
# UNIVERSIDAD PRIVADA DE TACNA

## FACULTAD DE INGENIERÍA

### Escuela Profesional de Ingeniería de Sistemas

# Sistema de alerta basado en tecnología LoRaWAN para optimizar la respuesta de emergencias por violencia contra la mujer e integrantes del grupo familiar en Perú, 2025

Curso: Construcción de Software I

Docente: Ing. Alberto Johnatan Flor Rodríguez

Integrantes:

Daleska Nicolle Fernandez Villanueva (2021070308)

**Tacna – Perú**

**2025**

# Sistema de alerta basado en tecnología LoRaWAN para optimizar la respuesta de emergencias por violencia contra la mujer e integrantes del grupo familiar en Perú, 2025

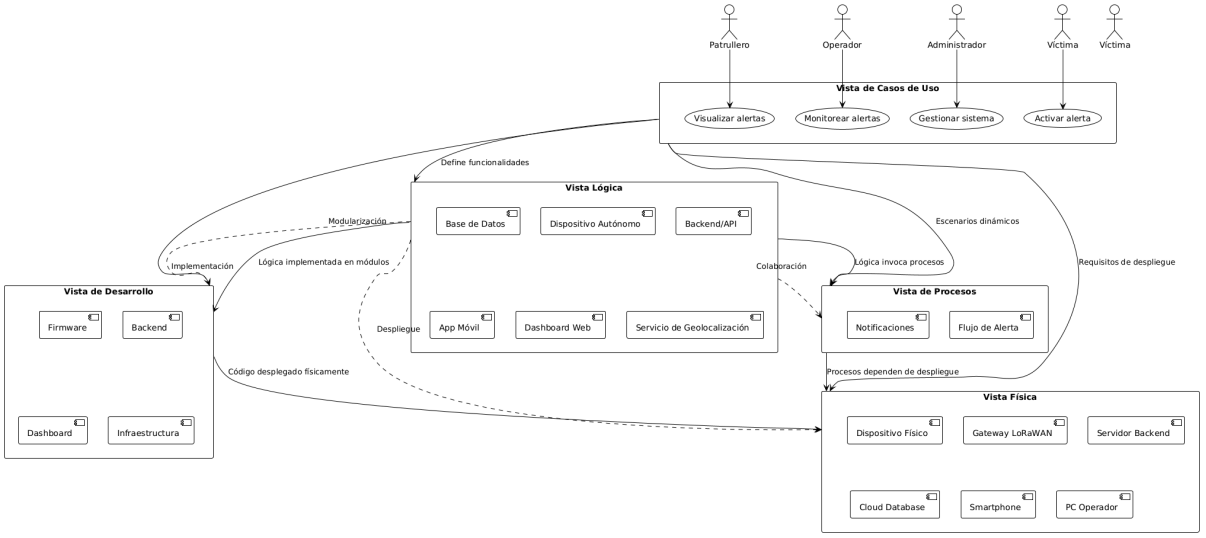
**Versión 1.0**

| CONTROL DE VERSIONES | | | | | |
|---|---|---|---|---|---|
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | DFV | DFV | DFV | 04/09/2025 | Versión Original |

ÍNDICE

# 1. Introducción

## 1.1. Propósito (Diagrama 4+1)




## 1.2. Alcance

El sistema se encargará de las siguientes funcionalidades y módulos principales:

Gestión del Dispositivo Autónomo de Alerta Personal:

- Implementar el firmware optimizado en C++ para la gestión eficiente de energía mediante modos de deep sleep, activación del GPS solo durante eventos de alerta y construcción de payload binario (Device ID, coordenadas GPS, nivel de batería, timestamp) transmitido vía LoRaWAN.
- Controlar LEDs RGB en el dispositivo para indicar estados de alerta: rojo durante el envío, verde al confirmar recepción de la alerta y ámbar para batería baja, proporcionando retroalimentación visual inmediata.
- Garantizar autonomía energética superior a 30 días mediante optimización de consumo y control de activación de módulos.

Conectividad y Reenvío de Datos LoRaWAN:

- Utilizar gateways LoRaWAN (RAK7248) para recepción de datos en distancias de 3–10 km según entorno urbano o rural.

- Emplear The Things Stack para la deduplicación, desencriptación (AES-128) y reenvío seguro de paquetes hacia el backend.
- Configurar webhooks para entrega de datos en tiempo real al backend y a la plataforma de monitoreo, asegurando integración sin necesidad de conectividad móvil convencional.

Procesamiento Central de Alertas:

- Desarrollar backend en ASP.NET Core que reciba y procese los datos enviados por TTS, incluyendo decodificación de payload y consulta de perfiles de víctimas en Firestore.
- Implementar lógica de asignación automática de alertas a la patrulla o unidad más cercana mediante cálculo de distancias Haversine.
- Controlar el ciclo de vida de las alertas (PENDIENTE, ASIGNADA, EN RUTA, EN SITIO, RESUELTA) y notificar en tiempo real a las aplicaciones móviles y dashboard web mediante SignalR.

Almacenamiento y Gestión de Datos (Firestore):

- Establecer base de datos Firestore en tiempo real para perfiles de víctimas, historial de alertas, ubicación de patrullas (actualización cada 15 s) y registros de operaciones.
- Facilitar sincronización inmediata de información entre dashboard web y aplicaciones móviles, asegurando que operadores y patrullas reciban datos actualizados en tiempo real.

Interfaz y Gestión para Unidades de Respuesta (Aplicación Móvil/Tablet):

- Desarrollar aplicación móvil Flutter para patrullas con visualización de alertas en mapas interactivos (API Google Maps), mostrando ubicación de la víctima, patrulla y otras unidades cercanas.
- Permitir actualización del estado de la alerta (aceptar, en ruta, resuelto) y envío automático de ubicación GPS de la patrulla cada 15 segundos.
- Notificar instantáneamente nuevas alertas, confirmaciones y cambios de estado en tiempo real.

Validación y Pruebas del Sistema:

- Realizar pruebas de laboratorio y simulaciones de campo para evaluar latencia de extremo a extremo (≤ 20 s), precisión de geolocalización, cobertura efectiva (3–10 km), eficiencia de consumo energético y efectividad del despacho automático.

- Generar reportes de desempeño y Open Data con métricas de alertas, tiempos de respuesta y cobertura, excluyendo datos sensibles de las víctimas, para análisis operativo y social.

## 2. Objetivos y restricciones arquitectónicas

### 2.1. Objetivos

Objetivo general

Diseñar e implementar un sistema de alertas basado en tecnología LoRaWAN para las respuestas de emergencia en casos de violencia contra la mujer e integrantes del grupo familiar en Perú, asegurando confiabilidad, cobertura y eficiencia en la transmisión de alertas.

Objetivos Específicos

- Diseñar e implementar el firmware del dispositivo autónomo para gestionar la activación de la alerta, la obtención de coordenadas GPS y la transmisión de datos vía LoRaWAN, priorizando la eficiencia energética.
- Desarrollar el hardware del dispositivo autónomo que integre un microcontrolador, un módulo LoRaWAN, un módulo GPS, una batería de larga duración y un indicador LED de confirmación de envío.
- Configurar la red LoRaWAN (The Things Stack) y sus integraciones para asegurar la recepción y el reenvío eficiente de los paquetes de datos del dispositivo al Servidor Central de Monitoreo.
- Implementar el Servidor Central de Monitoreo (basado en la nube) capaz de recibir, decodificar y procesar las alertas, identificar a la víctima y la unidad de respuesta más cercana, y enviar la alerta digital automatizada.

- Desarrollar la aplicación cliente para las unidades de respuesta (PNP/Serenazgo) que permita la visualización en tiempo real de las alertas, la ubicación de la víctima, y la gestión del estado de la emergencia.
- Realizar pruebas de rendimiento del sistema para validar que la latencia de transmisión de la alerta es de 5 segundos o menos y que la confiabilidad de transmisión supera el 98% en diversas condiciones de entorno.
- Evaluar la autonomía energética del dispositivo autónomo, verificando que la duración de la batería permita un funcionamiento continuo de al menos 6 meses en modo de reposo o un mínimo de 500 activaciones de emergencia.

2.2. Priorización de requerimientos

| **ID** | **Nombre del Caso de Uso** | **Actores** | **Requisitos Funcionales Asociados** |
|---|---|---|---|
| CUS-01 | Enviar Alerta de Emergencia | Víctima | RF-003: Compilación y Preparación de Datos de Alerta.<br>RF-004: Transmisión LoRaWAN de Alerta.<br>RF-005: Indicador LED de Confirmación |
| CUS-02 | Obtener Ubicación GPS | - | RF-002: Activación y Captura de Ubicación GPS |
| CUS-03 | Gestionar Batería | - | RF-001: Gestión de Energía del Dispositivo |
| CUS-04 | Recibir y Reenviar Alerta | Servidor LoRaWAN | RF-006: Recepción y Procesamiento de Datos LoRaWAN (TTS)<br>RF-007: Reenvío de Alertas a Servidor Central (TTS) |

| | | | |
|---|---|---|---|
| | | | RF-024: Reenvío de Alerta del Servidor LoRaWAN |
| CUS-05 | Procesar Alerta en Servidor | Servidor Central | RF-008: Recepción y Decodificación de Alertas (SCM)<br>RF-009: Correlación de Dispositivo/Víctima.<br>RF-010: Identificación de Unidad de Respuesta Cercana. |
| CUS-06 | Despachar Alerta a Unidad | Servidor Central | RF-011: Despacho Automatizado de Alerta |
| CUS-07 | Iniciar Sesión App | Patrullero | RF-019: Autenticación de Patrullero |
| CUS-08 | Visualizar Alerta y Gestionar Estado | Patrullero | RF-014: Visualización de Alertas en Mapa.<br>RF-015: Visualización de Perfil de Víctima.<br>RF-016: Gestión del Estado de la Alerta. |
| CUS-09 | Actualizar Ubicación de Patrulla | Patrullero | RF-017: Actualización de Ubicación de Patrulla (App Patrulla)<br>RF-023: Actualización de Ubicación de Patrulla. |
| CUS-10 | Iniciar Sesión Dashboard | Operador, Administrador | RF-020: Autenticación para Dashboard. |
| CUS-11 | Monitorear Alertas en la Central | Operador, Administrador | RF-018: Interfaz de Usuario para Central de Monitoreo. |

| | | | |
|---|---|---|---|
| CUS-12 | Ver Ubicación de Patrullas | Operador, Administrador | RF-018: Interfaz de Usuario para Central de Monitoreo. |
| CUS-13 | Ver Detalles de Alerta | Patrullero, Operador, Administrador | RF-015: Visualización de Perfil de Víctima. |
| CUS-14 | Gestionar Usuarios y Roles | Administrador del Sistema | RF-021: Gestión de Usuarios y Roles. |
| CUS-15 | Registrar Dispositivo/Víctima | Operador, Administrador | RF-022: Registro de Dispositivo y Víctima. |

2.3. Requerimientos Funcionales

| **ID** | **Nombre del Requisito** | **Descripción de Requisito** | **Prioridad** |
|---|---|---|---|
| RF-001 | Gestión de Energía del Dispositivo | El firmware del dispositivo autónomo debe gestionar eficientemente el consumo de energía, utilizando modos de bajo consumo (deep sleep), para maximizar la autonomía de la batería. | Alta |
| RF-002 | Activación y Captura de Ubicación GPS | El dispositivo autónomo debe activar el módulo GPS bajo demanda (tras una | Alta |

| | | | |
|---|---|---|---|
| | | pulsación) para adquirir las coordenadas geográficas precisas de la víctima. | |
| RF-003 | Compilación y Preparación de Datos de Alerta | El firmware debe compilar los datos de alerta, incluyendo el ID único del dispositivo, las coordenadas GPS obtenidas y el nivel actual de la batería, en un formato apto para transmisión LoRaWAN. | Alta |
| RF-004 | Transmisión LoRaWAN de Alerta | El dispositivo autónomo debe transmitir de forma robusta y segura el paquete de datos de alerta a través del módulo LoRaWAN hacia la red LoRaWAN (The Things Stack). | Alta |
| RF-005 | Indicador LED de Confirmación | El dispositivo autónomo debe activar un LED de confirmación para proporcionar retroalimentación visual al usuario sobre el envío exitoso de la alerta. | Alta |
| RF-006 | Recepción y Procesamiento de Datos LoRaWAN (TTS) | El servidor de red The Things Stack (TTS) debe recibir, desduplicar y desencriptar los paquetes de datos provenientes de los dispositivos LoRaWAN. | Alta |
| RF-007 | Reenvío de Alertas a Servidor Central (TTS) | The Things Stack debe reenviar los datos de alerta procesados al Servidor Central de Monitoreo mediante integraciones configuradas (Webhooks o MQTT) en tiempo real. | Alta |

| | | | |
|---|---|---|---|
| RF-008 | Recepción y Decodificación de Alertas (SCM) | El Servidor Central de Monitoreo debe ser capaz de recibir y decodificar los datos de alerta enviados por The Things Stack. | Alta |
| RF-009 | Correlación de Dispositivo/Víctima | El Servidor Central de Monitoreo debe consultar la base de datos Firestore para correlacionar el ID del dispositivo de alerta con el perfil completo de la víctima asociada. | Alta |
| RF-010 | Identificación de Unidad de Respuesta Cercana | El Servidor Central de Monitoreo debe identificar la unidad policial o de serenazgo más cercana a la ubicación de la alerta activa, basándose en las posiciones actualizadas de las patrullas en Firestore. | Alta |
| RF-011 | Despacho Automatizado de Alerta | El Servidor Central de Monitoreo debe enviar de forma automatizada la alerta digital, incluyendo los datos de la víctima y su ubicación, a la Aplicación Cliente de la unidad de respuesta identificada. | Media |
| RF-012 | Almacenamiento y Gestión de Datos (Firestore) | El sistema debe utilizar Firestore para almacenar de forma segura los perfiles de víctimas, los registros históricos de alertas y las ubicaciones en tiempo real de las unidades de respuesta. | Alta |

