

**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**

[Introducción 4](#_heading=h.gj1yv1xntg2)

[I. Generalidades de la Empresa 5](#_heading=h.m432xut9ngok)

[1. Nombre de la Empresa: 5](#_heading=h.55y5jutck6r6)

[2. Visión 5](#_heading=h.lt8pdbhvl352)

[3. Misión 5](#_heading=h.dtqg4h6fvfk)

[4. Organigrama 5](#_heading=h.coiirm5m1g66)

[II. Visionamiento de la Empresa: 6](#_heading=h.ht10a6hngypz)

[1. Descripción del Problema 6](#_heading=h.qft78aruq8so)

[2. Objetivos de Negocios 6](#_heading=h.ht8bgusqsdk2)

[3. Objetivos de Diseño 6](#_heading=h.xm0b0c7fmd80)

[4. Alcance del proyecto 7](#_heading=h.yrhnealcn9n)

[5. Viabilidad del sistema 7](#_heading=h.oeeuljl83bad)

[III. Análisis de Procesos 10](#_heading=h.nzc0412v9ufc)

[a) Diagrama del Proceso Actual - Diagrama de Actividades 10](#_heading=h.g1unk8lcdhw8)

[b) Diagrama del Proceso Propuesto - Diagrama de actividades Inicial 11](#_heading=h.gevnsa7dloic)

[IV. Especificación de Requerimientos de Software 12](#_heading=h.jzwqy39g5c6z)

[a) Cuadro de Requerimientos Funcionales Inicial 12](#_heading=h.fkmykcbp1tk)

[b) Cuadro de Requerimientos No Funcionales 13](#_heading=h.ql42pird3a93)

[c) Cuadro de Requerimientos Funcionales Final 14](#_heading=h.csndj24pdi0z)

[d) Reglas de Negocio 16](#_heading=h.m5dk4u3eujd8)

[V. Fases de Desarrollo 17](#_heading=h.rm8tvcjz3yw5)

[1. Perfiles de Usuario 17](#_heading=h.foch0vj8luvr)

[2. Modelo Conceptual 18](#_heading=h.gmjl1vdcb3va)

[a) Diagrama de Paquetes 18](#_heading=h.i8nk23gw3pr9)

[b) Diagramas de Casos de Uso 21](#_heading=h.amfrge9drnvv)

[c) Escenarios de casos de uso (Narrativo) 22](#_heading=h.9tixiil1mocb)

[3. Modelo Lógico 30](#_heading=h.18yge0ehi6z7)

[a) Análisis de Objetos 30](#_heading=h.fqoi7tdxe9r8)

[b) Diagrama de Actividades con Objetos 32](#_heading=h.jyuydr3q6w5t)

[c) Diagrama de Secuencia 38](#_heading=h.tqsay65m3ywd)

[d) Diagrama de Clases 39](#_heading=h.vl3gea45mdlt)

[**Conclusiones 40**](#_heading=h.5k1y4oywf5d8)

[**Recomendaciones 40**](#_heading=h.xh0m9u92dcz5)

**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**

## Nombre de la Empresa:

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

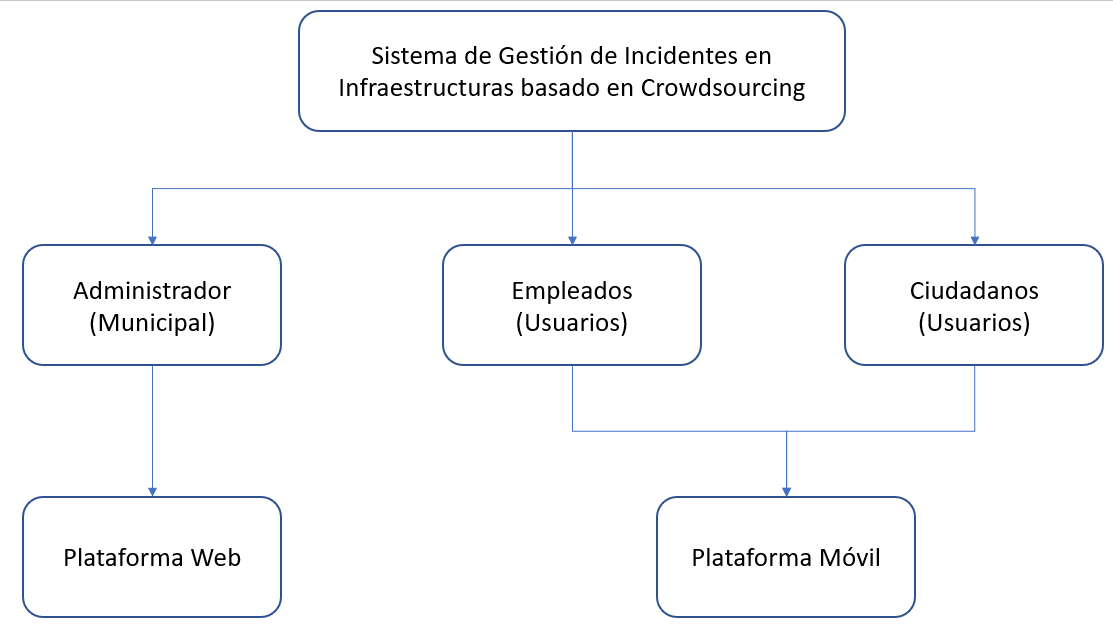
## Visión

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.

## Misión

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

## Organigrama

****

# **II. Visionamiento de la Empresa:**

## 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.

## Objetivos de Negocios

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

## 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.

## 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.

## Viabilidad del sistema

* 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. **Viabilidad Operativa**
* **Beneficios para los Usuarios**:

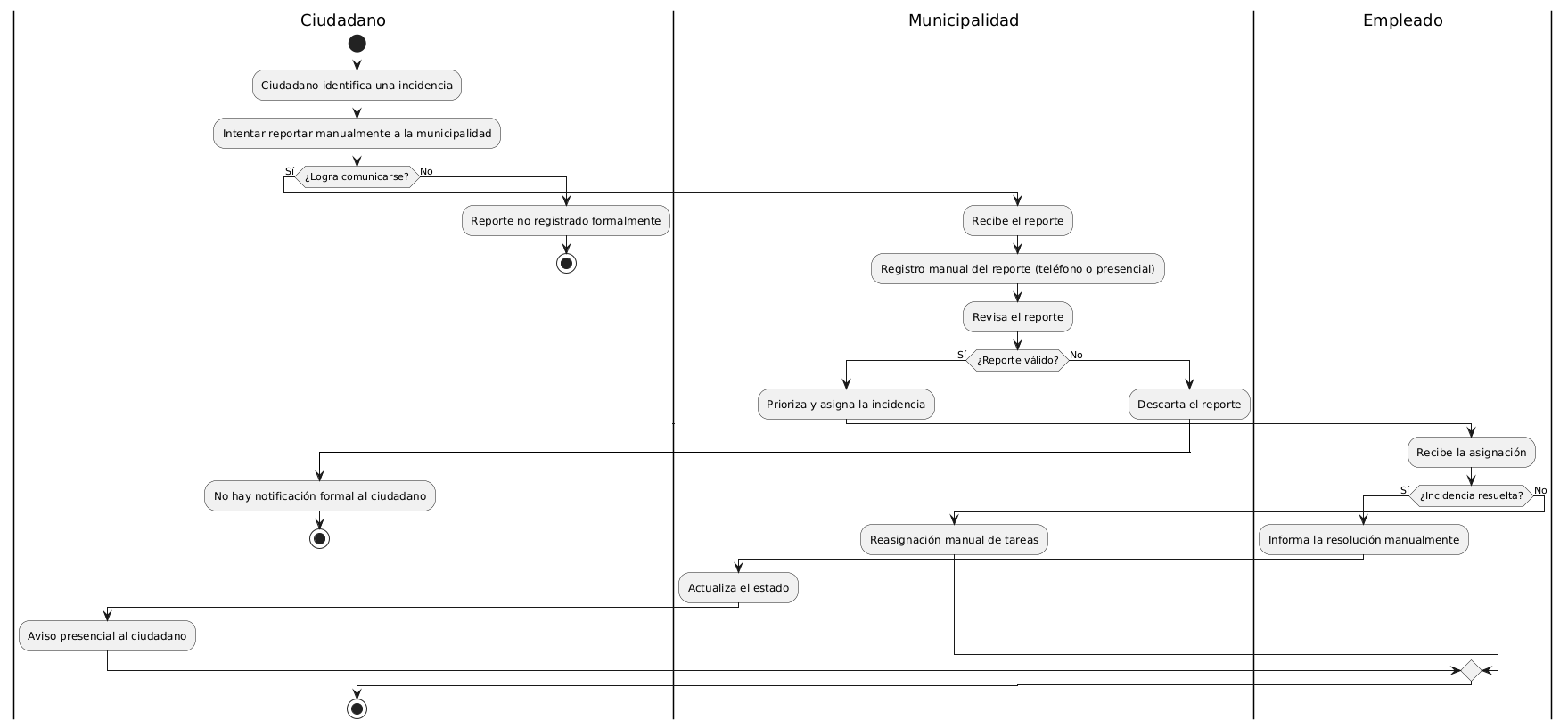
El proyecto ofrecerá una plataforma segura y eficiente para generar los reportes de las incidencias y sus respectivas gestiones gracias al administrador en la versión Web, y con una API backend, los usuarios podrán acceder de manera segura al sistema móvil, gracias al API de Google Maps podrán encontrar 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. **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.
3. **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.

