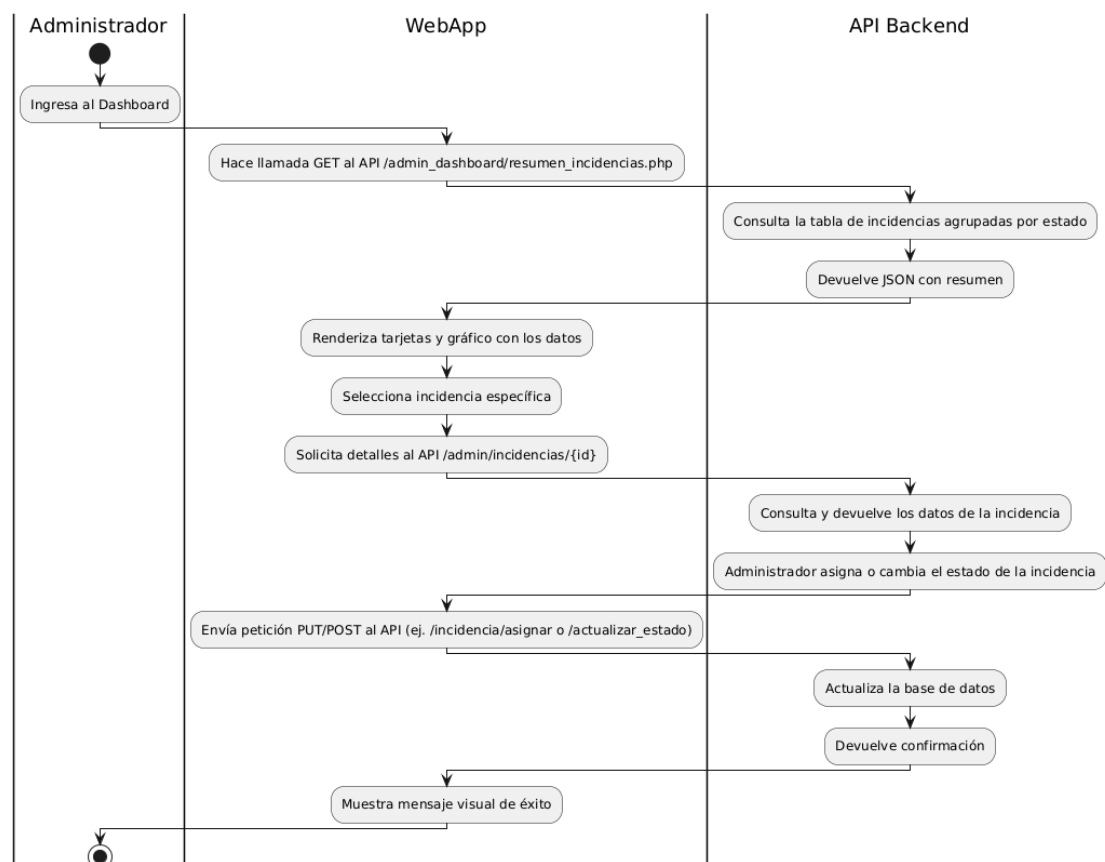




## RF-04: Autenticación de Administradores

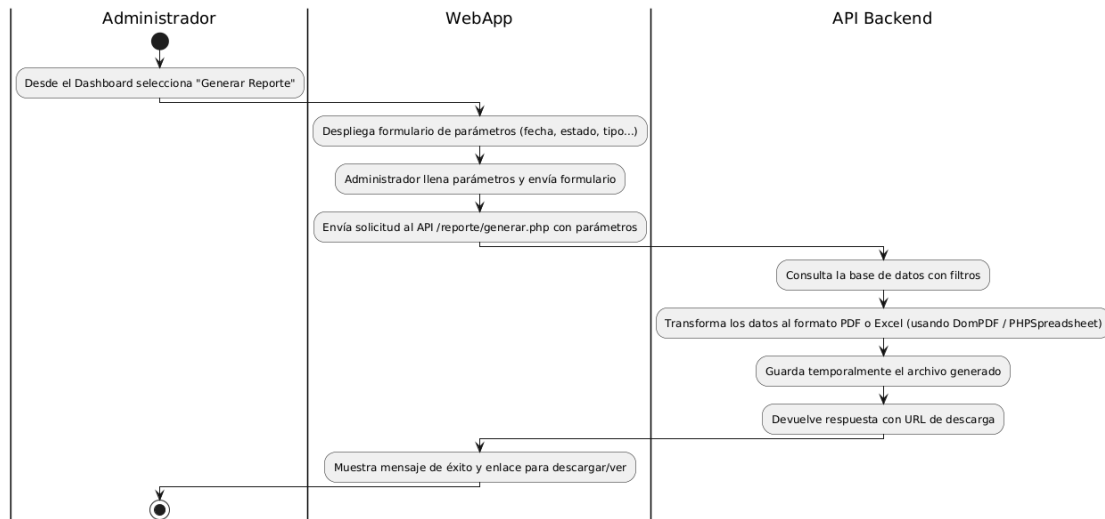