| | | | |
|---|---|---|---|
| RF-013 | Sincronización de Datos en Tiempo Real | Firestore debe sincronizar en tiempo real las ubicaciones de las patrullas y el estado de las alertas entre el Servidor Central y las Aplicaciones Cliente, garantizando información actualizada para todos los usuarios. | Alta |
| RF-014 | Visualización de Alertas en Mapa (App Patrulla) | La Aplicación Cliente (móvil/tablet) para las unidades de respuesta debe mostrar interactivamente la ubicación de las alertas activas, la víctima y la propia patrulla en un mapa (API Google Maps). | Alta |
| RF-015 | Visualización de Perfil de Víctima (App Patrulla) | La Aplicación Cliente debe permitir al personal de la patrulla visualizar el perfil completo de la víctima asociada a una alerta. | Media |
| RF-016 | Gestión del Estado de la Alerta (App Patrulla) | La Aplicación Cliente debe permitir a la unidad de respuesta actualizar el estado de la emergencia (ej. "aceptar", "en ruta", "resuelto"). | Alta |
| RF-017 | Actualización de Ubicación de Patrulla (App Patrulla) | La Aplicación Cliente debe enviar la ubicación actual de la patrulla al Servidor Central de Monitoreo cada 5 segundos para su registro y análisis de proximidad. | Alta |
| RF-018 | Interfaz de Usuario para Central de | El Servidor Central de Monitoreo debe proporcionar una interfaz de usuario web (dashboard) para la visualización y gestión | Alta |

| | | | |
|---|---|---|---|
| | Monitoreo (Dashboard) | global de alertas y unidades por parte del personal de la central. | |
| RF-019 | Autenticación de Patrullero | El sistema debe requerir que el patrullero inicie sesión en la aplicación móvil con credenciales válidas antes de acceder a las funcionalidades de gestión de alertas. | Alta |
| RF-020 | Autenticación para Dashboard | El sistema debe requerir que el operador y el administrador inicien sesión en la interfaz web (dashboard) para acceder a las funciones de monitoreo y administración. | Alta |
| RF-021 | Gestión de Usuarios y Roles | El administrador del sistema debe poder crear, editar, eliminar y asignar roles (patrullero, operador, administrador) a los usuarios del sistema. | Alta |
| RF-022 | Registro de Dispositivo y Víctima | El operador o el administrador deben poder registrar un nuevo dispositivo de alerta, asociándolo con el perfil de una víctima específica en la base de datos del sistema. | Alta |
| RF-023 | Actualización de Ubicación de Patrulla | La aplicación del patrullero debe enviar automáticamente la ubicación actual de la unidad al servidor central en intervalos regulares para su monitoreo y correlación con las alertas. | Alta |

| | | | |
|---|---|---|---|
| RF-024 | Reenvío de Alerta del Servidor LoRaWAN | El Servidor LoRaWAN (The Thing Stack) debe estar configurado para reenviar los datos de alerta, de forma inmediata y automática, al Servidor Central de Monitoreo. | Alta |

2.4. Requerimientos No Funcionales – Atributos de Calidad

| | **ID** | **Nombre del Requerimiento** | **Descripción** | **Prioridad** |
|---|---|---|---|---|
| Fiabilidad | RNF001 | Disponibilidad del Sistema | El Servidor Central de Monitoreo, la base de datos Firestore y las integraciones con The Things Stack deben estar operativos y accesibles (24/7). | Alta |
| | RNF002 | Resistencia a Fallos de Transmisión (Dispositivo) | El firmware del dispositivo autónomo debe implementar mecanismos, como por ejemplo: reintentos de envío para asegurar que, ante una pérdida momentánea de señal LoRaWAN, la alerta se transmite exitosamente tan pronto como la red esté disponible. | Alta |
| | RNF003 | Tolerancia a Fallos de | El dispositivo debe estar diseñado con componentes | Media |

| | | | | |
|---|---|---|---|---|
| | | Componentes (Dispositivo) | robustos para resistir condiciones ambientales moderadas (temperatura, humedad) propias del uso diario en exteriores, sin que esto comprometa su funcionalidad principal. | |
| | RNF004 | Manejo de Errores (Sistema) | El Servidor Central de Monitoreo y las aplicaciones cliente deben manejar de forma elegante y robusta los errores, por ejemplo: GPS no disponible, pérdida de conexión a Internet, datos inválidos. | Alta |
| Rendimiento | RNF005 | Latencia de Alerta | El tiempo transcurrido desde la activación del botón en el dispositivo hasta la notificación de la alerta en la Aplicación Cliente del patrullero y en el Dashboard del operador no debe exceder los 15 segundos. Esto es crucial para la inmediatez de la respuesta. | Crítica |
| | RNF006 | Frecuencia de Actualización de Ubicación de Patrullas | La Aplicación Cliente de los patrulleros debe actualizar su ubicación en la base de datos con una frecuencia de cada 10 segundos para permitir una identificación precisa de la unidad más cercana. | Alta |
| | RNF007 | Capacidad de Procesamiento | El Servidor Central de Monitoreo debe ser capaz de procesar y | Media |

| | | | | |
|---|---|---|---|---|
| | | | despachar simultáneamente al menos 10 alertas por minuto sin degradación significativa del rendimiento. (Este valor puede ajustarse en base a la expectativa de incidentes). | |
| Usabilidad | RNF008 | Simplicidad de Activación (Dispositivo) | La activación de la alerta en el dispositivo autónomo debe realizarse mediante una única acción de pulsar un botón, sin requerir secuencias complejas, pantallas o habilidades técnicas previas por parte del usuario vulnerable. | Crítica |
| | RNF009 | Claridad de Interfaz (Aplicaciones) | La Aplicación Cliente del patrullero y el Dashboard de la Central deben presentar la información de manera clara, concisa e intuitiva, con elementos visuales fáciles de interpretar (mapas, iconos, estados de alerta). | Alta |
| | RNF010 | Operación Táctil Intuitiva (Aplicaciones) | La Aplicación Cliente debe ser totalmente funcional y fácil de operar mediante gestos táctiles en dispositivos móviles, incluso en condiciones de estrés. | Alta |
| | RNF011 | Autenticación de Dispositivos | Cada dispositivo autónomo debe autenticarse de forma segura en la red LoRaWAN (TTS) utilizando credenciales únicas (DevEUI, | Alta |

| | | | | |
|---|---|---|---|---|
| | | | AppKey, NwkKey) para asegurar que solo dispositivos autorizados puedan transmitir alertas. | |
| Seguridad | RNF012 | Acceso Controlado a la Información | El Servidor Central de Monitoreo (SCM) y las aplicaciones deben implementar un sistema de autenticación y autorización basado en roles, como: patrullero, operador, para asegurar que solo el personal autorizado acceda a la información pertinente (perfiles de víctimas, ubicación de patrullas, gestión de alertas). | Alta |
| | RNF013 | Comunicación Segura | Todas las comunicaciones entre los componentes del sistema (TTS a SCM, SCM a Aplicaciones Cliente) deben realizarse a través de canales seguros, como: HTTPS, MQTT sobre TL para proteger la integridad y confidencialidad de los datos transmitidos. | Alta |
| Mantenibilid | RNF014 | Autonomía del Dispositivo | El dispositivo autónomo debe tener una autonomía de batería de al menos 6 meses aproximadamente en modo de reposo o soportar un mínimo de 100 activaciones de emergencia, para reducir la frecuencia de recarga y asegurar su disponibilidad a largo plazo. | Crítica |

| | | | | |
|---|---|---|---|---|
| ad y Escalabilidad | RNF015 | Modularidad y Claridad del Código | El código fuente del firmware (ESP-IDF), el backend (Python/Node.js) y las aplicaciones cliente debe estar bien estructurado, modularizado y documentado para facilitar su futura modificación, mantenimiento y extensión. | Media |

2.5. Restricciones

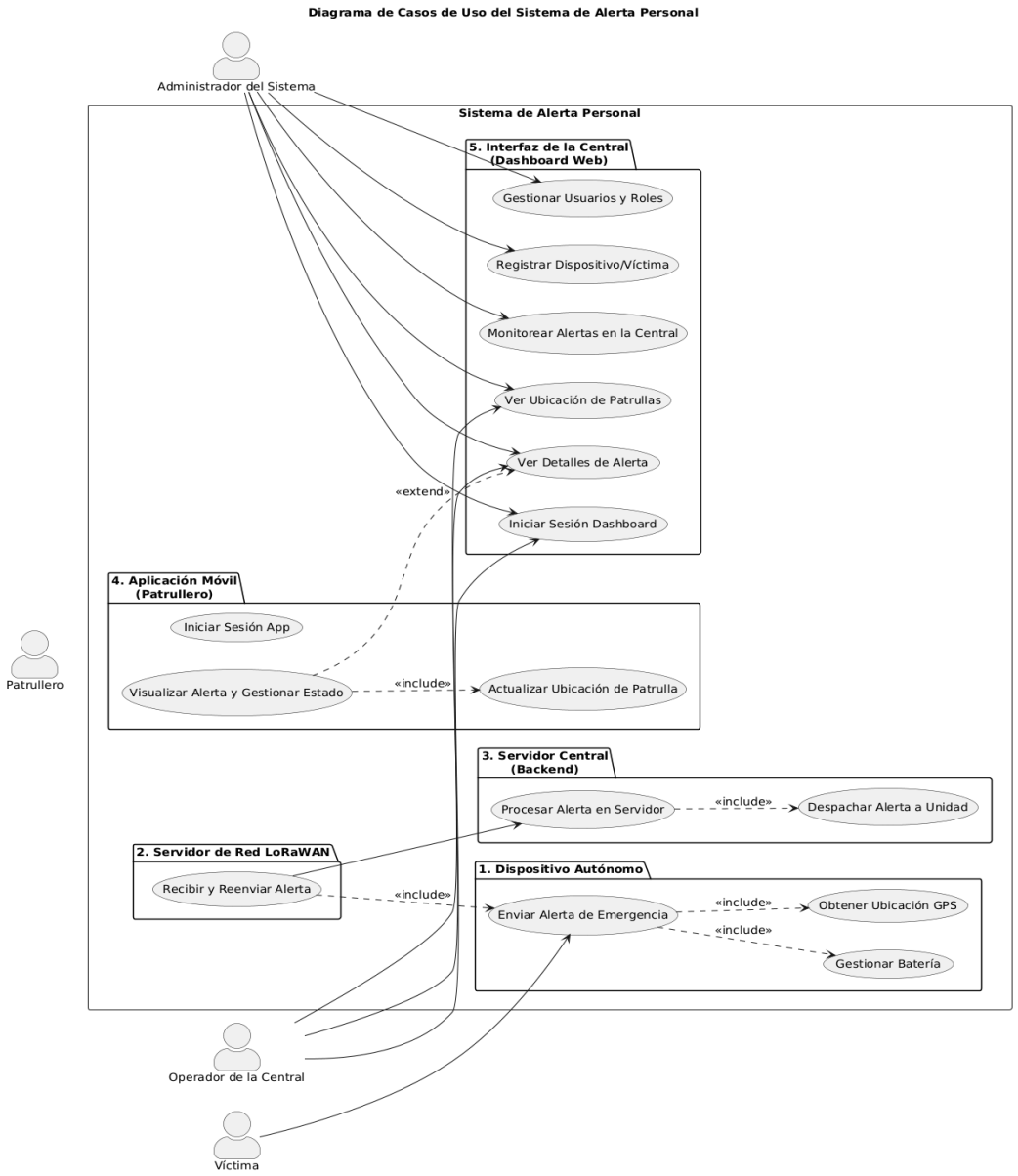
- Tecnológicas:
  Existen restricciones impuestas por la tecnología seleccionada. El sistema depende de la cobertura y capacidad de la red LoRaWAN en la provincia de Tacna, así como de la compatibilidad entre los dispositivos, gateways y las plataformas de backend. La autonomía y capacidad de procesamiento del dispositivo autónomo están limitadas por las características del hardware (memoria, batería, precisión del GPS). Asimismo, se deben considerar las limitaciones de las APIs externas (como Google Maps), la escalabilidad de FireBase.

- Seguridad y Privacidad:
  Debido a la naturaleza sensible de la información gestionada —datos personales de víctimas, ubicaciones en tiempo real y registros de emergencias—, el sistema debe cumplir con estrictos requisitos de seguridad y privacidad. Esto implica implementar mecanismos de autenticación robustos, cifrado de datos en tránsito y en reposo, control de acceso basado en roles y buenas prácticas de protección de información personal según la legislación peruana vigente. Cualquier vulnerabilidad en estos aspectos podría comprometer la confidencialidad e integridad de los datos y la confianza de los usuarios en el sistema.

## 3. Representación de la arquitectura del sistema

## 3.1. Vista de Caso de uso

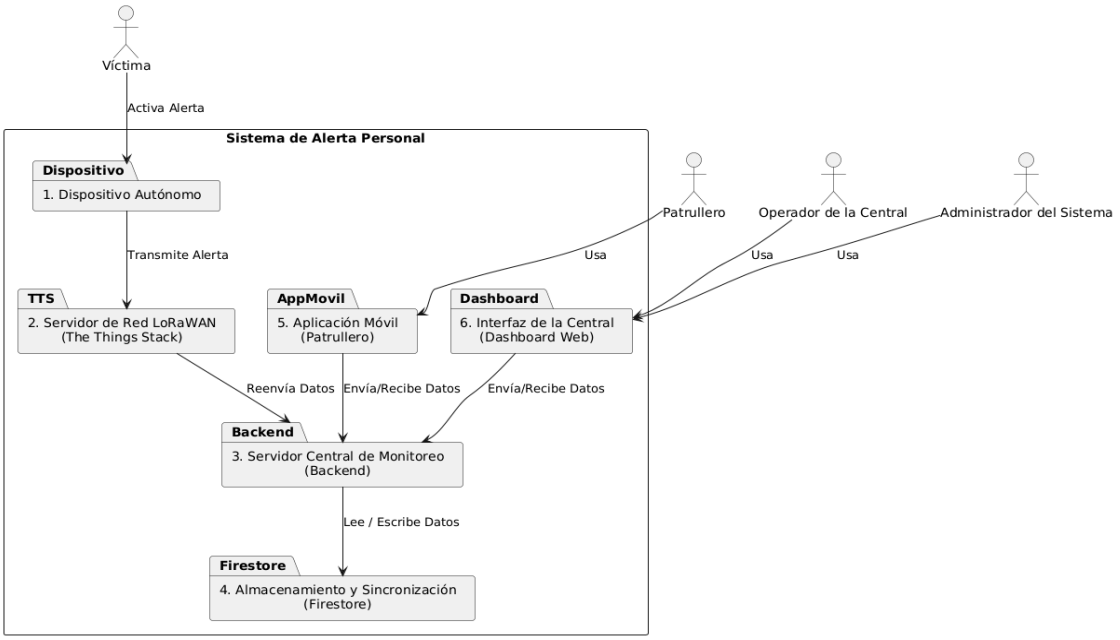
### 3.1.1. Diagramas de Casos de uso




https://drive.google.com/file/d/1wRhitCffQwXFgBCiw5y32uQml-79trWV/view?usp=sharing