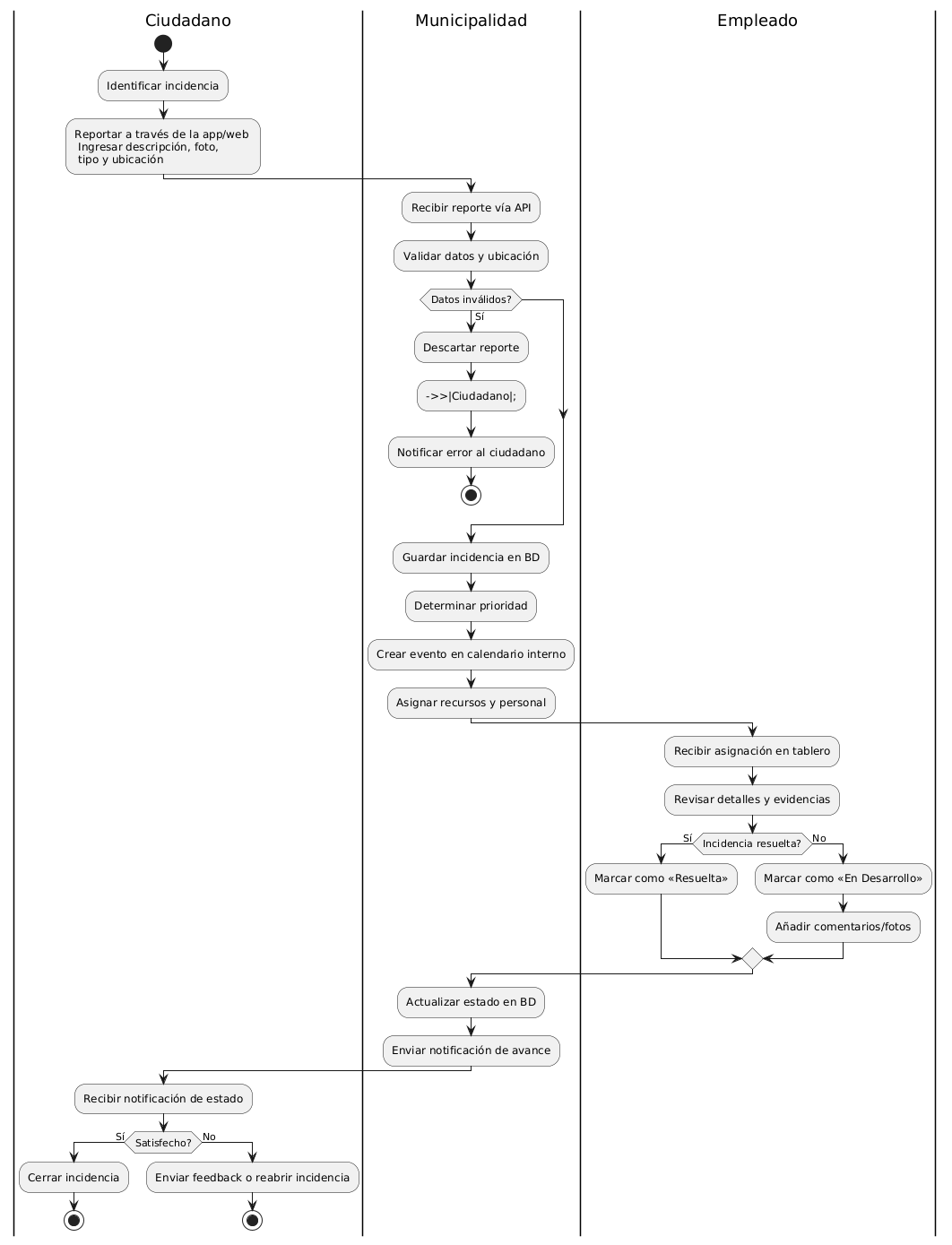
# III. [Análisis de Procesos](https://docs.google.com/document/d/1kzyTY_VoV_pDJ15dD-avtakwMccfG6fT/edit#heading=h.30j0zll)

## Diagrama del Proceso Actual - Diagrama de Actividades



*Este diagrama muestra el flujo actual de gestión de incidencias, destacando las interacciones manuales entre el ciudadano y la municipalidad, así como los puntos críticos donde el reporte puede no ser registrado o gestionado de manera efectiva.*

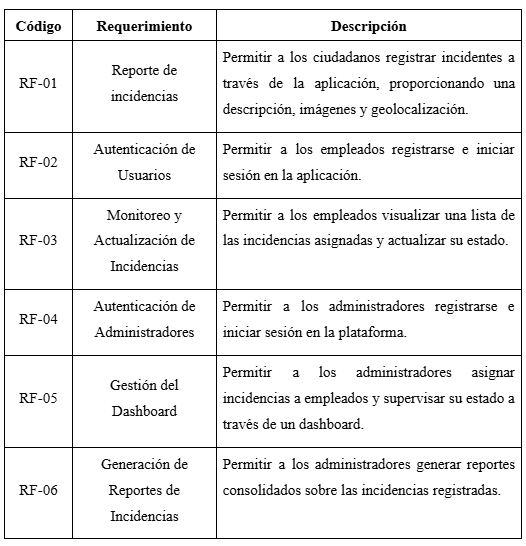
## Diagrama del Proceso Propuesto - Diagrama de actividades Inicial



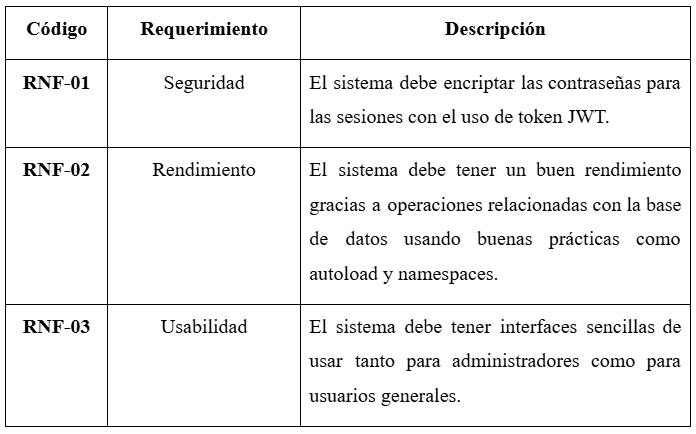
*Diagrama 01: Este diagrama proporciona una vista clara del flujo del sistema, destacando las responsabilidades y las interacciones entre los usuarios y el sistema. Esto permite identificar puntos clave de interacción y garantizar que cada funcionalidad esté correctamente implementada en el sistema.*

# IV. Especificación de Requerimientos de Software

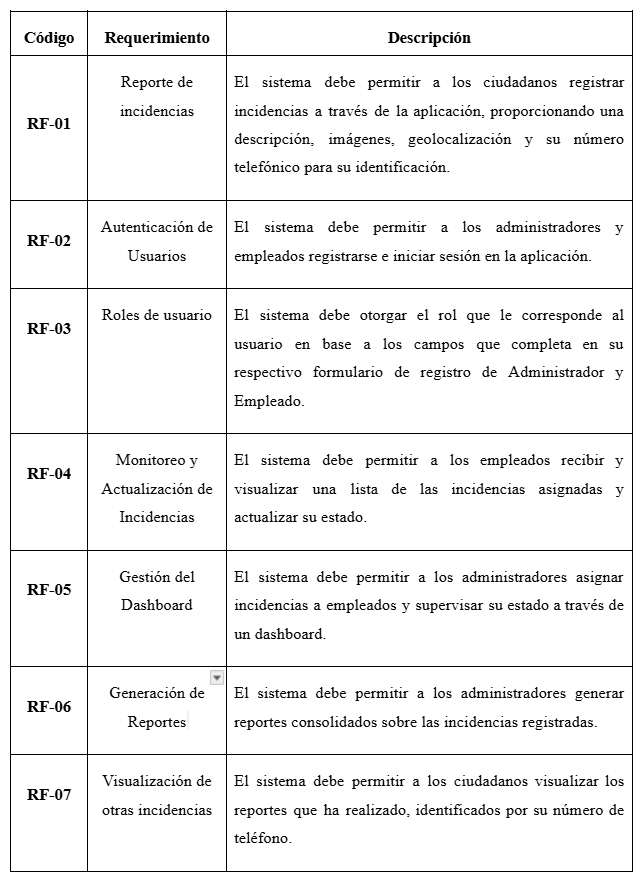
## Cuadro de Requerimientos Funcionales Inicial



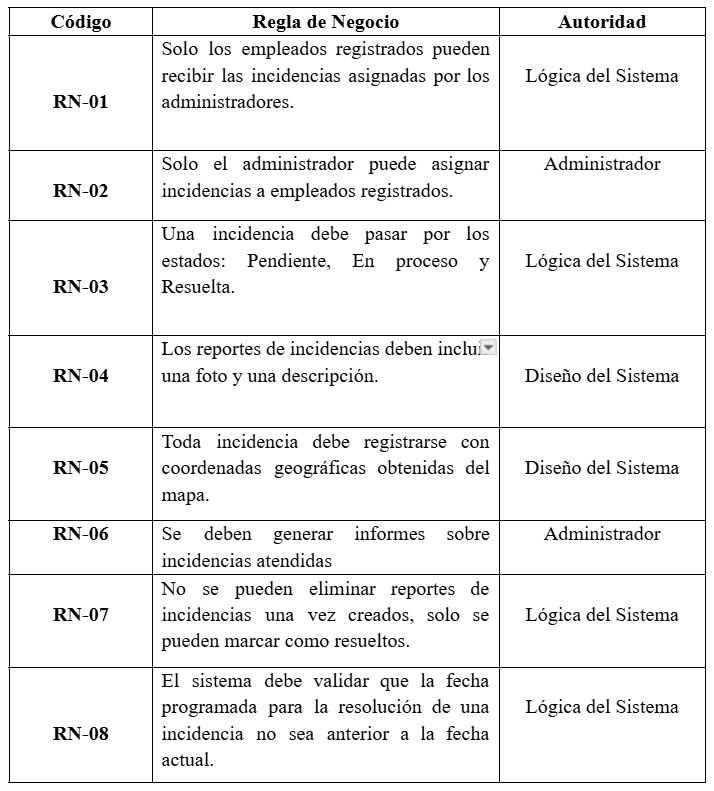
## Cuadro de Requerimientos No Funcionales



## Cuadro de Requerimientos Funcionales Final



## Reglas de Negocio



# **V. Fases de Desarrollo**

## 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 con su número de teléfono, el cual servirá para su identificación y relación con su reporte correspondiente, 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. En otro apartado, el ciudadano puede ver en un mapa los reportes que ha mandado, siendo su identificador el número telefónico.