## RF-05: Gestión del Dashboard





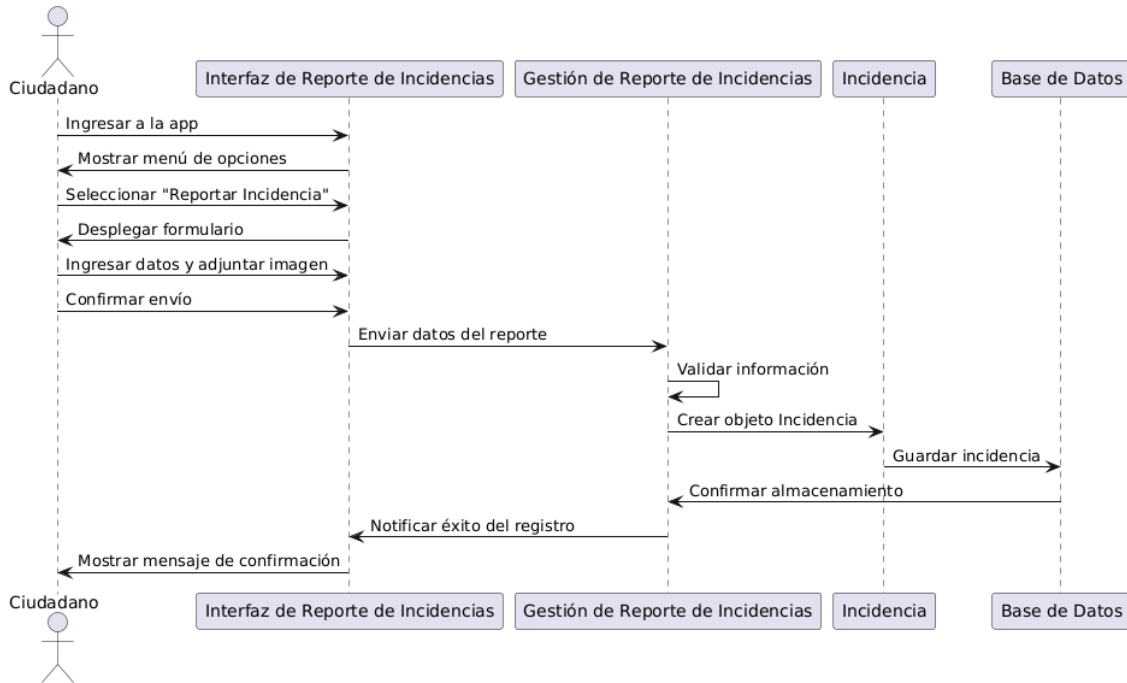
## RF-06: Generación de Reportes de Incidencias