## 3.2. Vista Lógica

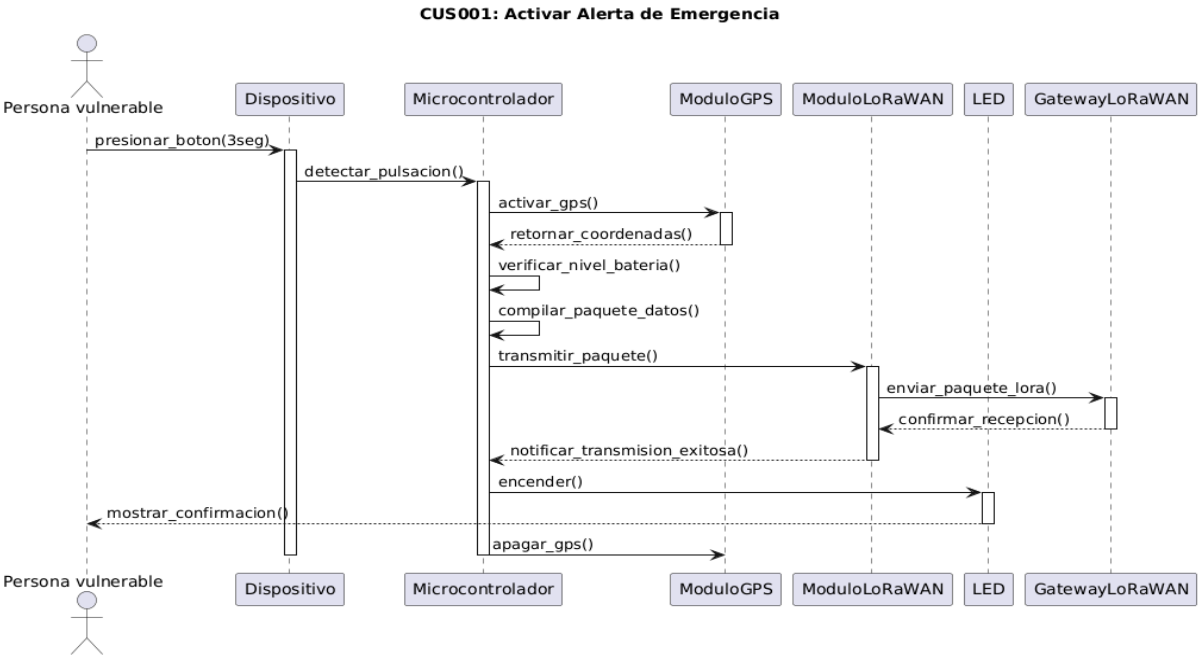
### 3.2.1. Diagrama de Subsistemas (paquetes)



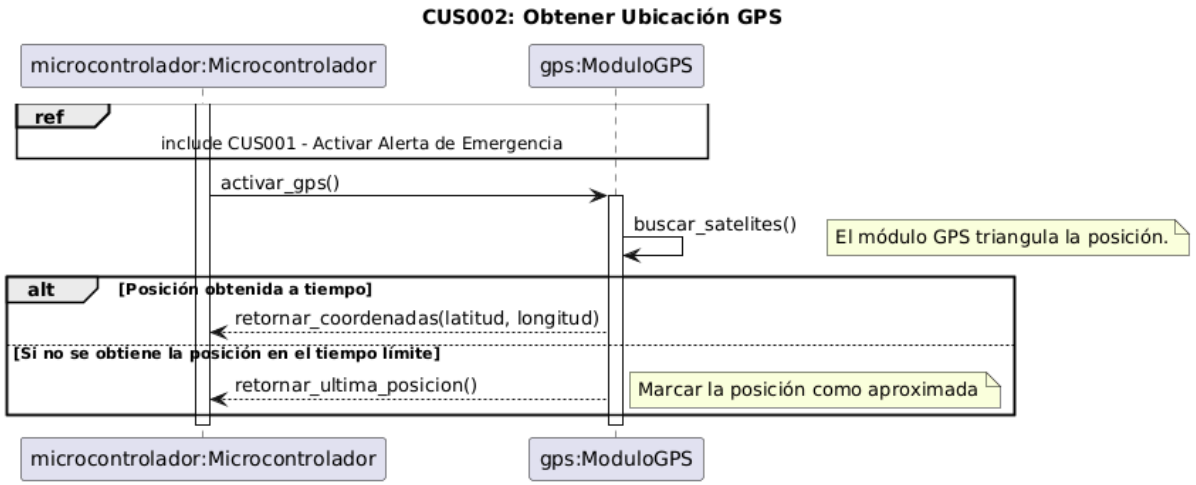

https://drive.google.com/file/d/1WlPatHQCFipPHUgErLxDde5q0Vqii72j/view?usp=sharing