* **Empleado (Usuario)**

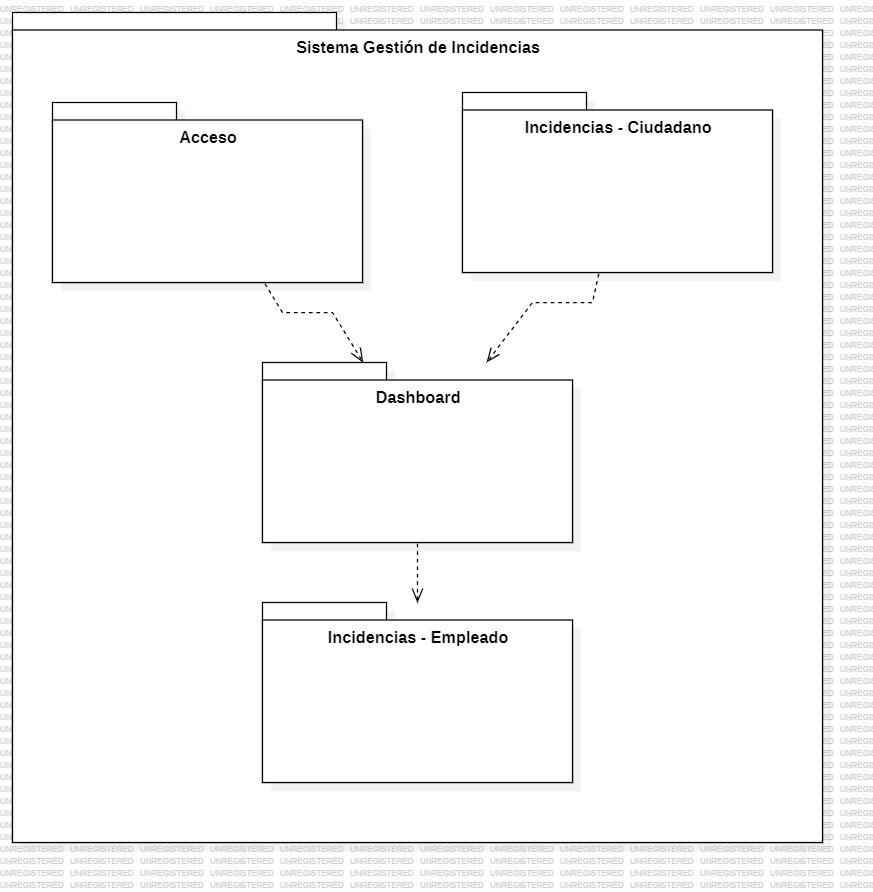
El empleado es un individuo que se ha registrado en la aplicación móvil, encargado de gestionar los problemas reportados por los ciudadanos. Para ingresar al sistema, necesita registrarse e iniciar sesión. Después de autenticarse, puede recibir alertas cuando le asignan una nueva tarea. Su labor implica 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. Así mismo, es responsable de la seguridad y mantenimiento del sistema, asegurándose de que solo los usuarios autorizados puedan acceder a las diversas funcionalidades.

## Modelo Conceptual

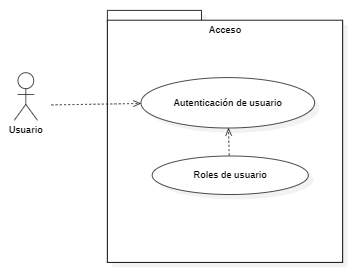
### Diagrama de Paquetes



**Fuente:** Elaboración Propia

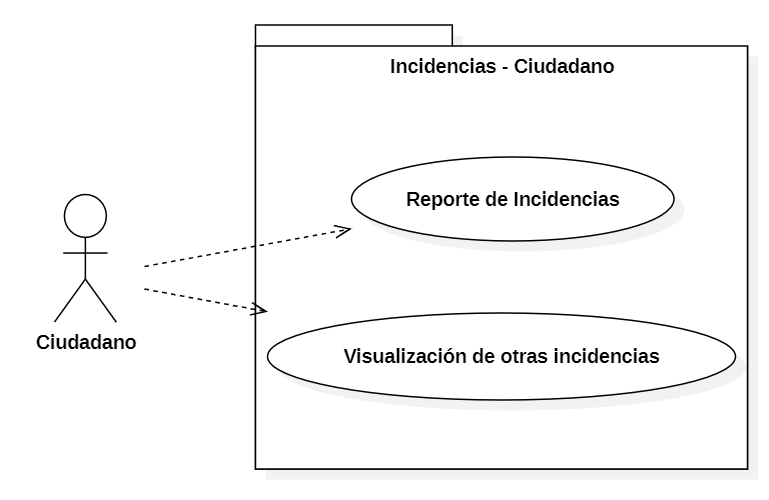
*El diagrama muestra la estructura del proyecto con los subpaquetes principales: Acceso, Incidencias para Ciudadanos y Empleado, y el Dashboard junto a las relaciones correspondientes.*