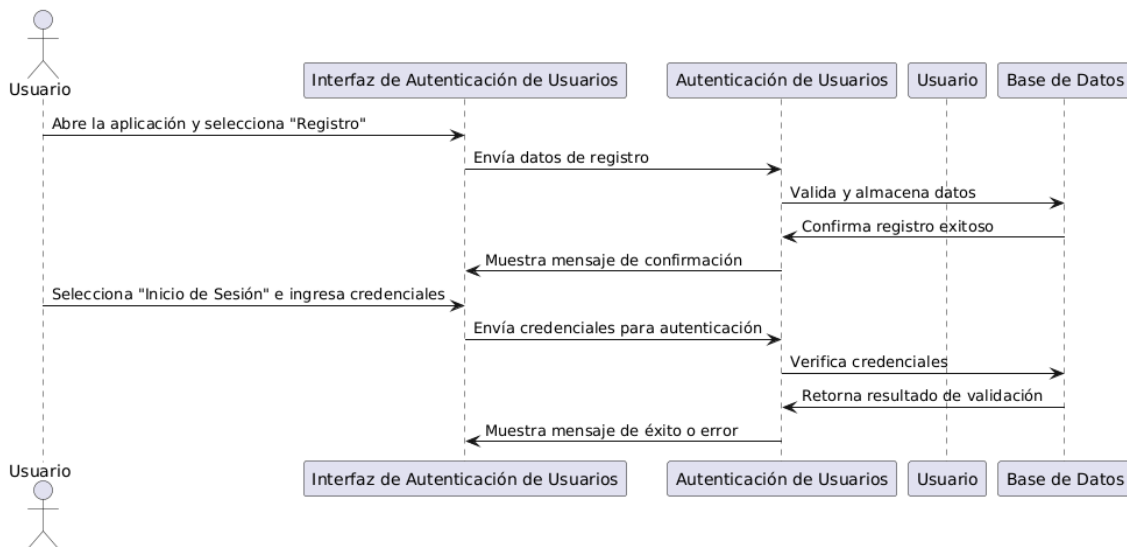


## b) Diagrama de Secuencia

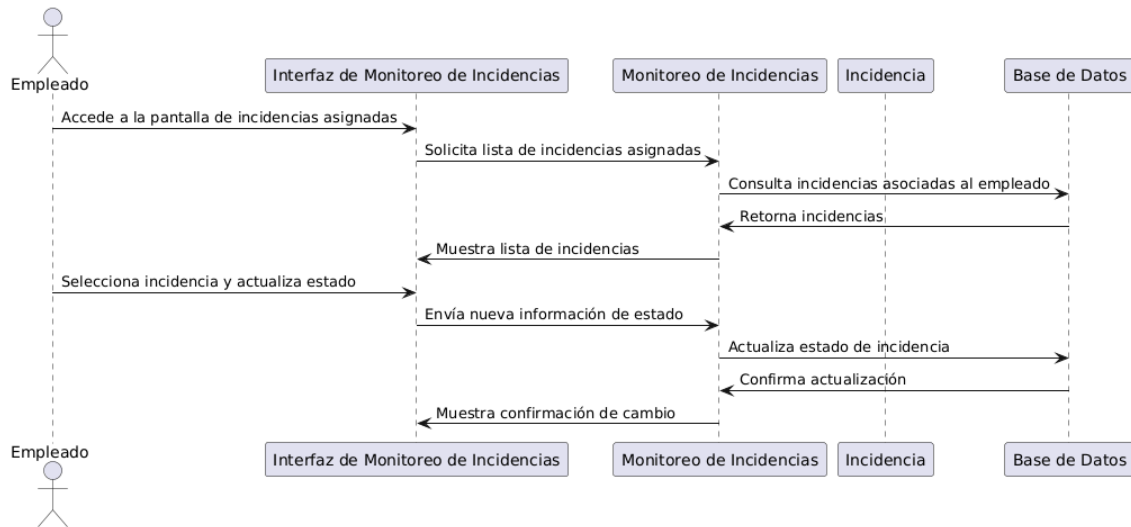
### Diagrama de secuencia: Reporte de incidencias



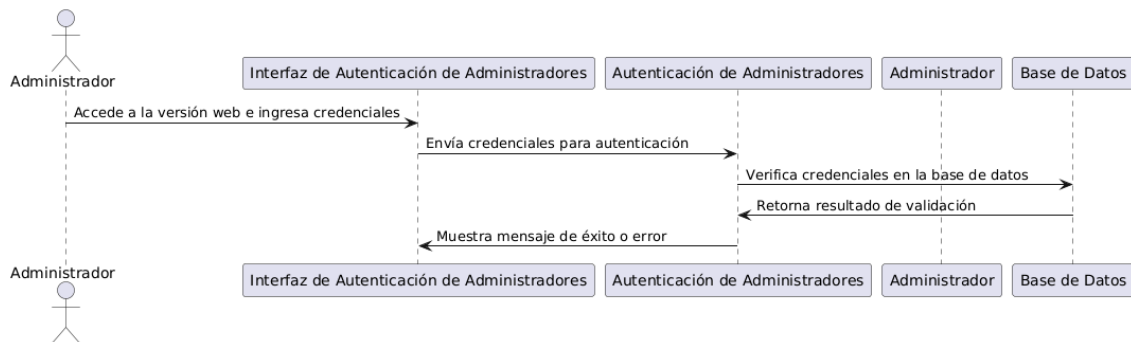
### Diagrama de secuencia: Autenticación de Usuarios