### 3.2.2. Diagrama de Secuencia (vista de diseño)

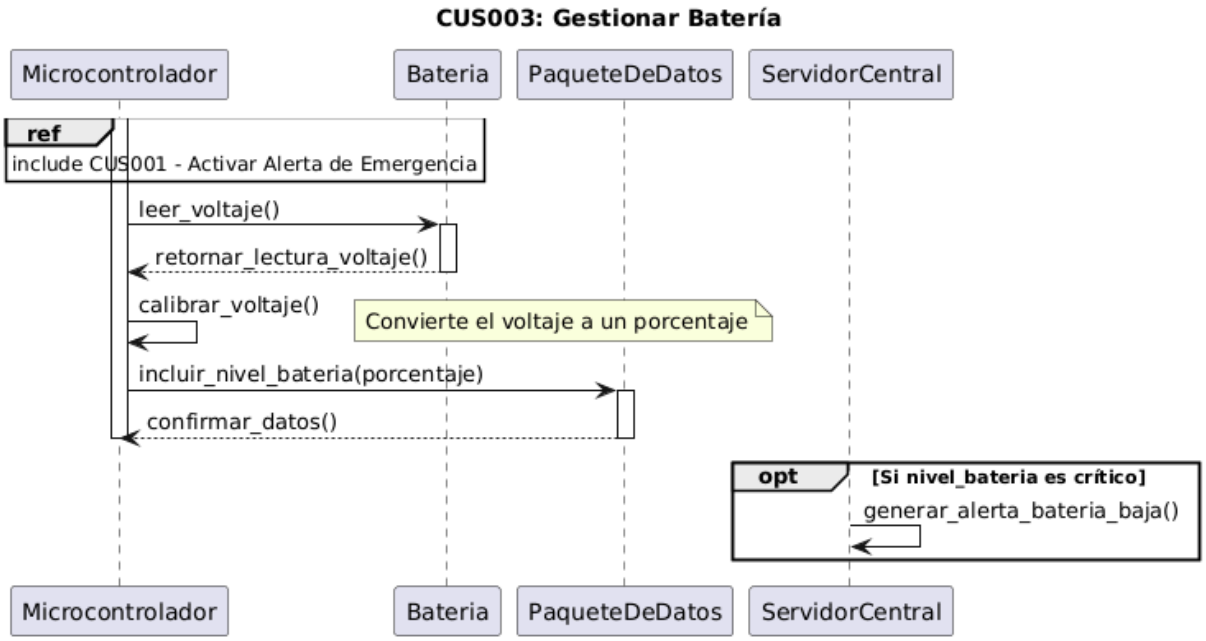
CUS001 - Activar Alerta de Emergencia



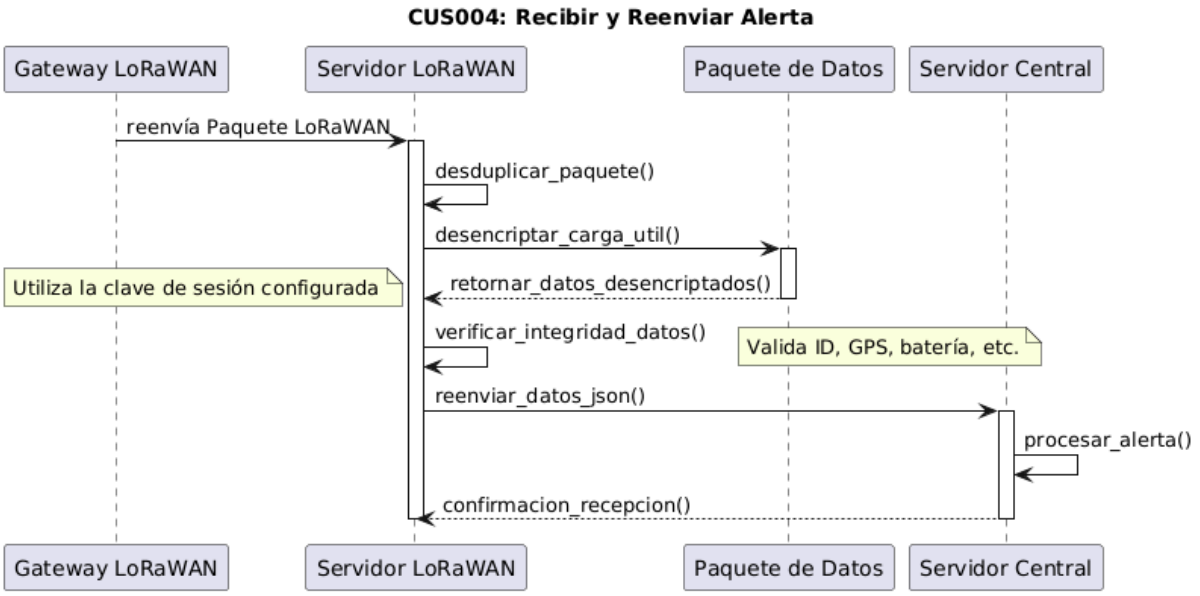
CUS002 - Obtener Ubicación GPS



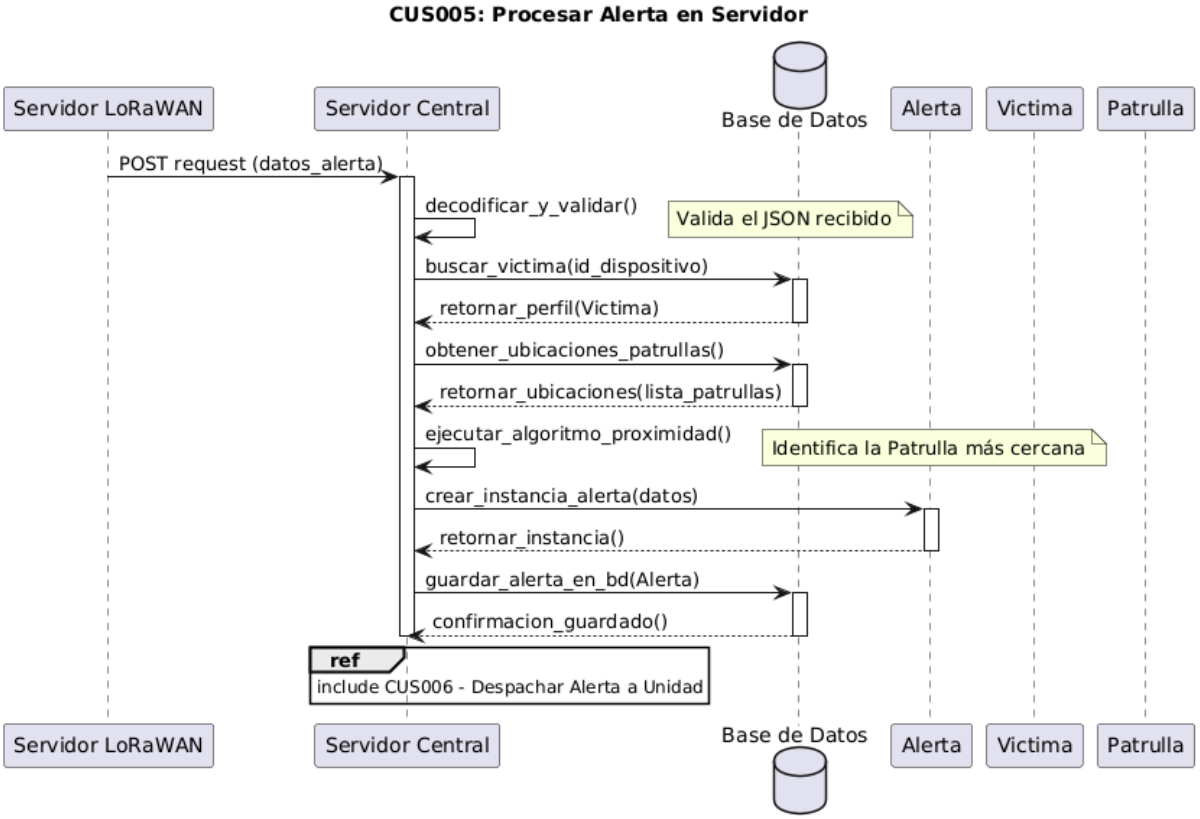

CUS003 - Gestionar Batería



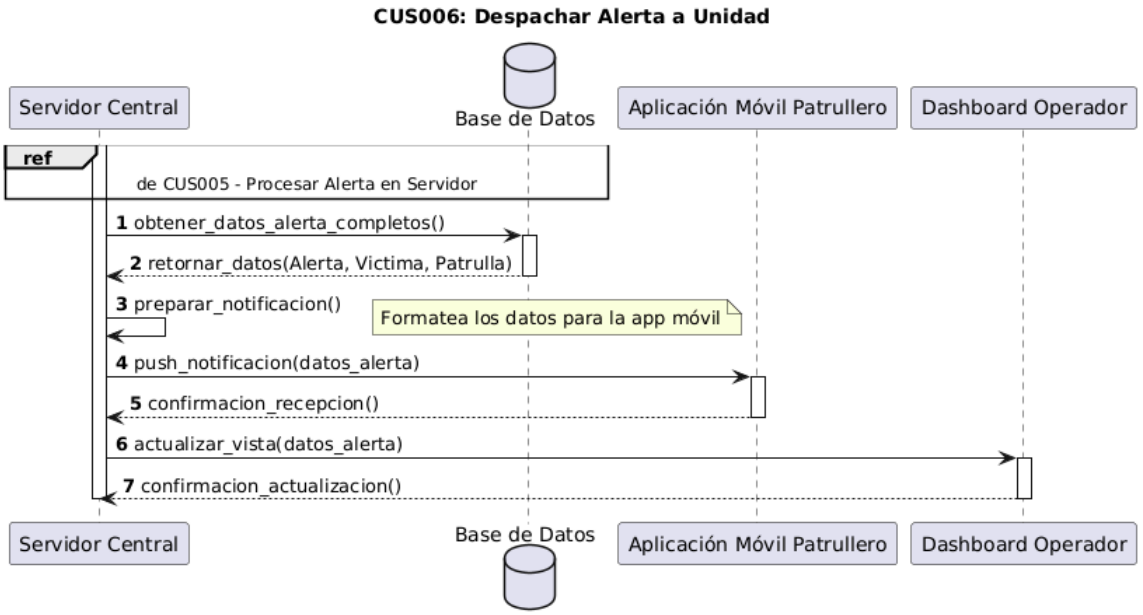
CUS004 - Recibir y Reenviar Alerta



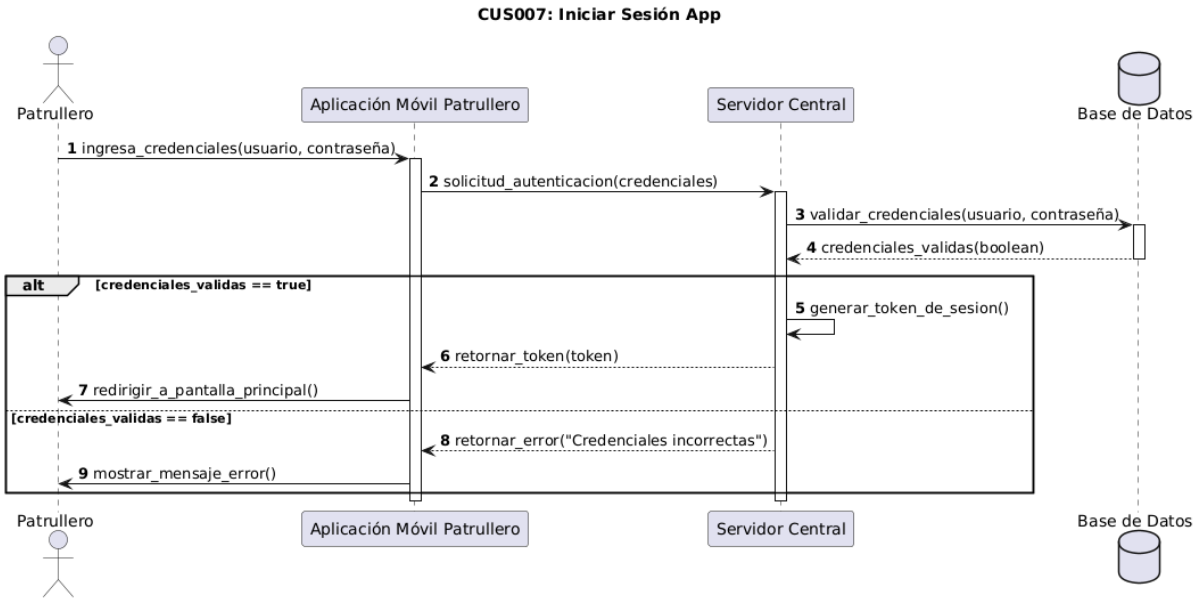

CUS005 - Procesar Alerta en Servidor



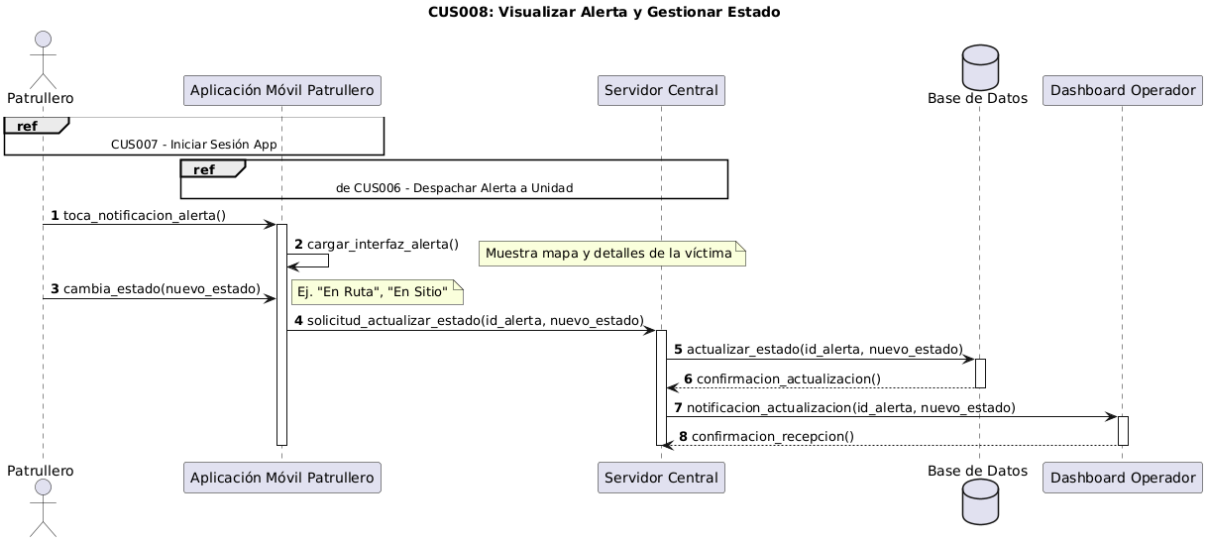
CUS006 - Despachar Alerta a Unidad



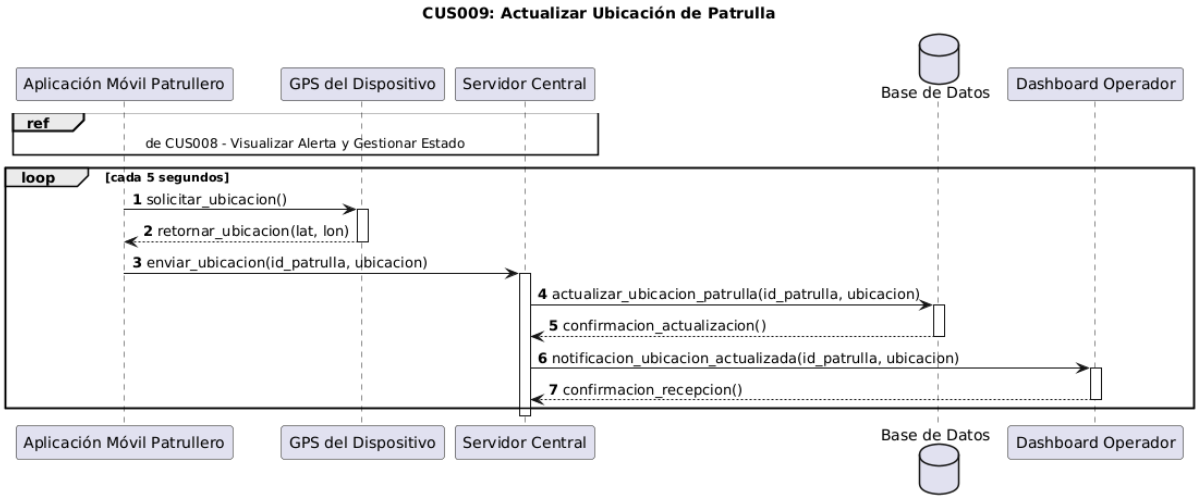

CUS007 - Iniciar Sesión App



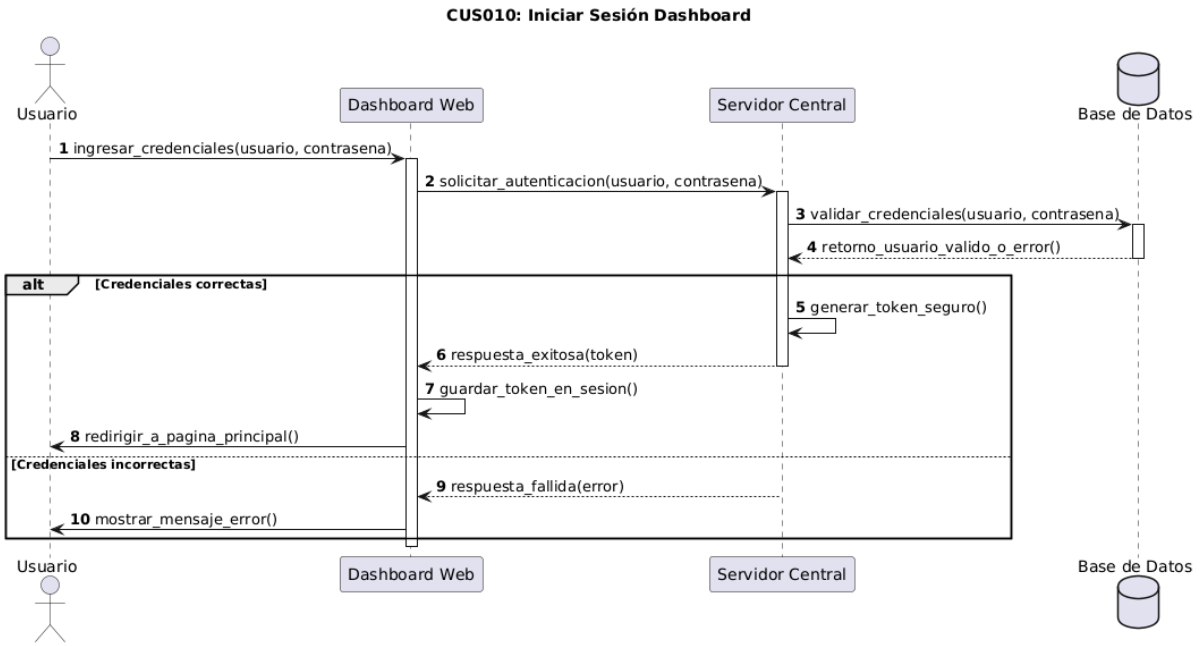
## CUS008 - Visualizar Alerta y Gestionar Estado