**Paquete Acceso**



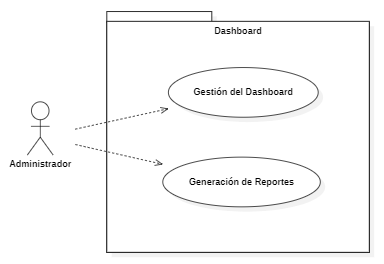
**Fuente:** Elaboración Propia

**Paquete Incidencias - Ciudadano**



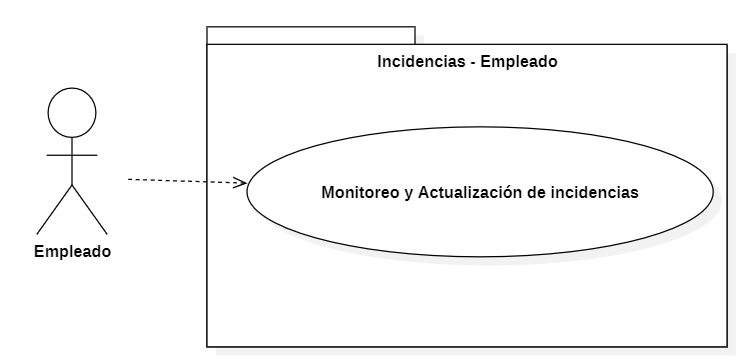
**Fuente:** Elaboración Propia

**Paquete Dashboard**



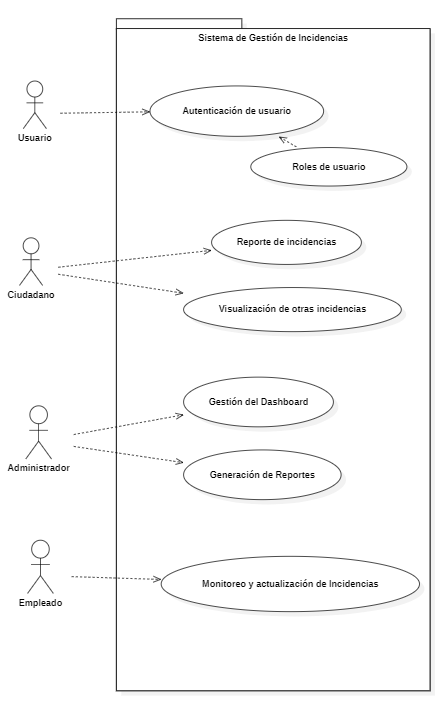
**Fuente:** Elaboración Propia

**Paquete Incidencias - Empleados**



**Fuente:** Elaboración Propia

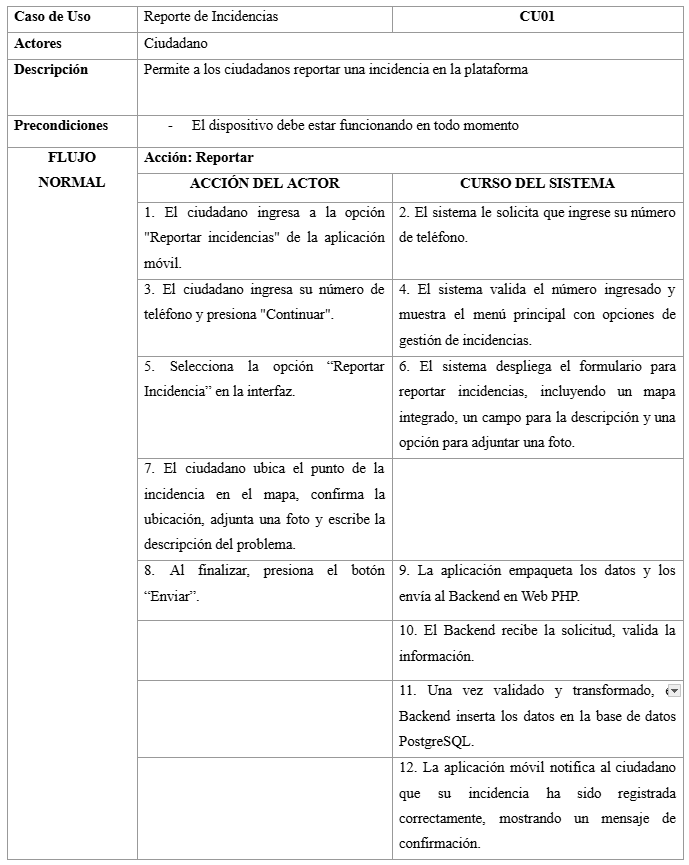
### Diagramas de Casos de Uso



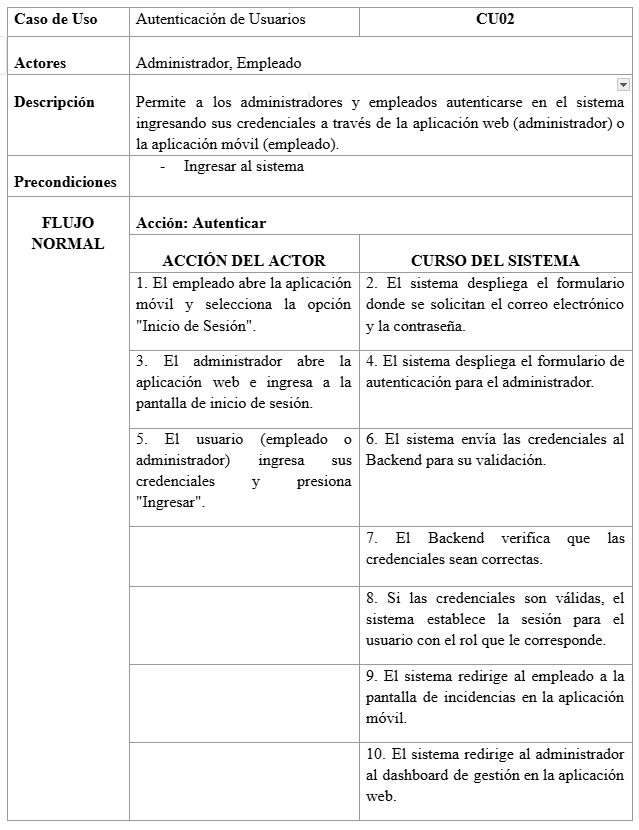
**Fuente:** Elaboración Propia

### Escenarios de casos de uso (Narrativo)

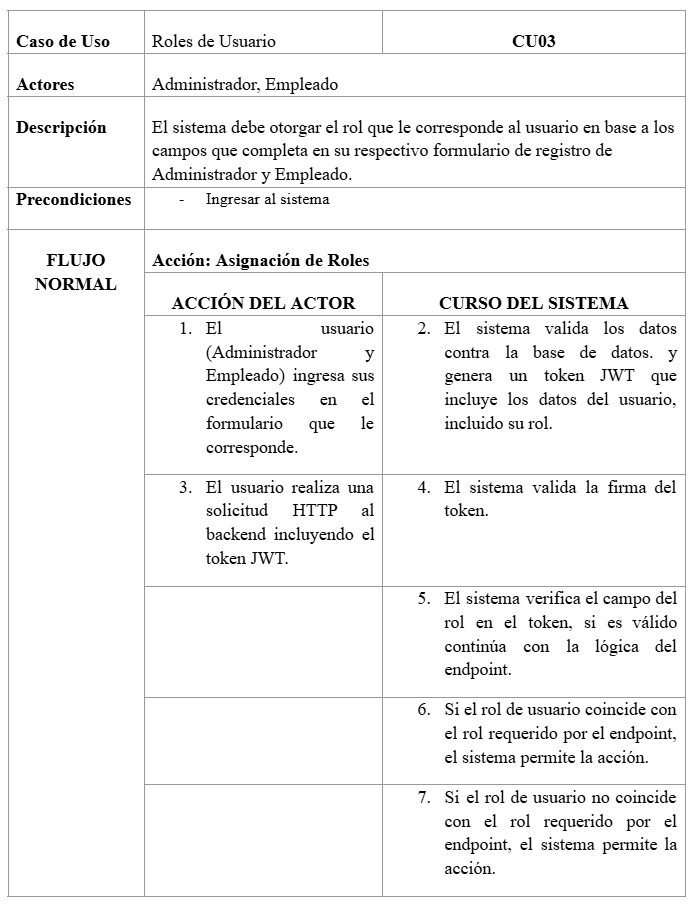
**Narrativa de CU01 - Reporte de Incidencias**



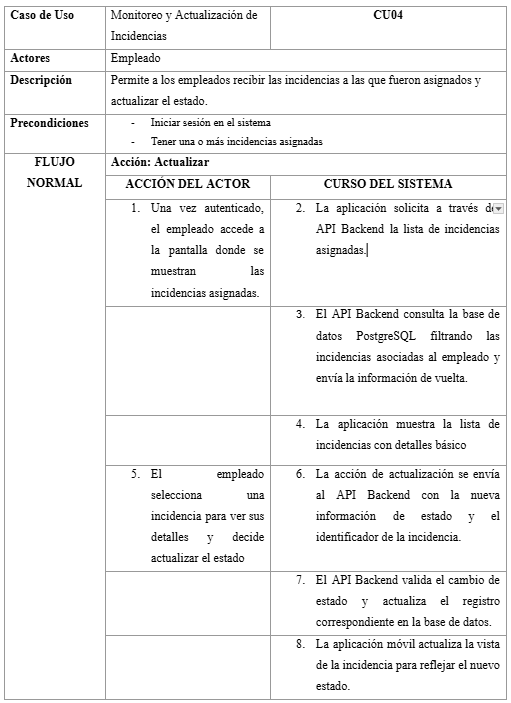
**Narrativa de CU02 - Autenticación de Usuarios**



**Narrativa de CU03 - Roles de usuario**



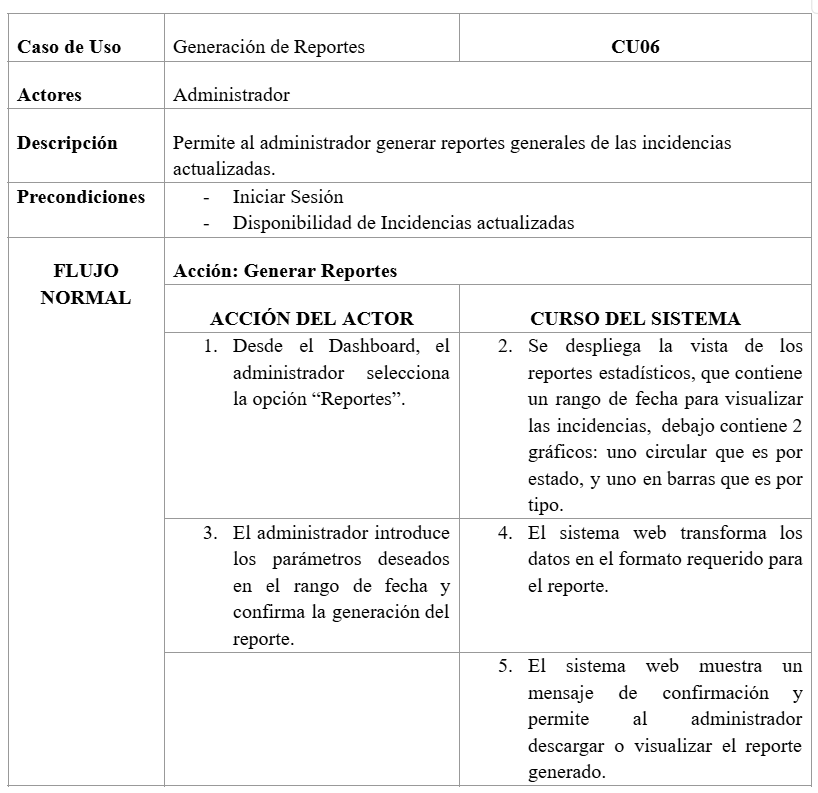
**Narrativa de CU04 - Monitoreo y Actualización de Incidencias**



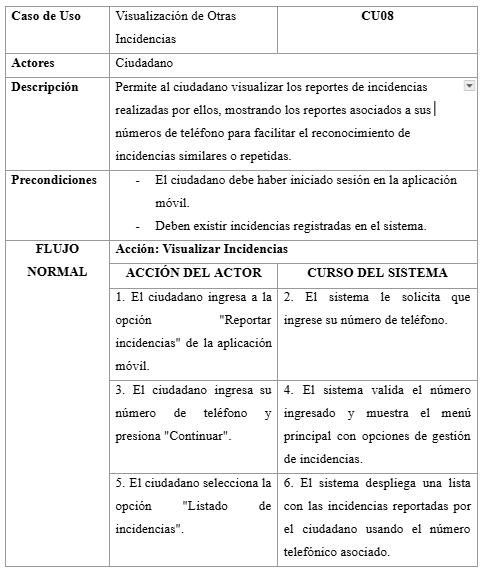
**Narrativa de CU05 - Gestión del Dashboard**