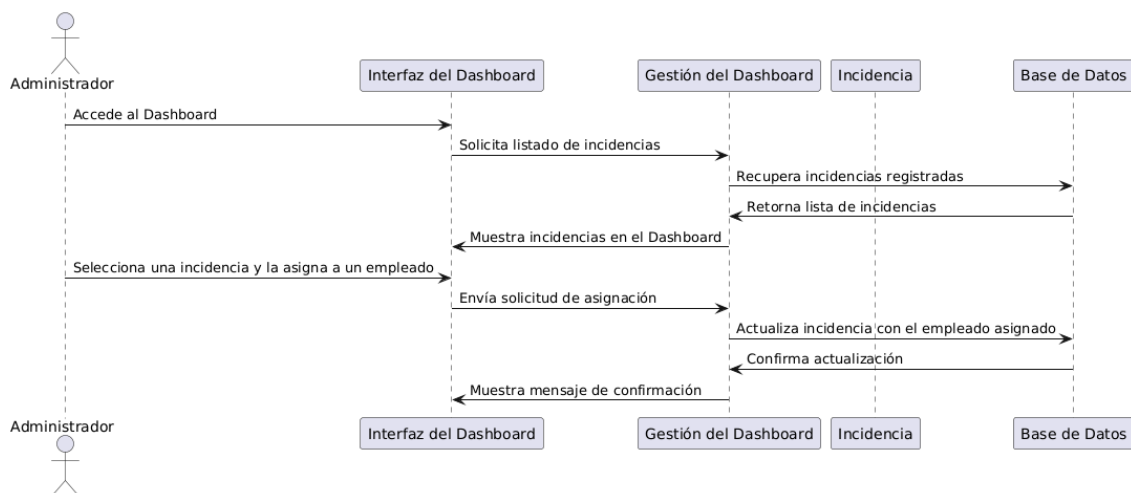
## Diagrama de secuencia: Monitoreo y Actualización de Incidencias