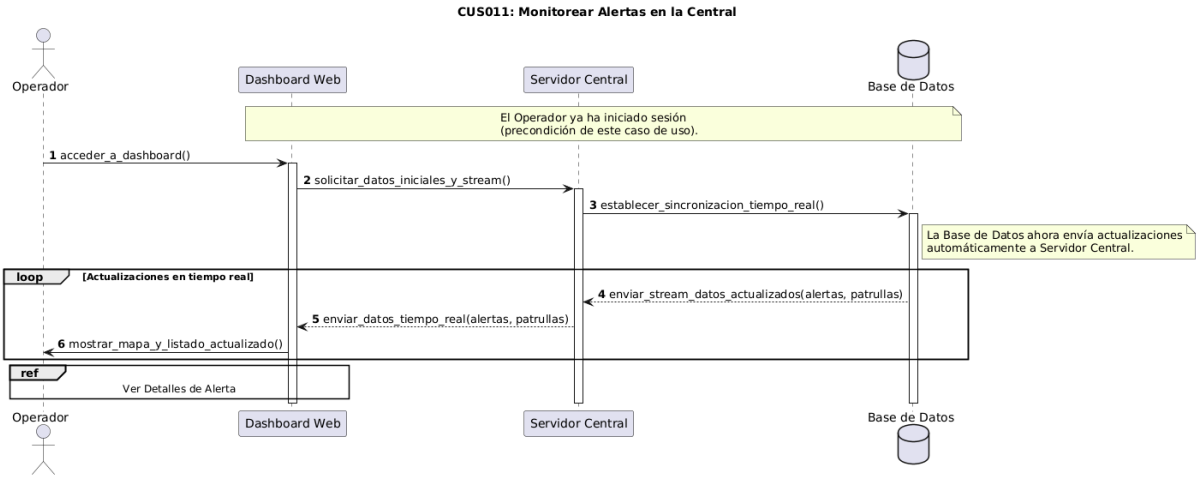

## CUS009 - Actualizar Ubicación de Patrulla



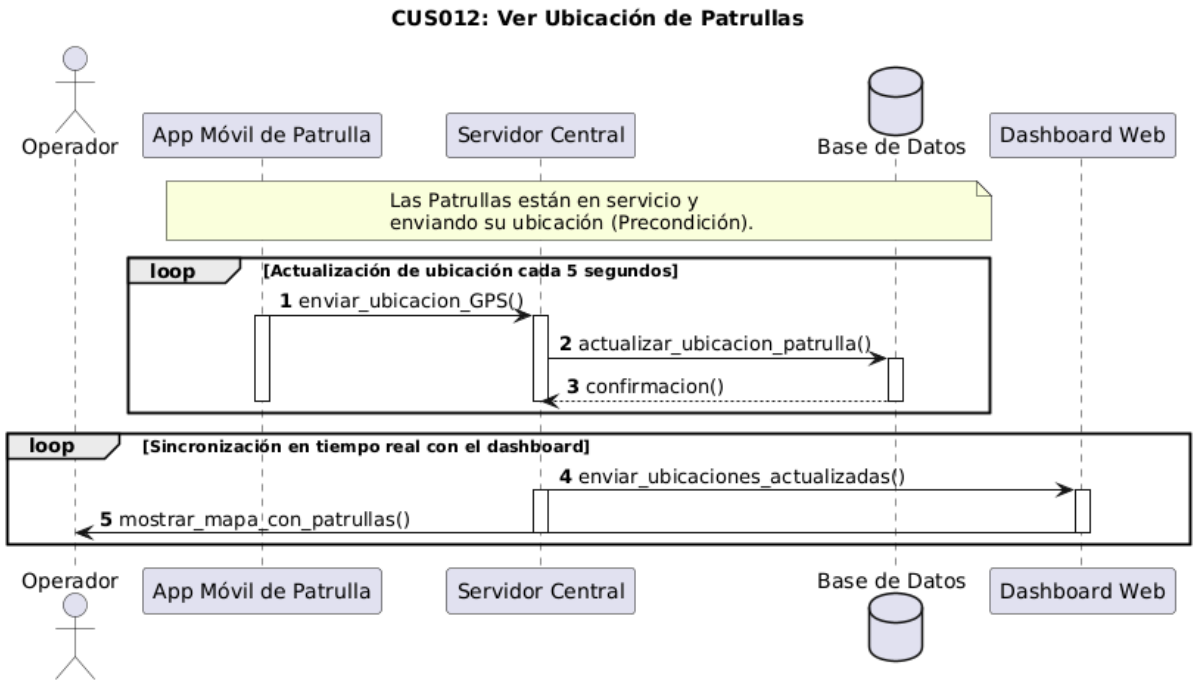
## CUS010 - Iniciar Sesión Dashboard




## CUS011 - Monitorear Alertas en la Central

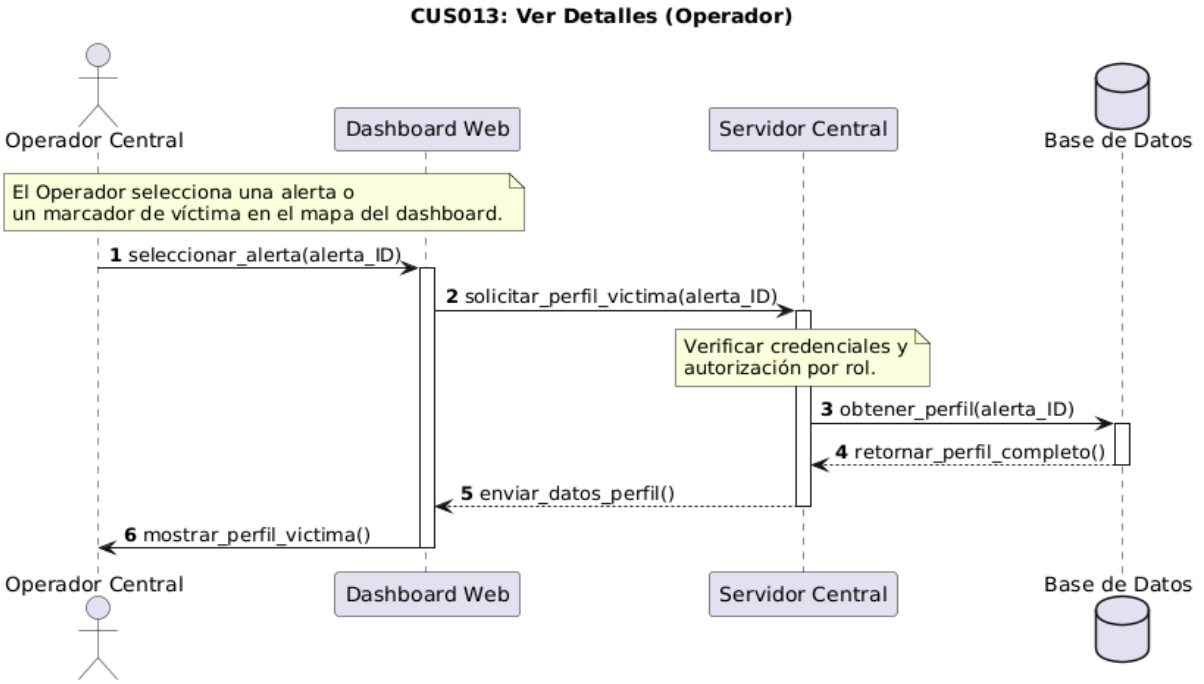


CUS012 - Ver Ubicación de Patrullas

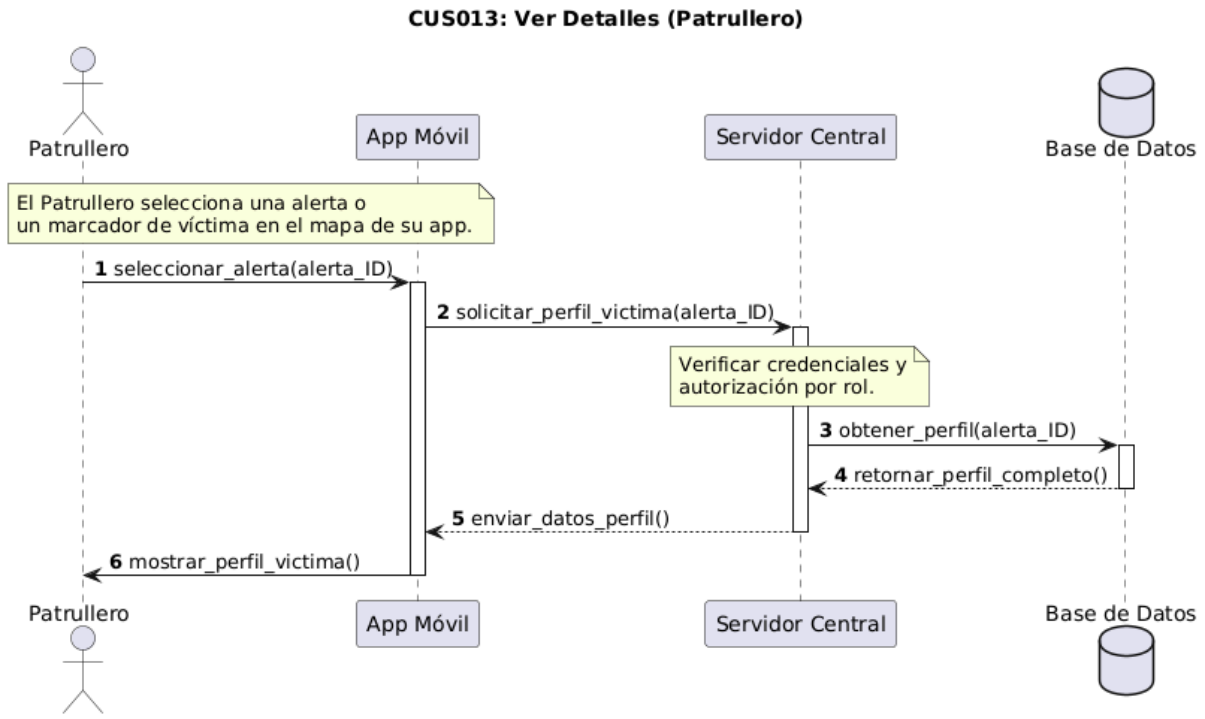



CUS013 - Ver Detalles de Alerta

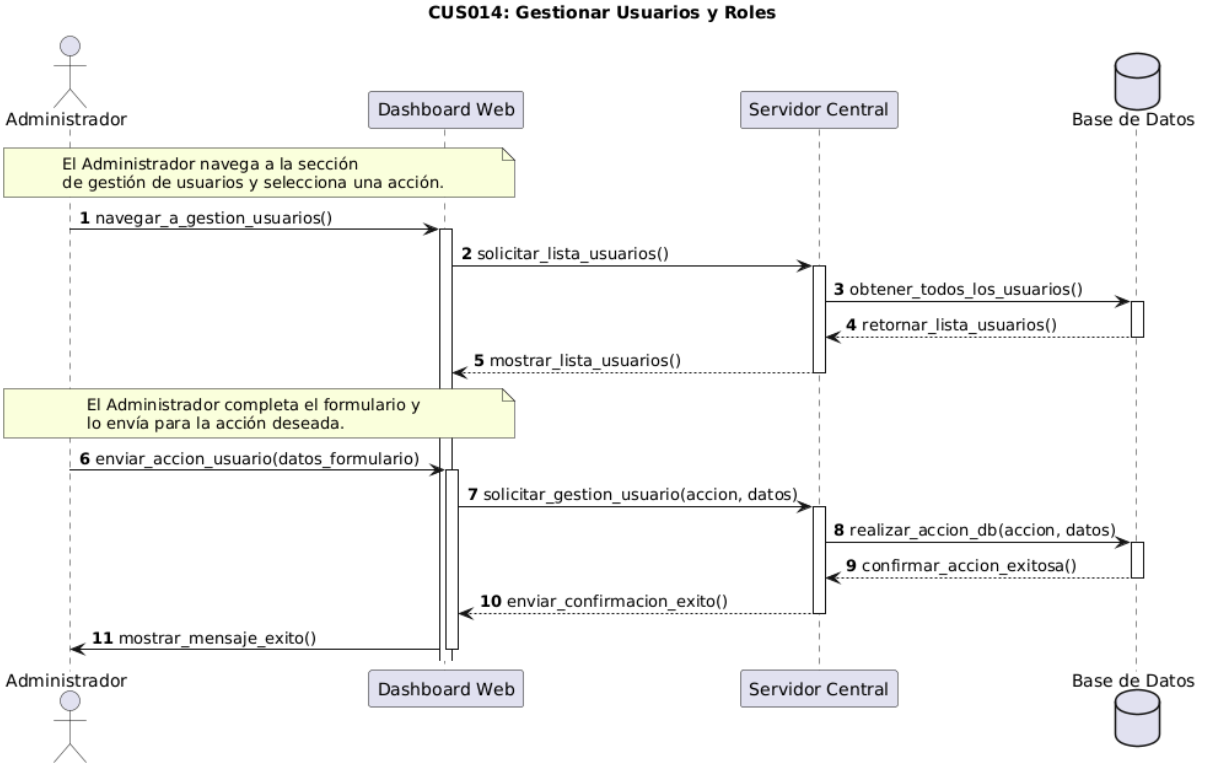
Operador



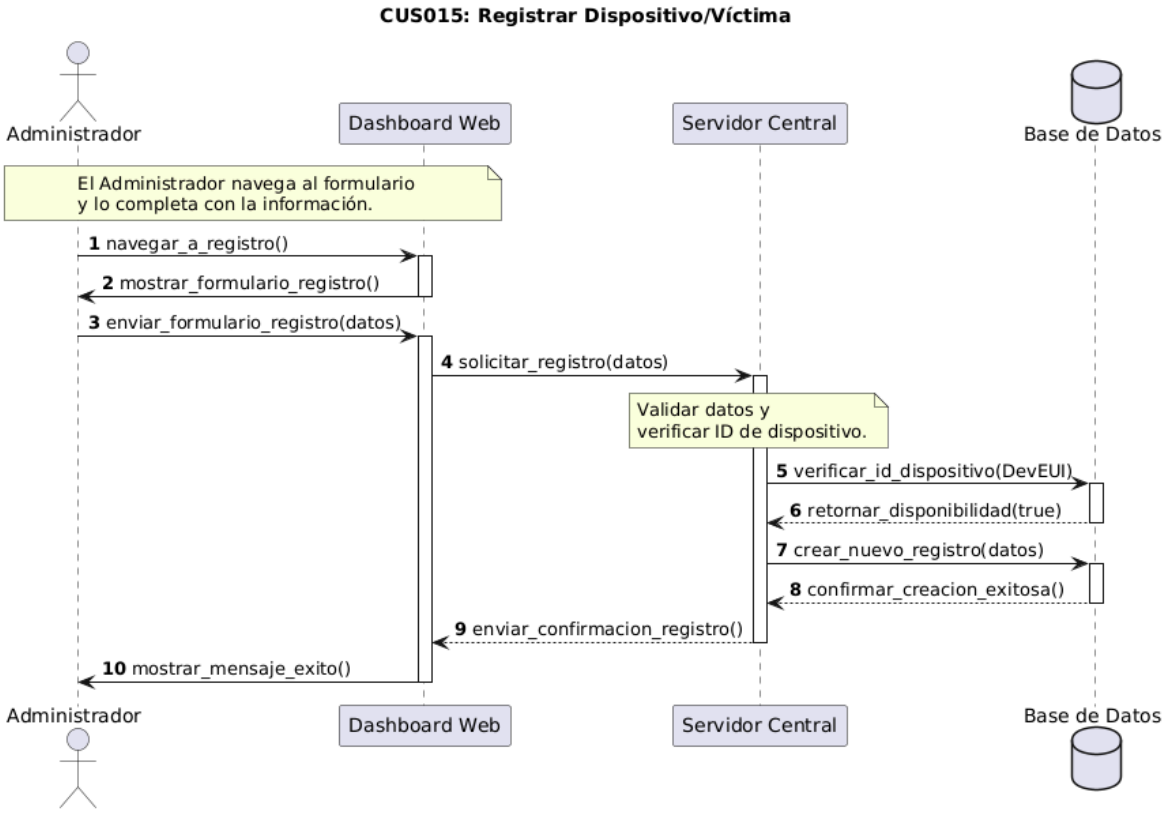
Patrullero




CUS014 - Gestionar Usuarios y Roles

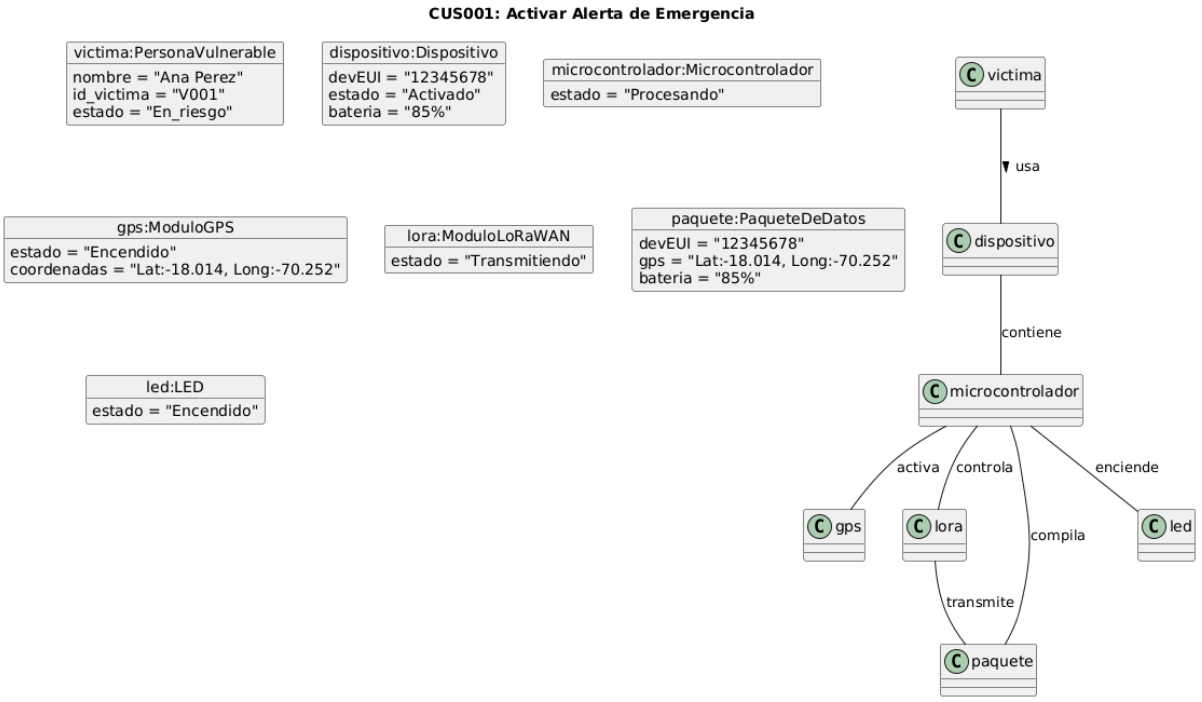


CUS015 - Registrar Dispositivo/Víctima

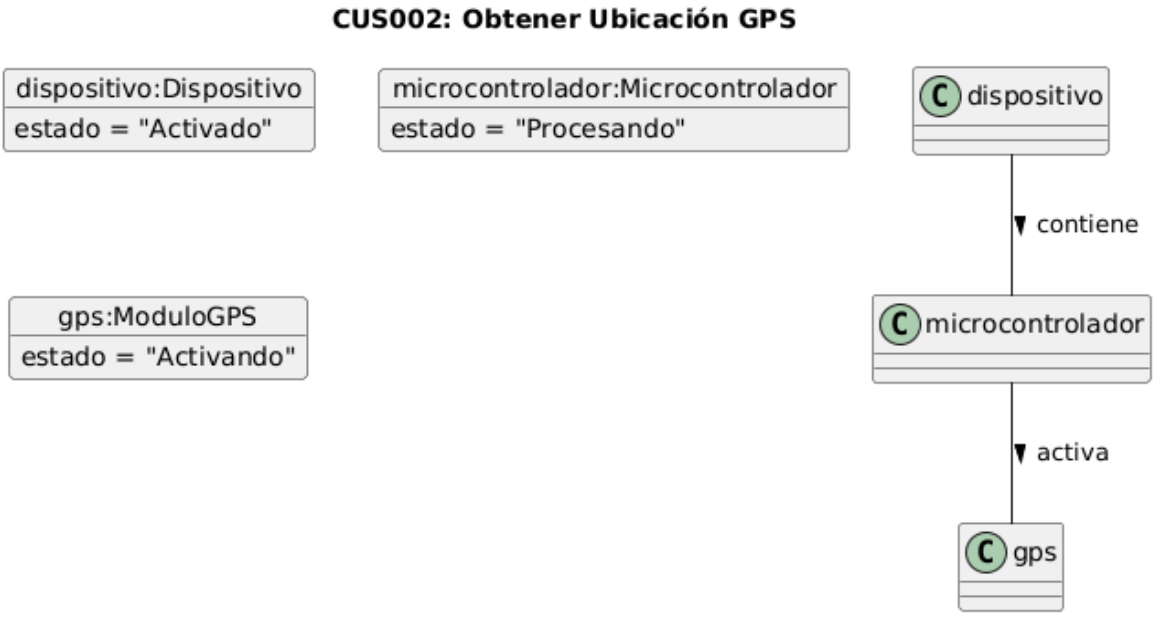