**Narrativa de CU06 - Generación de Reportes**



**Narrativa de CU07 - Visualización de Otras Incidencias**

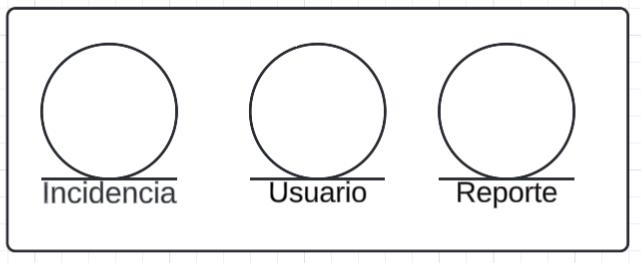


## Modelo Lógico

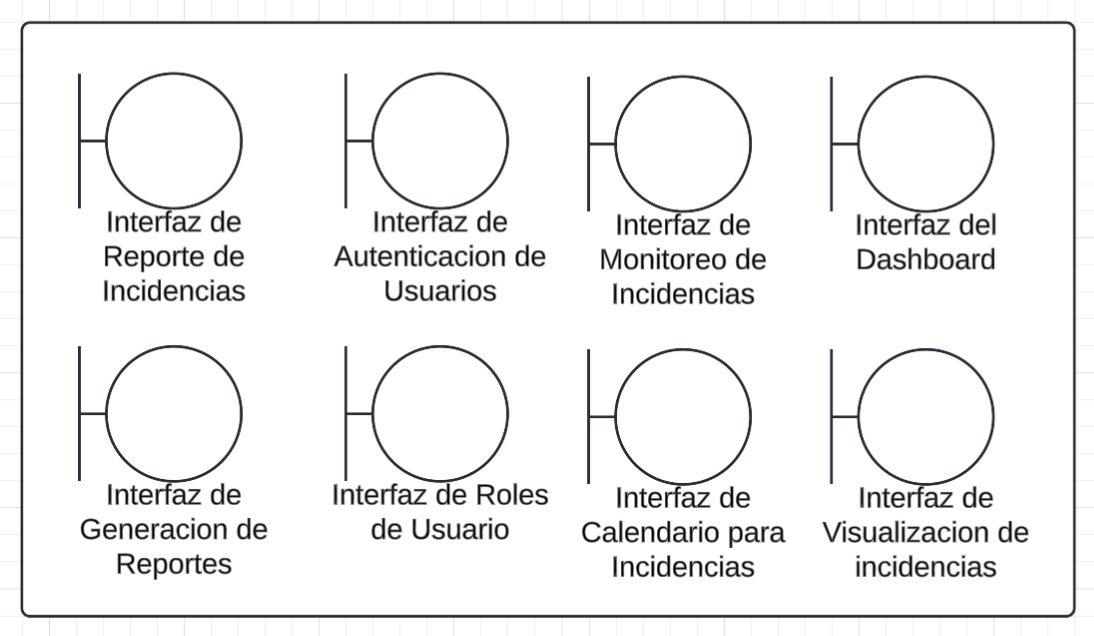
### Análisis de Objetos

El análisis de objetos permite reconocer y definir las clases principales, sus atributos y las relaciones que integran el sistema. Este proceso es fundamental para el diseño orientado a objetos, ya que facilita la estructuración del software en componentes coherentes y reutilizables.