## Diagrama de secuencia: Autenticación de Administradores

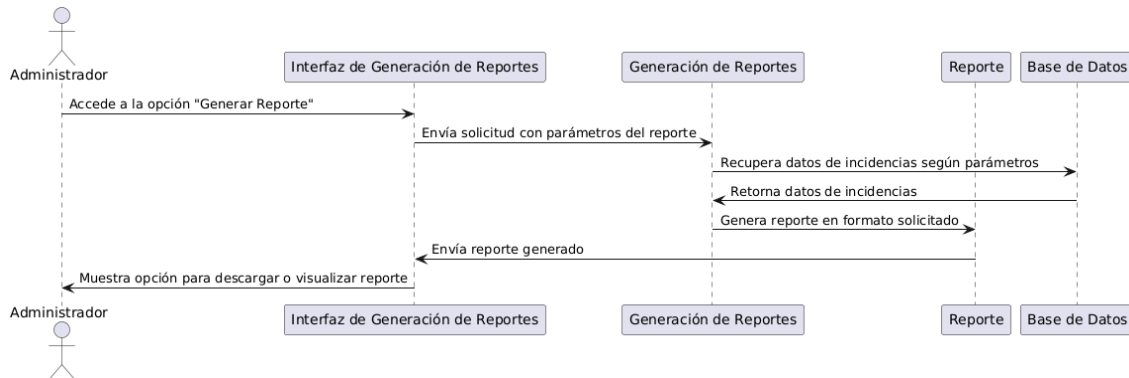


## Diagrama de secuencia: Gestión del Dashboard

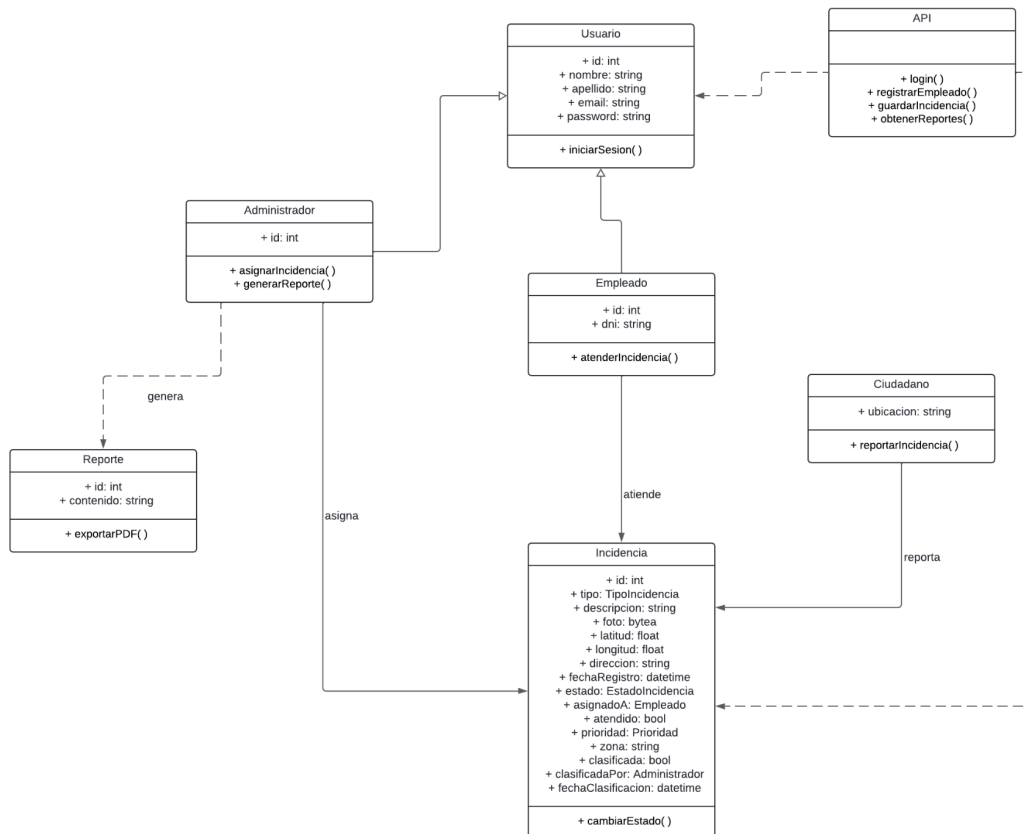




## Diagrama de secuencia: Generación de Reportes de Incidencias



## c) Diagrama de Clases







## **Conclusiones**

El presente documento SRS ha permitido estructurar de manera clara y detallada los requisitos del **Sistema Web de Gestión de Incidentes en Infraestructuras basado en Crowdsourcing**, estableciendo una base sólida para su desarrollo e implementación. A través del análisis del problema, la identificación de los actores involucrados y la especificación de los requerimientos funcionales y no funcionales, se ha logrado definir un sistema eficiente y accesible para ciudadanos, empleados y administradores. Además, la representación gráfica mediante diagramas de casos de uso, actividades, secuencia y clases ha facilitado la comprensión de la estructura y flujo del sistema. Con esta documentación, se garantiza que el desarrollo del software esté alineado con los objetivos planteados, asegurando su usabilidad, seguridad y rendimiento en la gestión de incidencias urbanas.

## **Recomendaciones**

- **Validación y prueba temprana:** Se recomienda realizar pruebas tempranas con usuarios reales (ciudadanos, empleados y administradores) para validar la usabilidad y funcionalidad del sistema, asegurando que se ajuste a sus necesidades.
- **Optimización del rendimiento:** Es importante monitorear el desempeño del sistema, especialmente en la carga de reportes en el mapa interactivo y la gestión de notificaciones automáticas, para evitar retrasos o sobrecarga del servidor.
- **Seguridad y protección de datos:** Se debe garantizar la seguridad de la información mediante el cifrado de contraseñas, la protección de datos personales y la implementación de mecanismos contra accesos no autorizados.
- **Escalabilidad del sistema:** Se recomienda diseñar la arquitectura del sistema de manera modular para permitir futuras expansiones, como la integración de nuevas funcionalidades o el aumento en la capacidad de almacenamiento.
- **Mantenimiento y actualizaciones:** Se debe establecer un plan de mantenimiento y actualizaciones periódicas para mejorar la funcionalidad del sistema y corregir posibles errores que puedan surgir tras su implementación.