### 3.2.3. Diagrama de Objetos

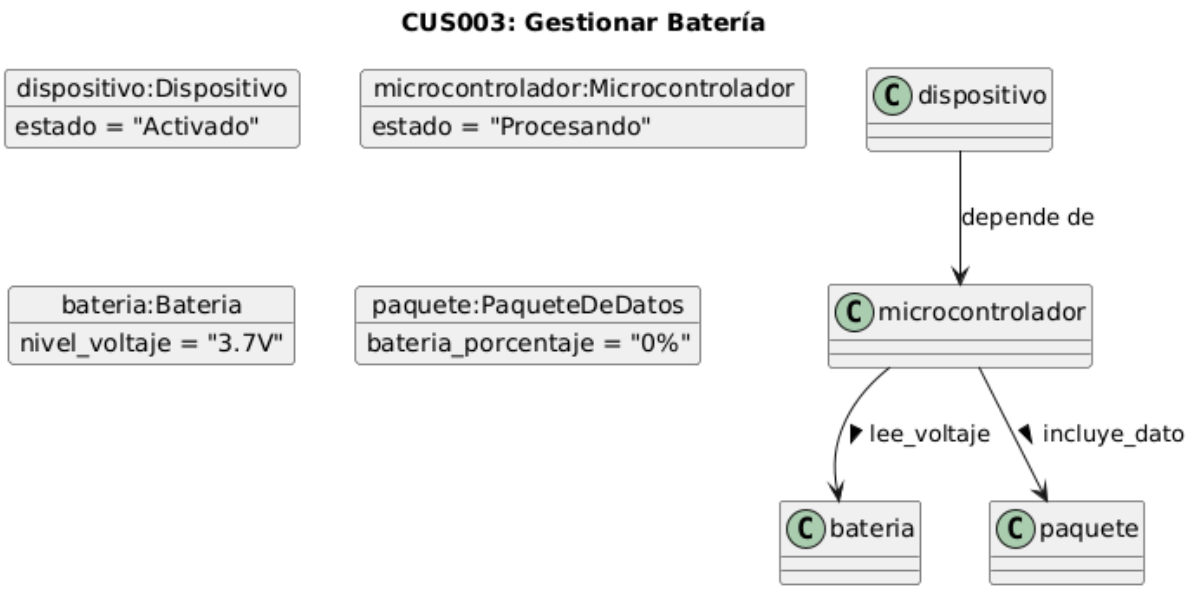
CUS001 - Activar Alerta de Emergencia



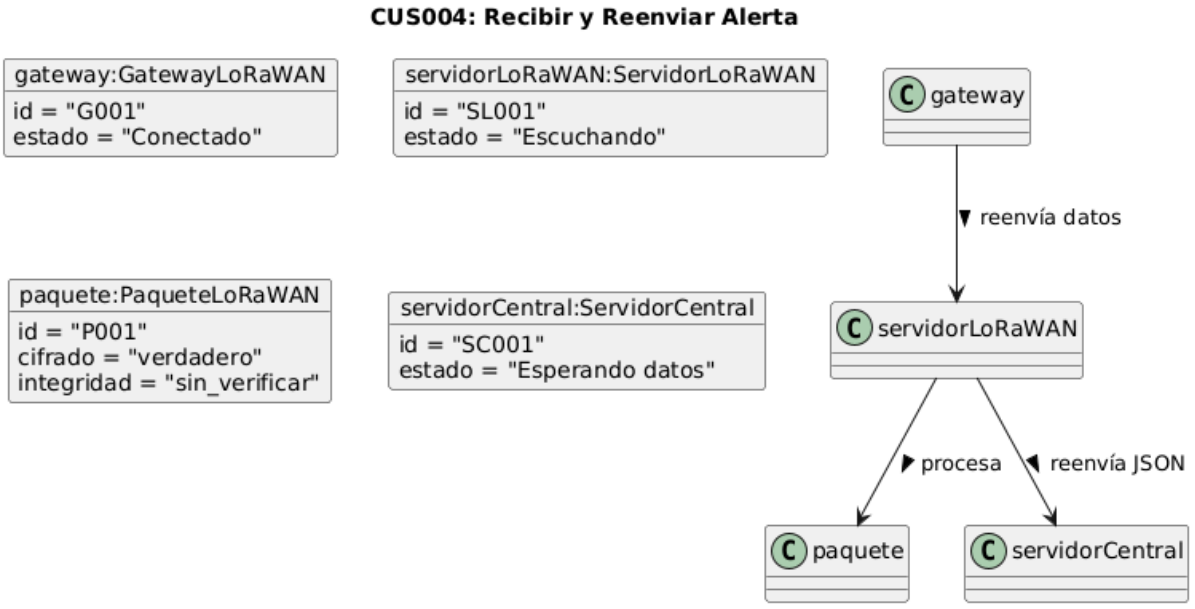

CUS002 - Obtener Ubicación GPS



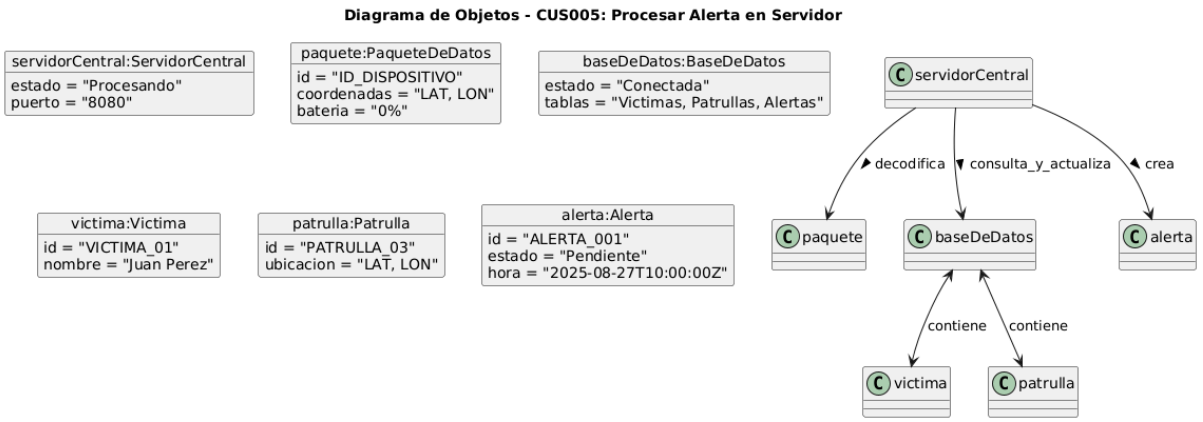
CUS003 - Gestionar Batería



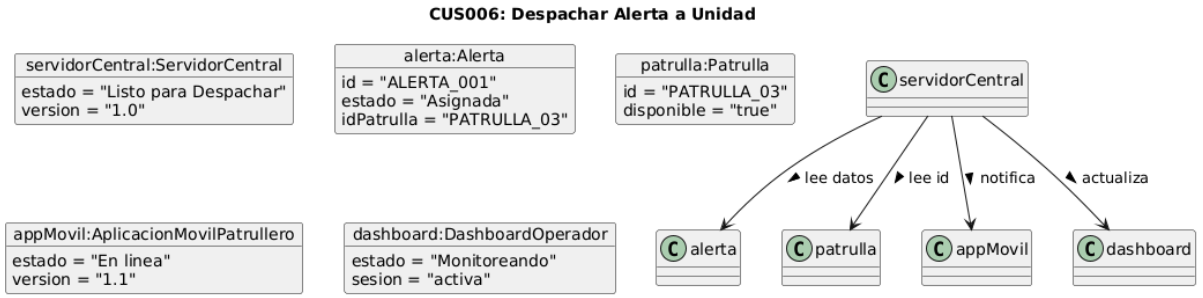

CUS004 - Recibir y Reenviar Alerta



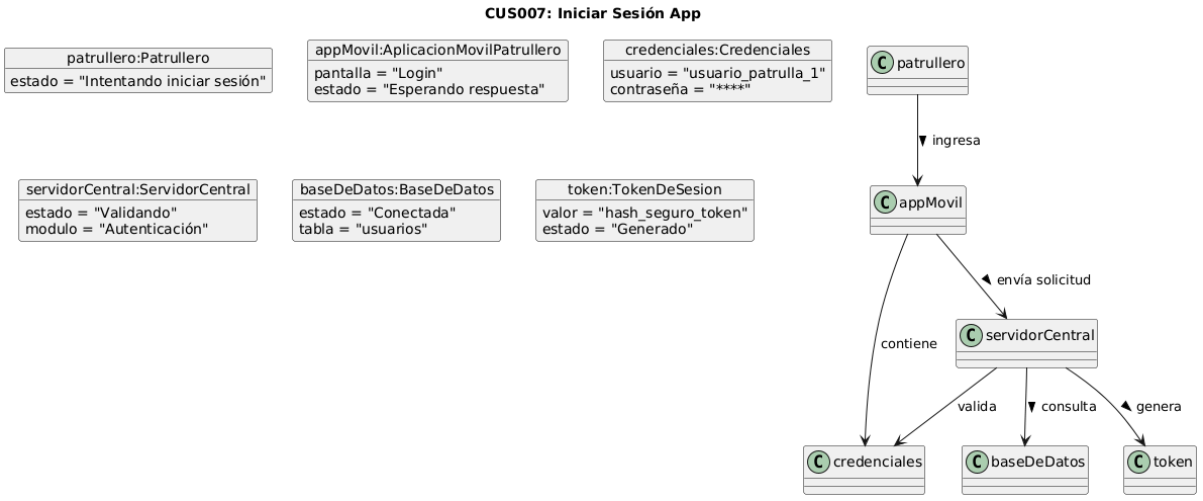
## CUS005 - Procesar Alerta en Servidor



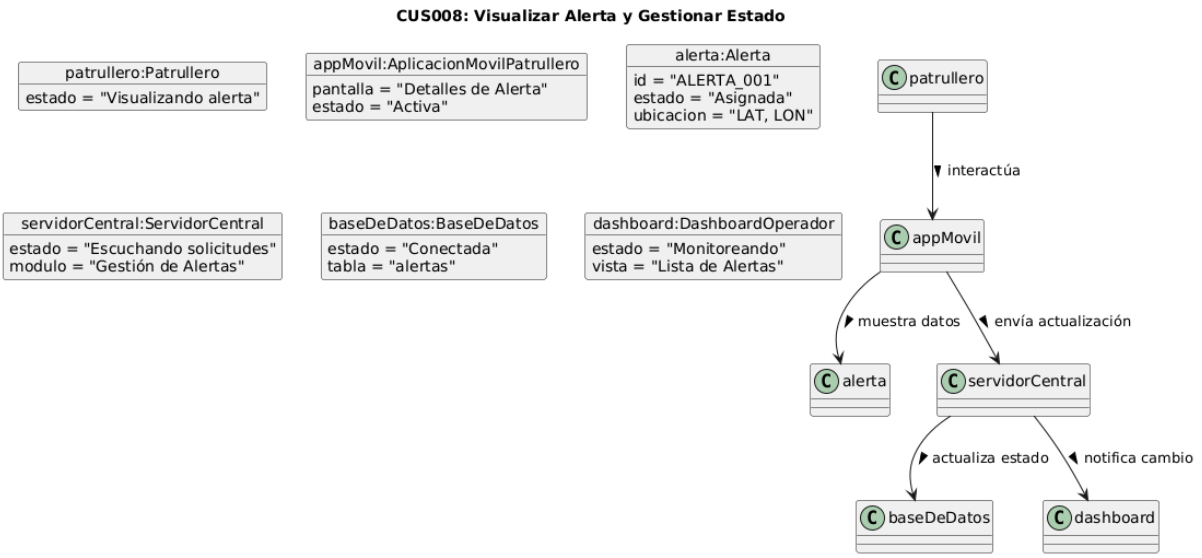

## CUS006 - Despachar Alerta a Unidad



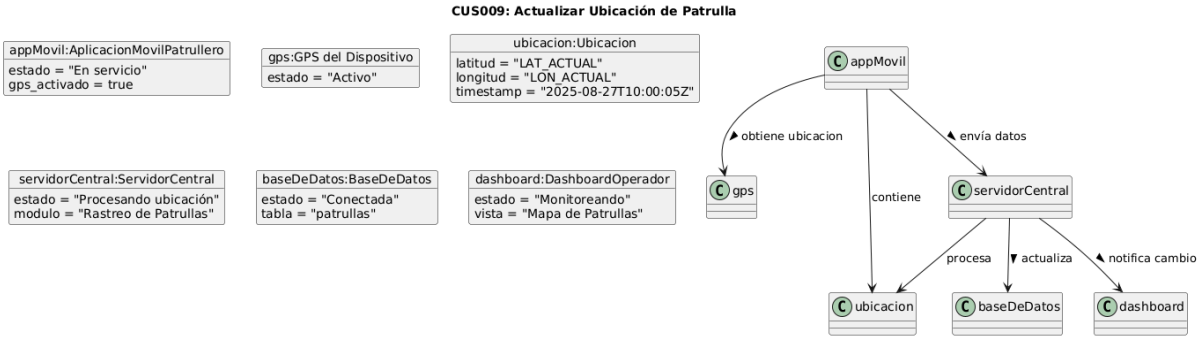

## CUS007 - Iniciar Sesión App



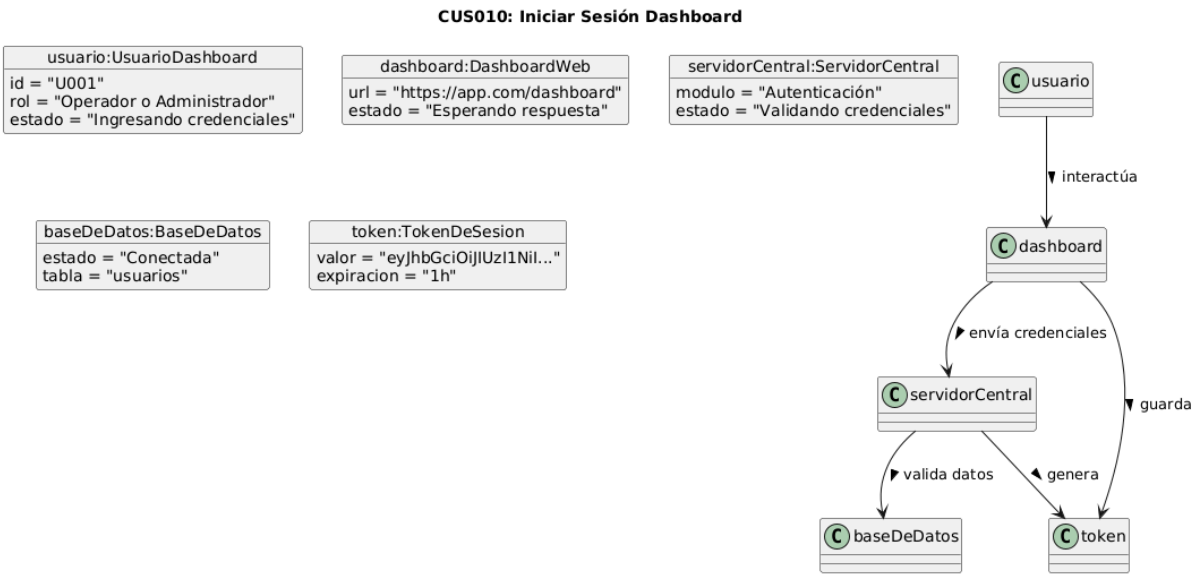
## CUS008 - Visualizar Alerta y Gestionar Estado