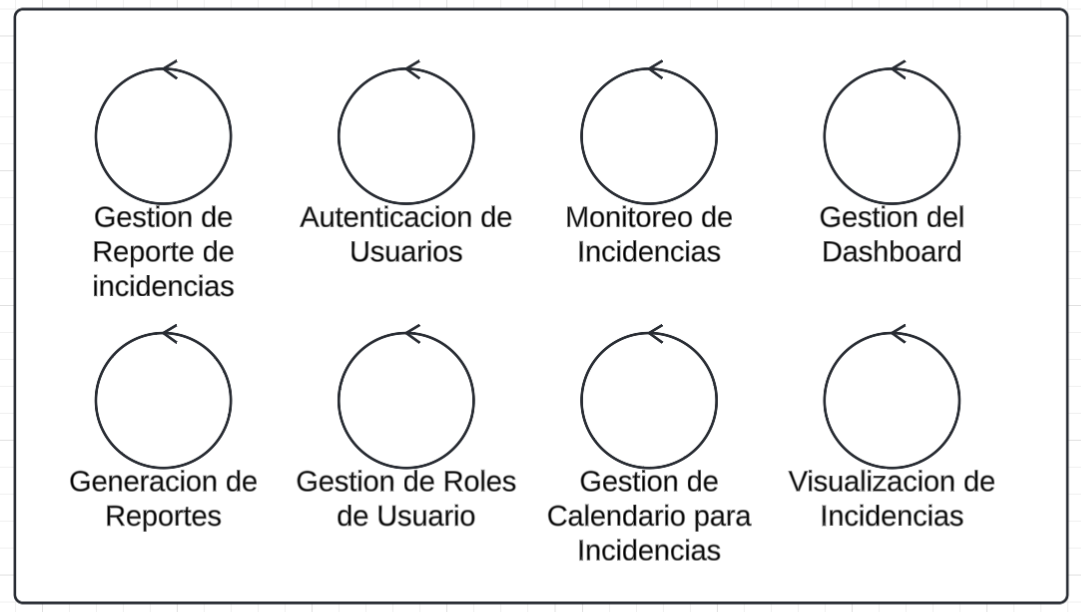
**Objetos Entidad**

****

**Objetos Frontera**

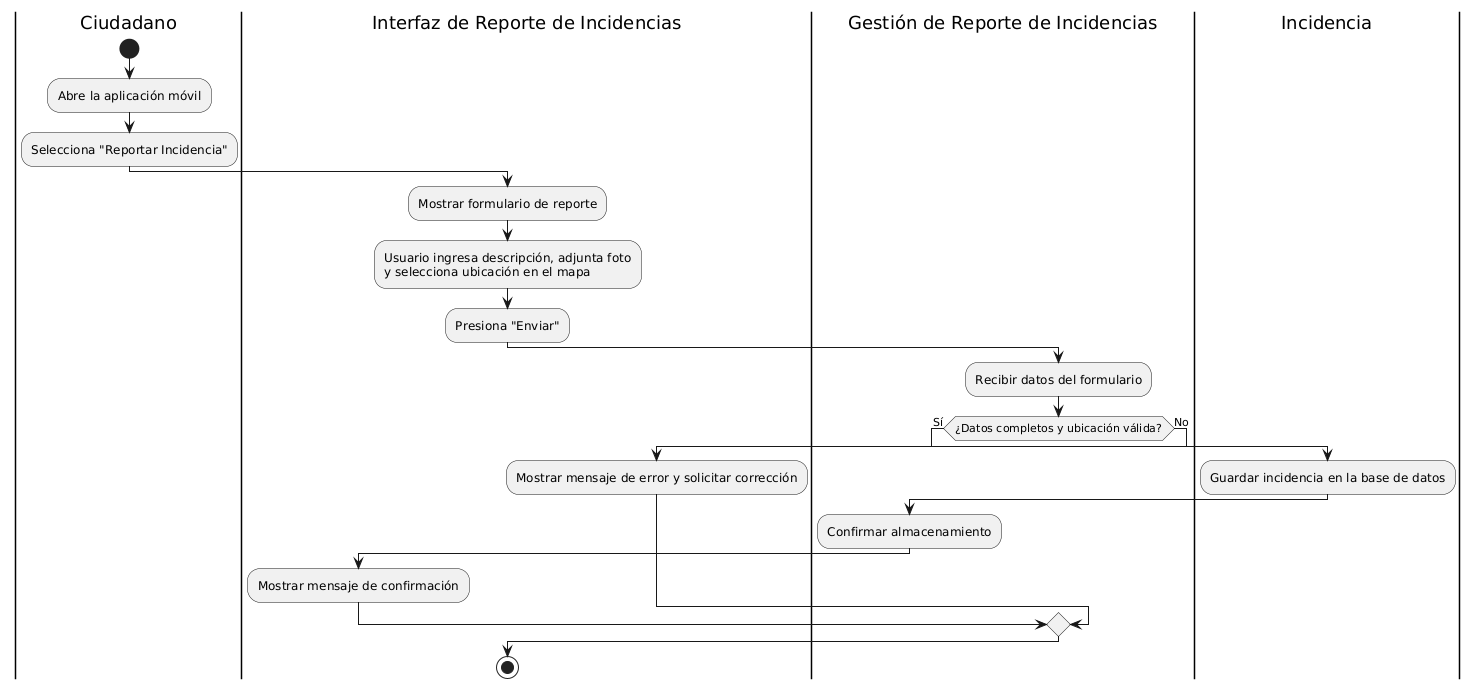
****

**Objetos Control**

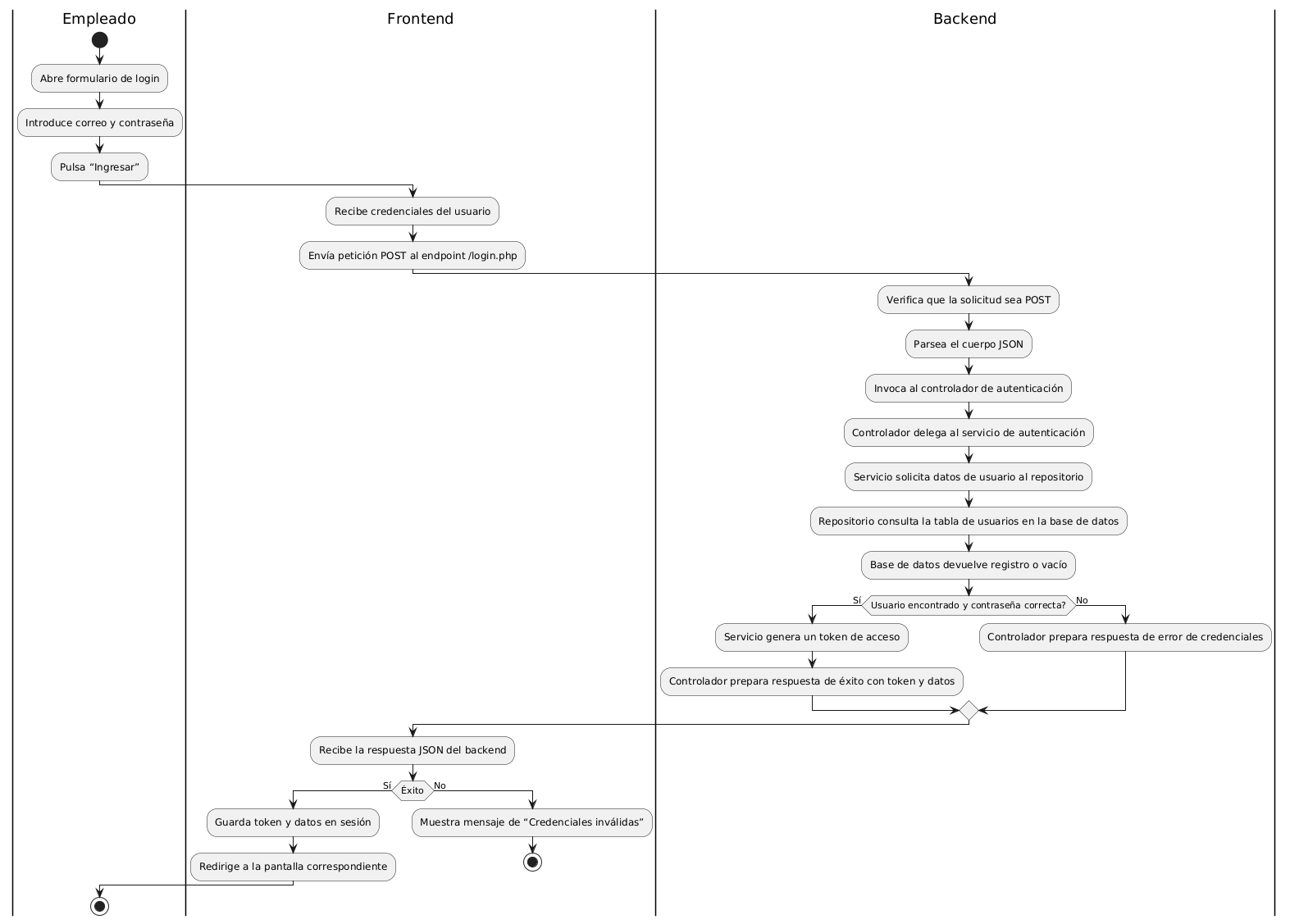
****

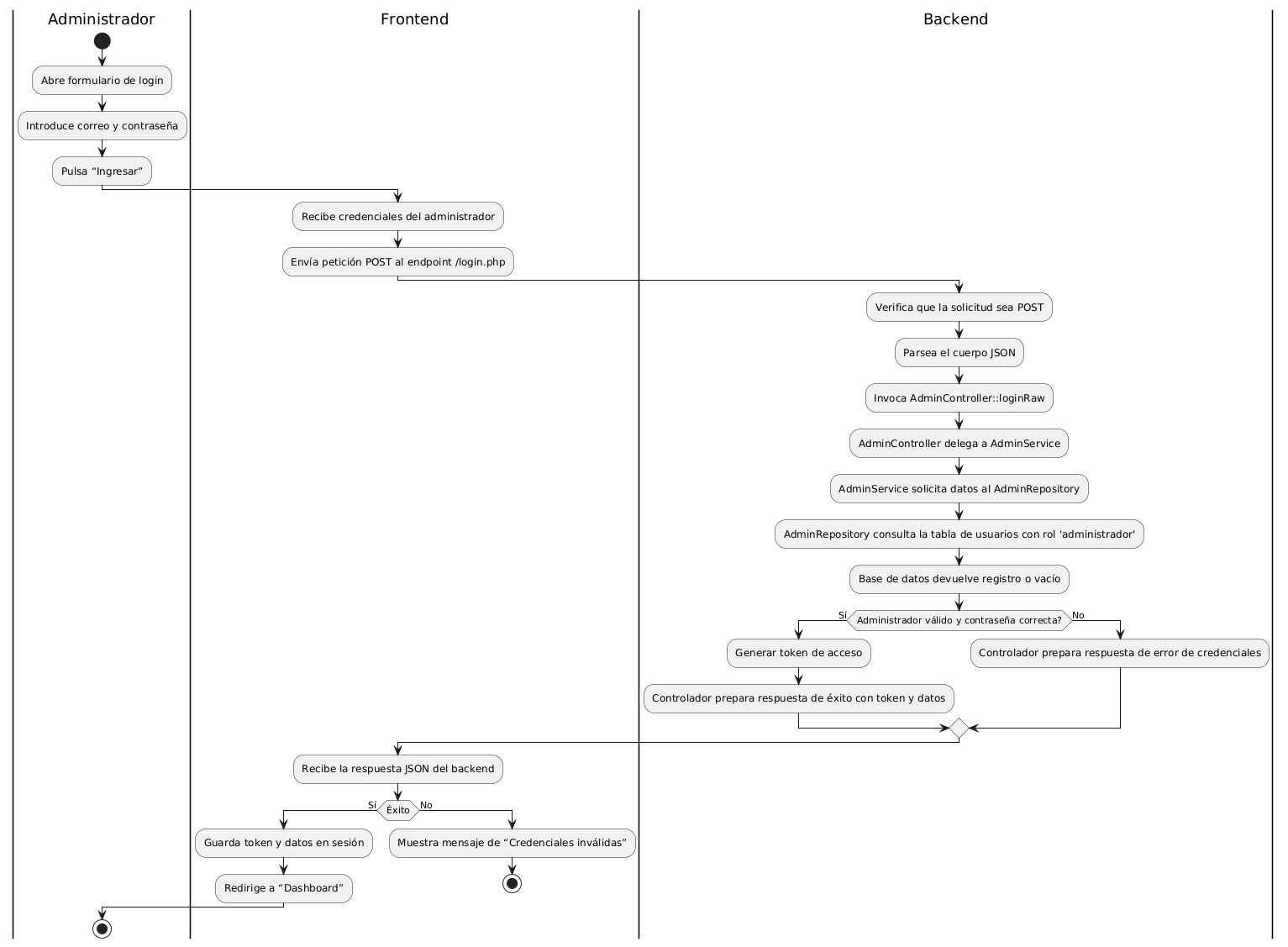
### Diagrama de Actividades con Objetos

**RF-01: Reporte de Incidencias**

****

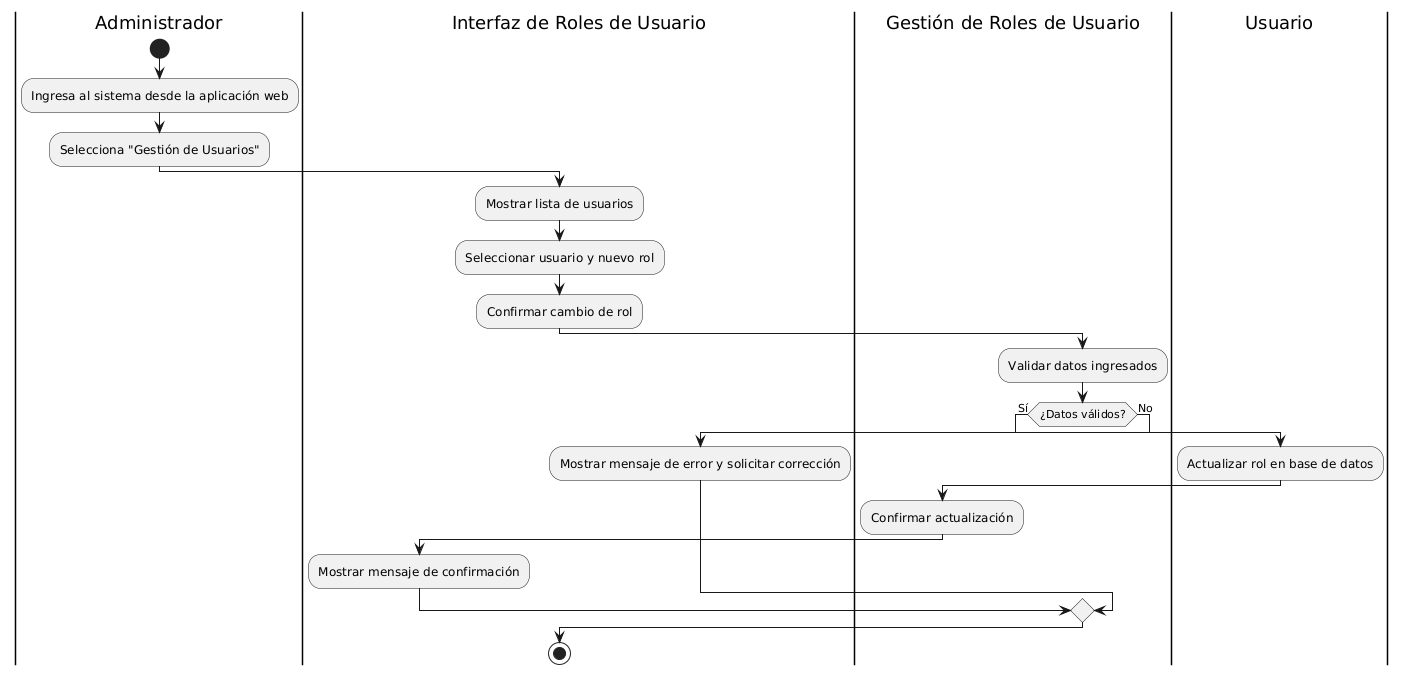
**RF-02: Autenticación de Usuarios**

****

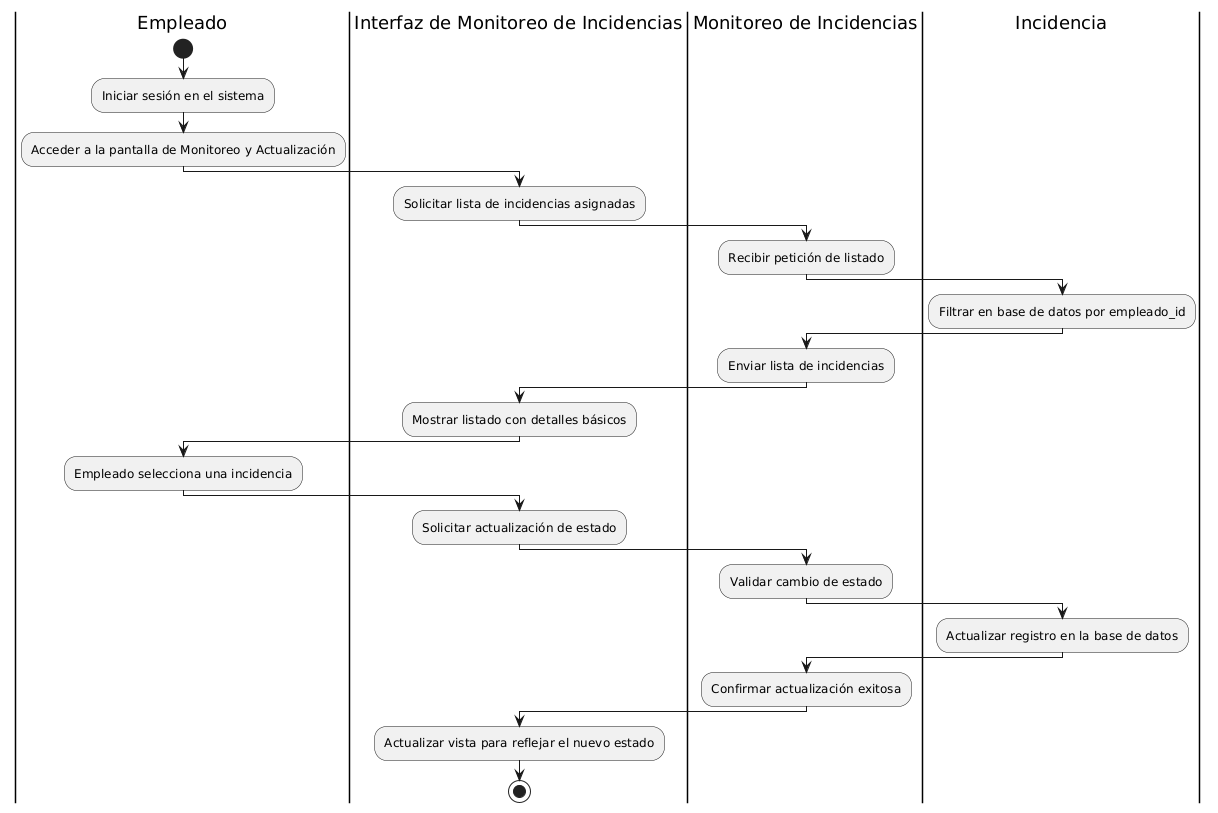
****

**Link** [Diagramas de Actividades](https://drive.google.com/drive/folders/1x5RISF7hs2QMxa20DCb71--Fm0Zh6R5y?usp=sharing)

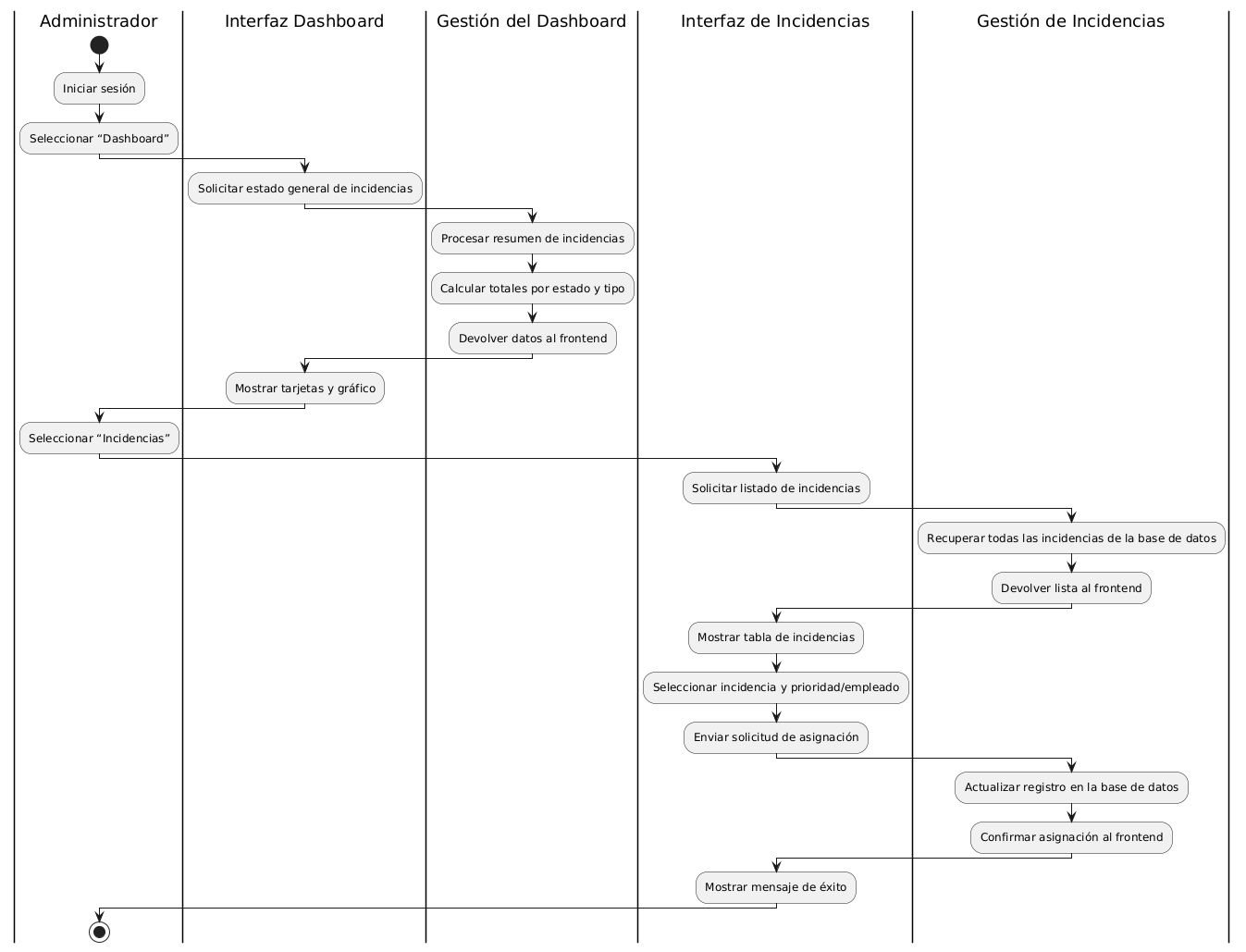
**RF-03: Roles de usuario**

****

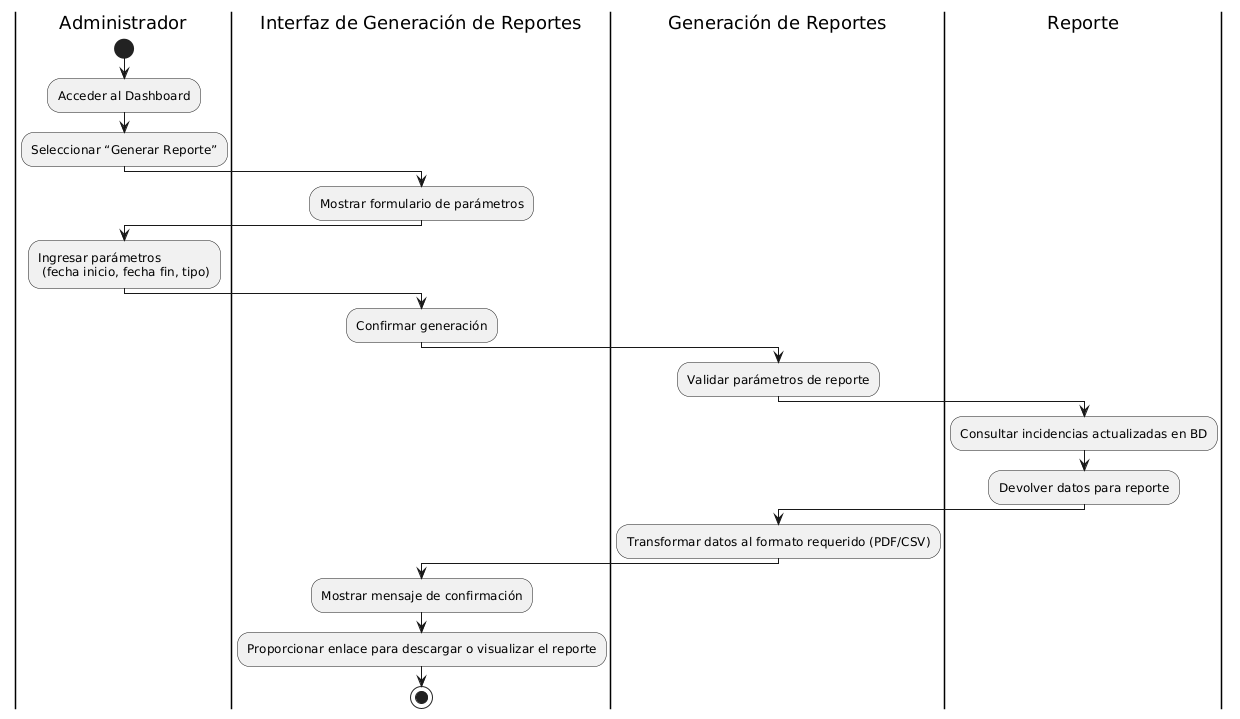
**RF-04: Monitoreo y Actualización de Incidencias**