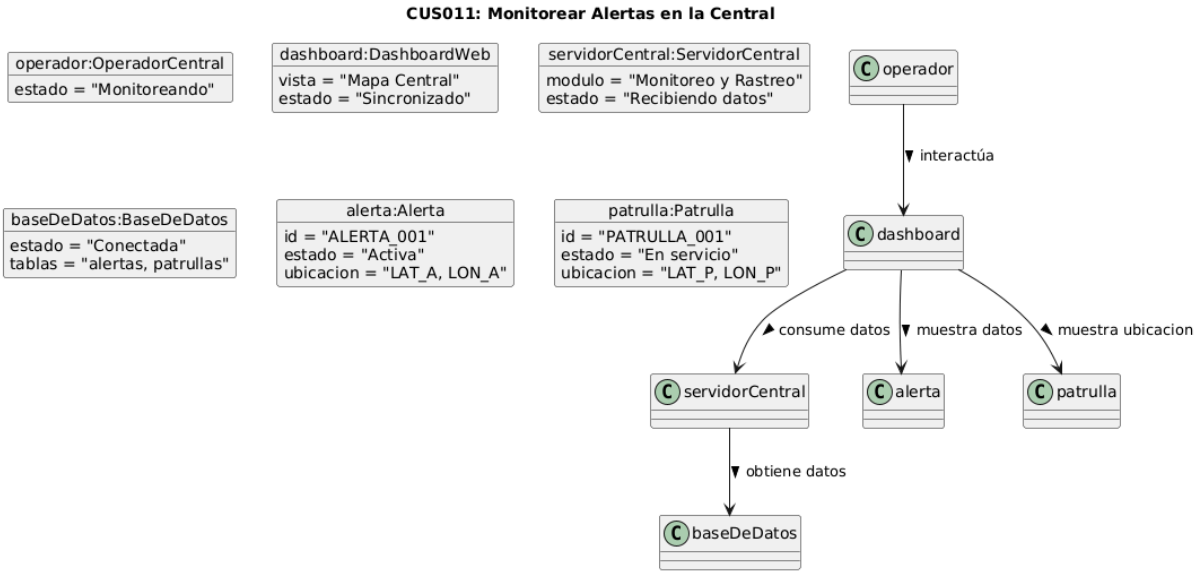

## CUS009 - Actualizar Ubicación de Patrulla



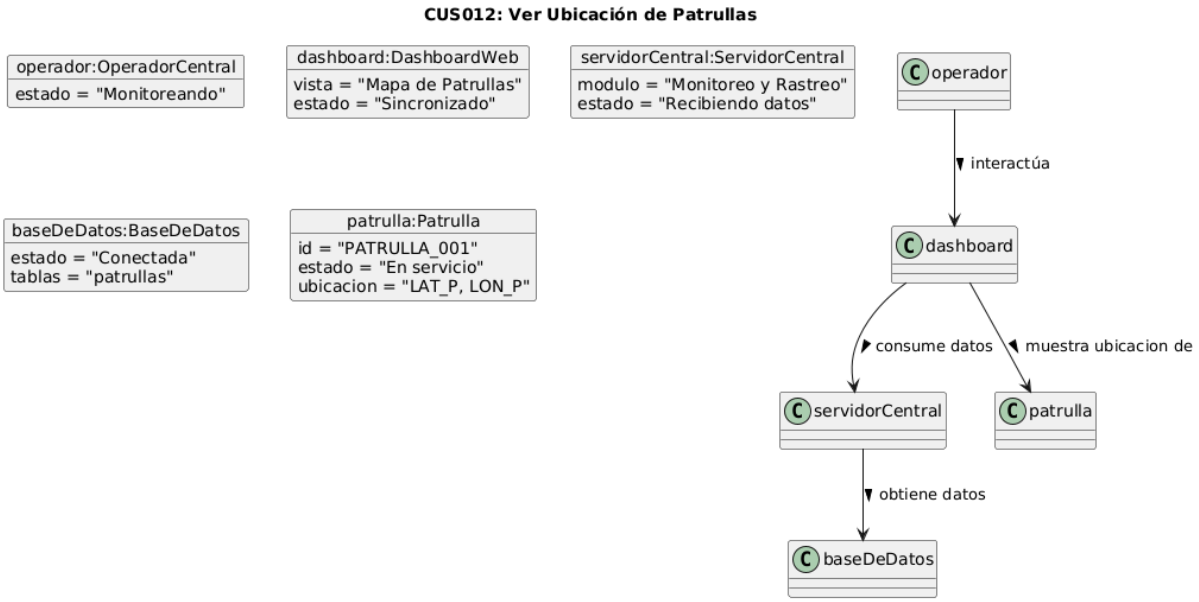
## CUS010 - Iniciar Sesión Dashboard



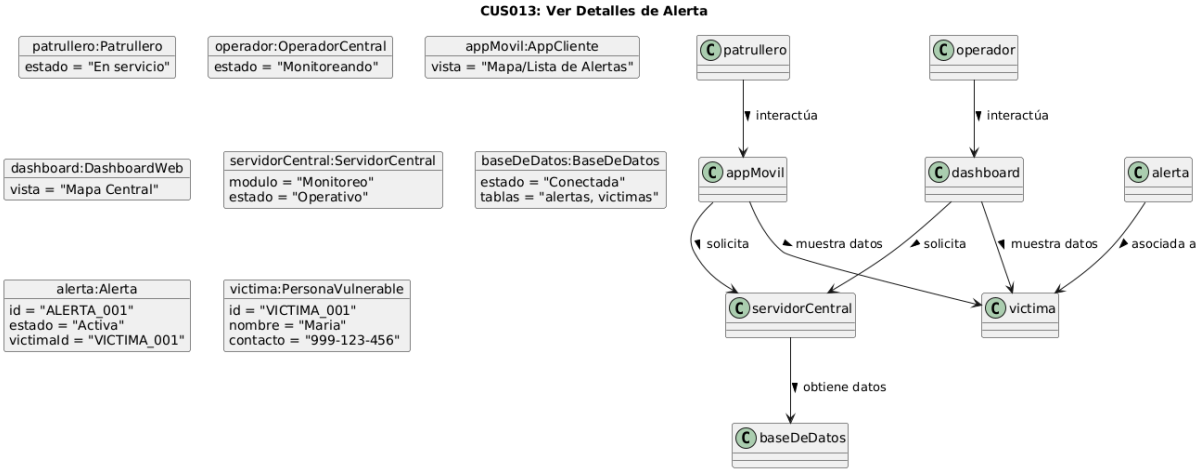

## CUS011 - Monitorear Alertas en la Central



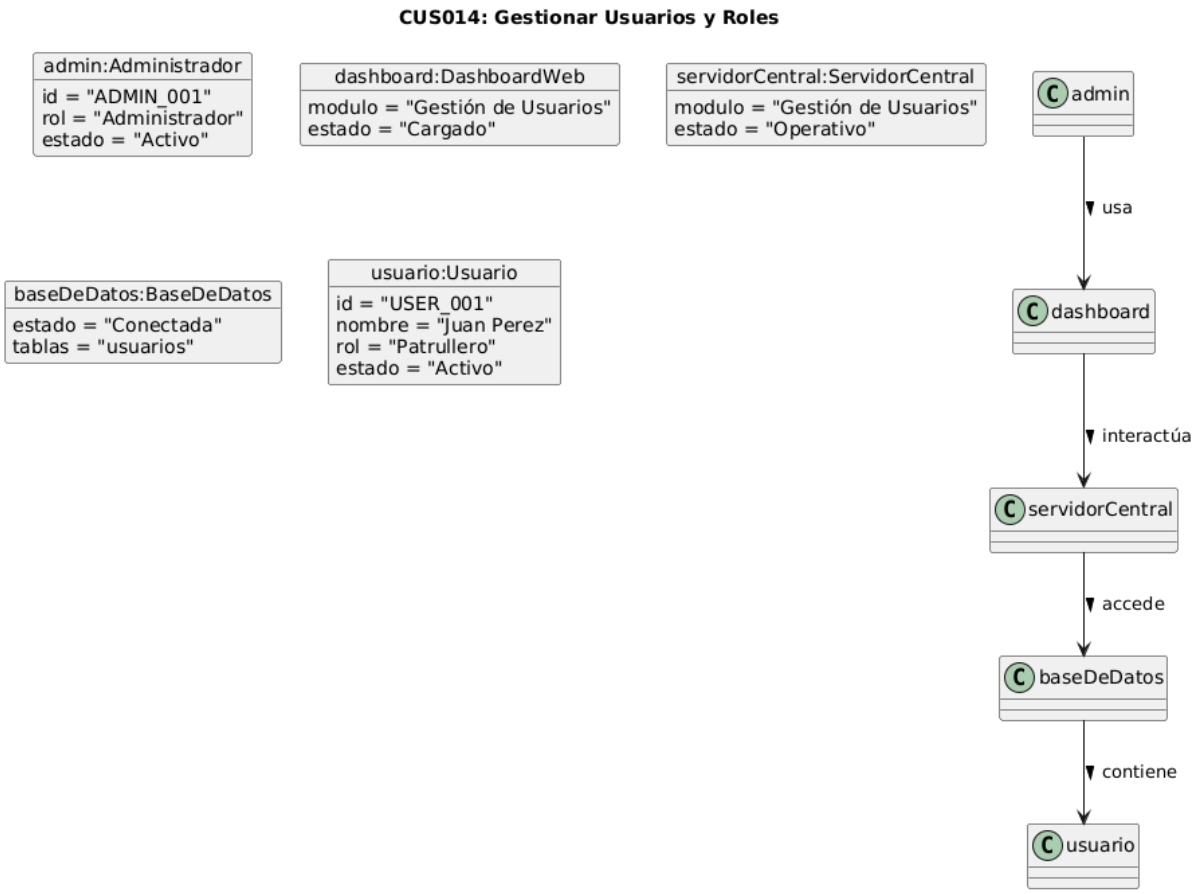
## CUS012 - Ver Ubicación de Patrullas



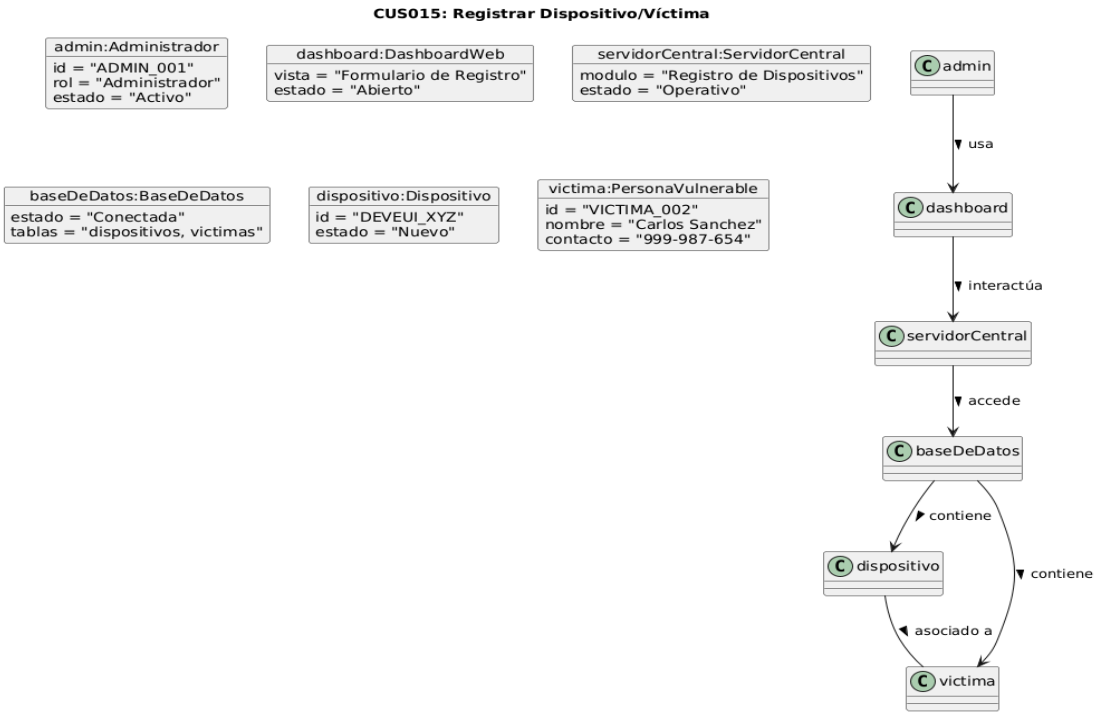

## CUS013 - Ver Detalles de Alerta



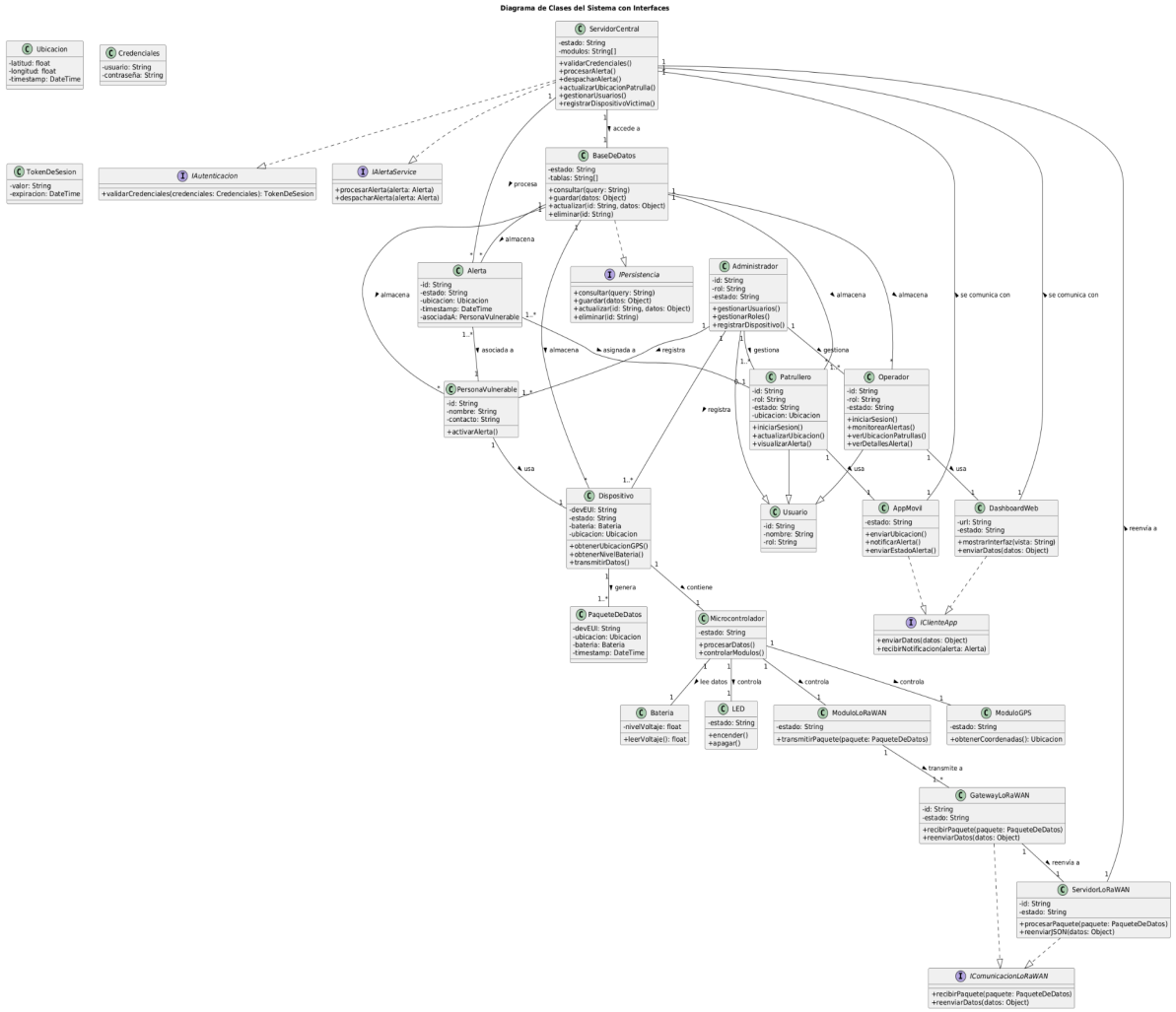
CUS014 - Gestionar Usuarios y Roles




CUS015 - Registrar Dispositivo/Víctima