****

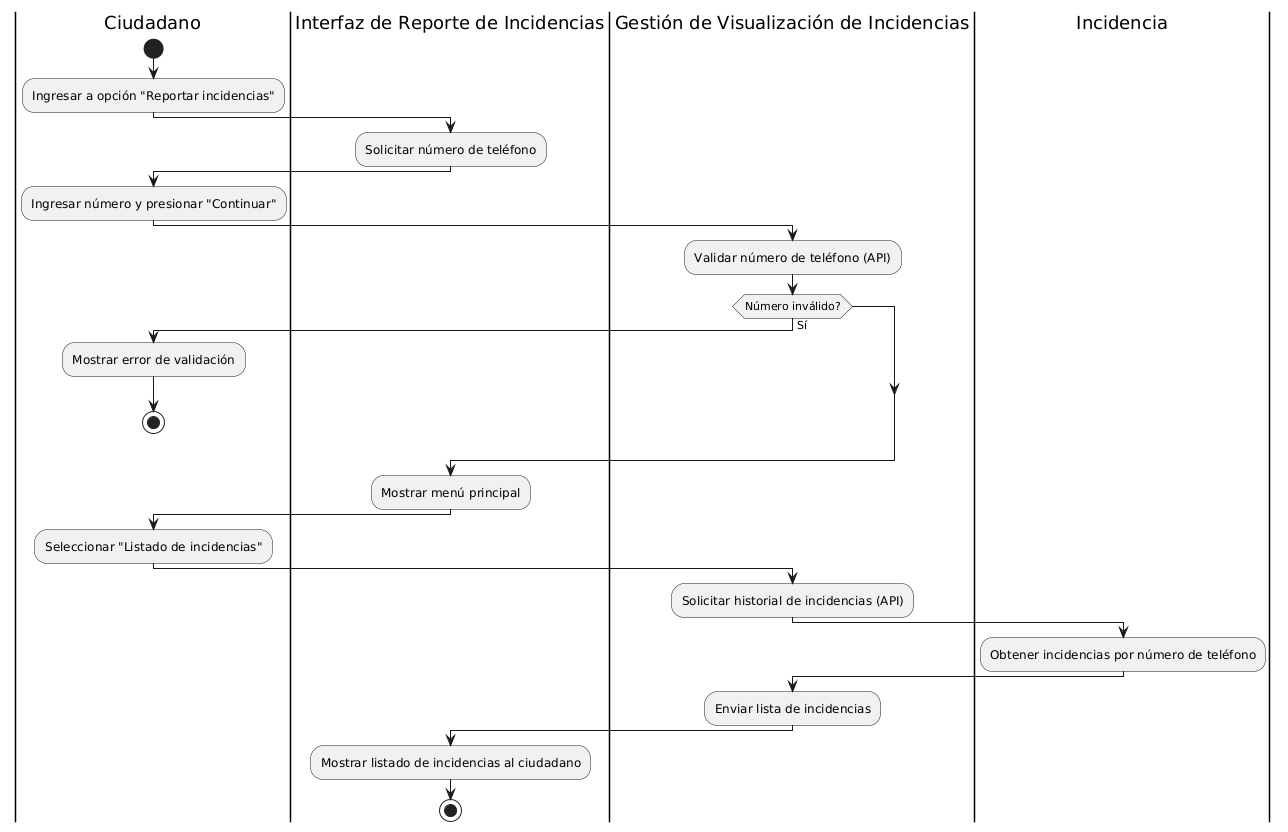
**RF-05: Gestión del Dashboard**

****

**RF-06: Generación de Reportes de Incidencias**



**RF-07: Visualización de otras incidencias**

~~~~

### Diagrama de Secuencia

**Diagrama de secuencia: Reporte de incidencias**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

**Diagrama de secuencia: Autenticación de Usuarios**

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

**Diagrama de secuencia: Roles de usuario**

Tabla

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Diagrama de secuencia: Monitoreo y Actualización de Incidencias**

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

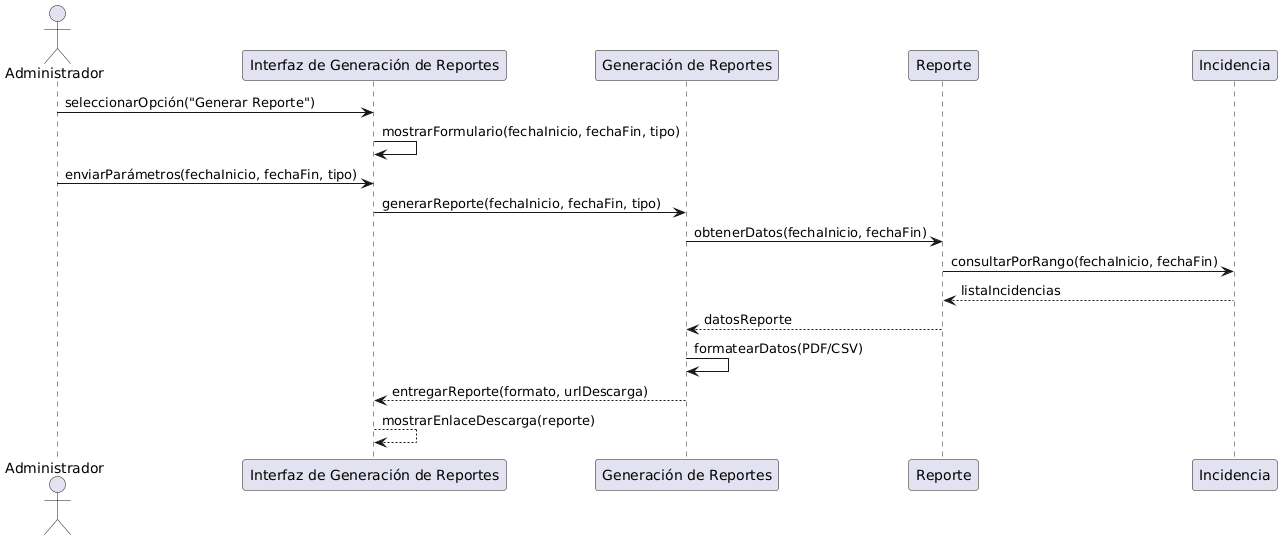
El contenido generado por IA puede ser incorrecto.

**Diagrama de secuencia: Gestión del Dashboard**

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

**Diagrama de secuencia: Generación de Reportes de Incidencias**

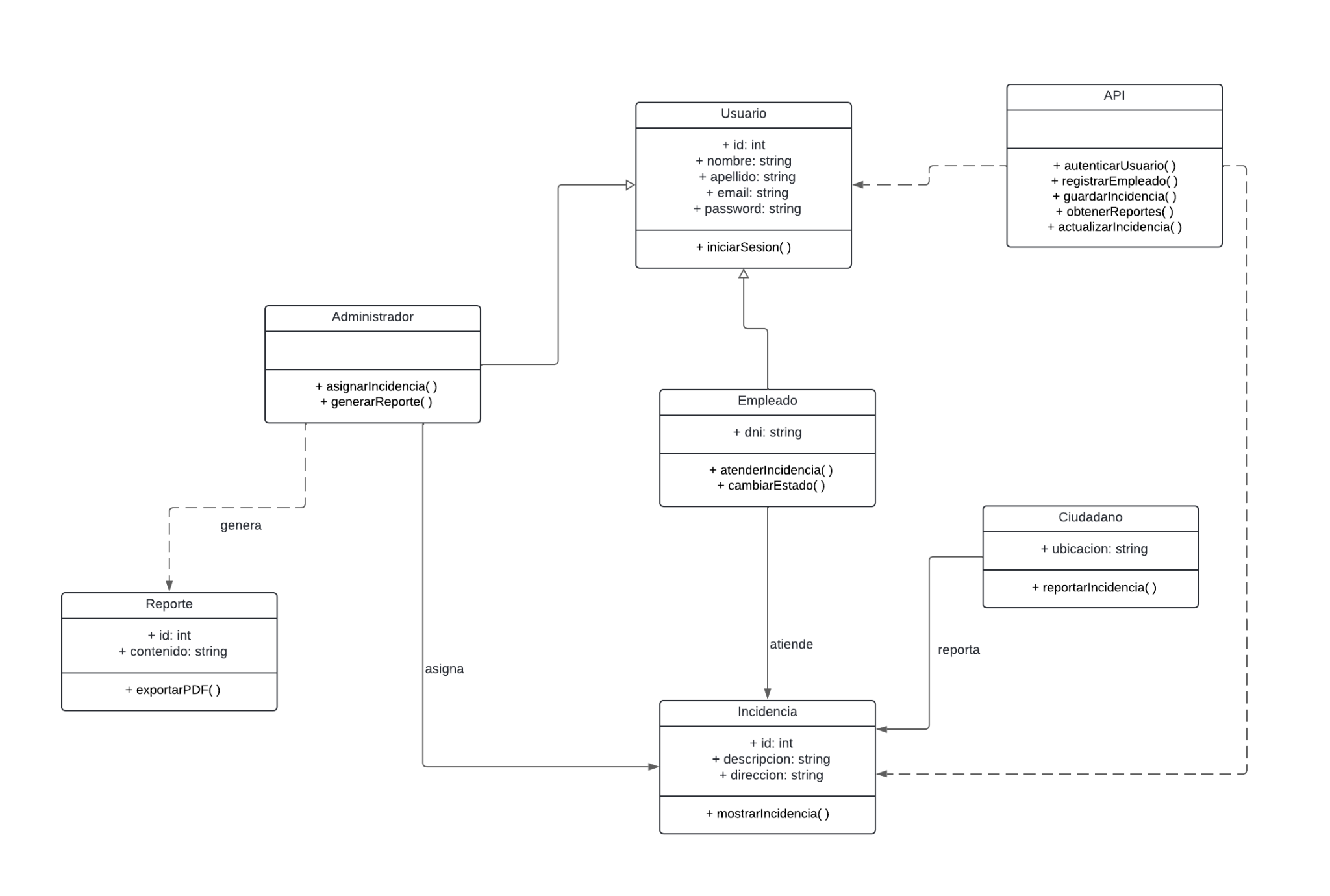


**Diagrama de secuencia: Visualización de otras incidencias**

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

### Diagrama de Clases



El diagrama de clases UML presenta la estructura del sistema, donde la clase Usuario actúa como base para Administrador y Empleado a través de herencia. La clase Incidencia se conecta directamente con Ciudadano, Empleado y Administrador, ya que el ciudadano la reporta, el empleado la atiende y el administrador la asigna. Además, el Administrador genera reportes, mientras que la clase API interactúa con Usuario e Incidencia mediante dependencias para ejecutar operaciones.

# **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.