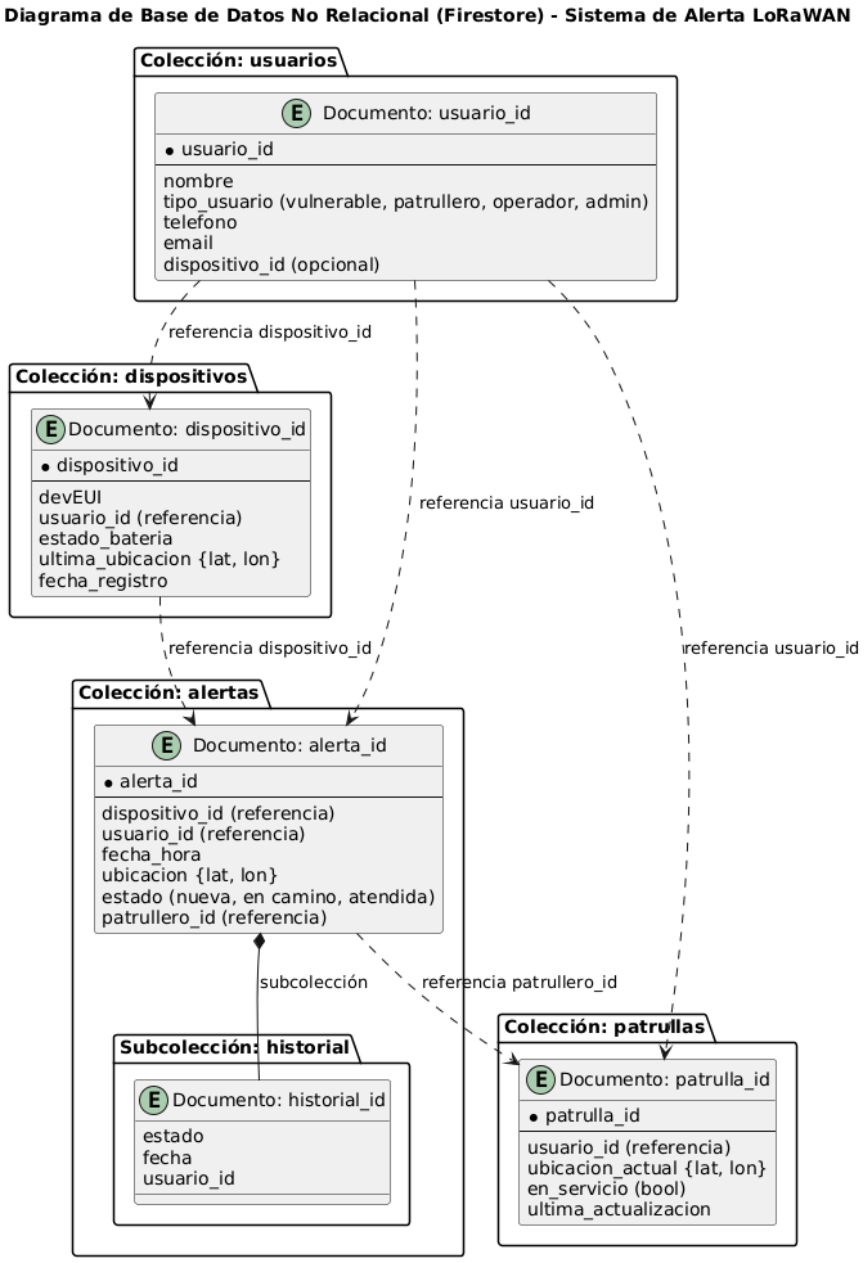


### 3.2.4. Diagrama de Clases



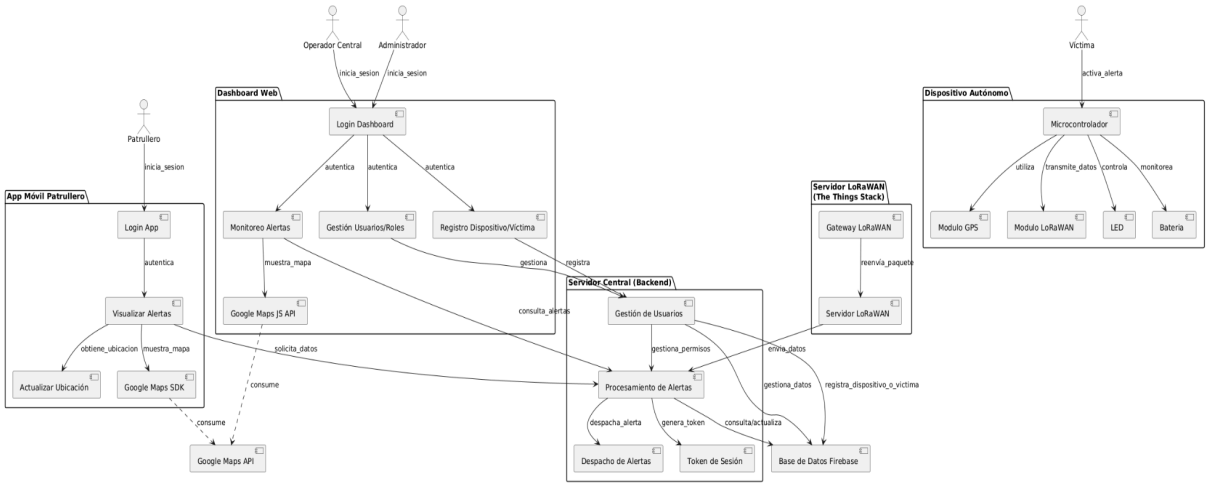

https://drive.google.com/file/d/1zb_EjA5_ykg9YMr7FKTM9ll_HPMgcr0C/view?usp=sharing

### 3.2.5. Diagrama de Base de datos (relacional o no relacional)



## 3.3. Vista de Implementación

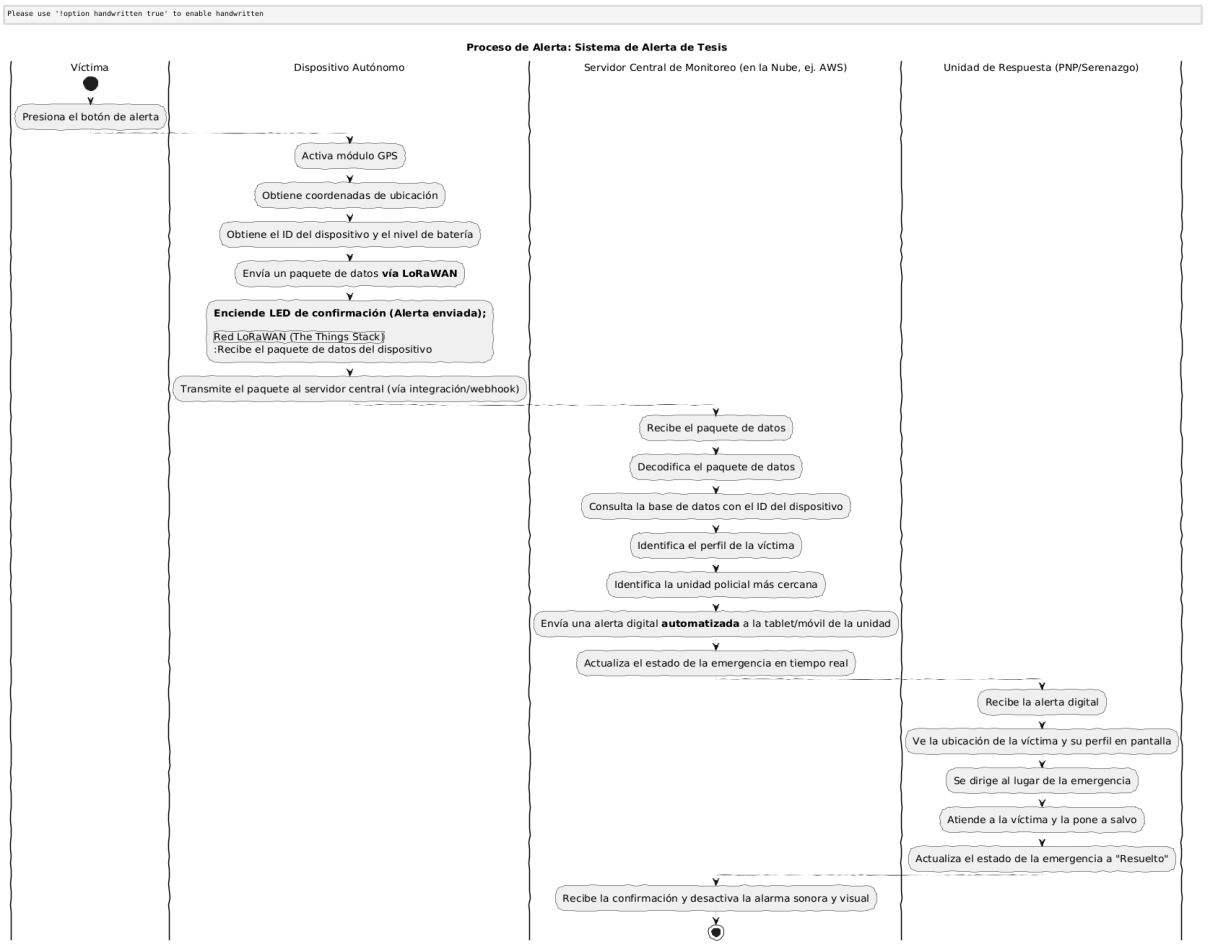
### 3.3.1. Diagrama de arquitectura del sistema (Diagrama de componentes)




https://drive.google.com/file/d/1wnlvsEkHclv984ql_HJhuHNSZZiMt2y6/view?usp=sharing

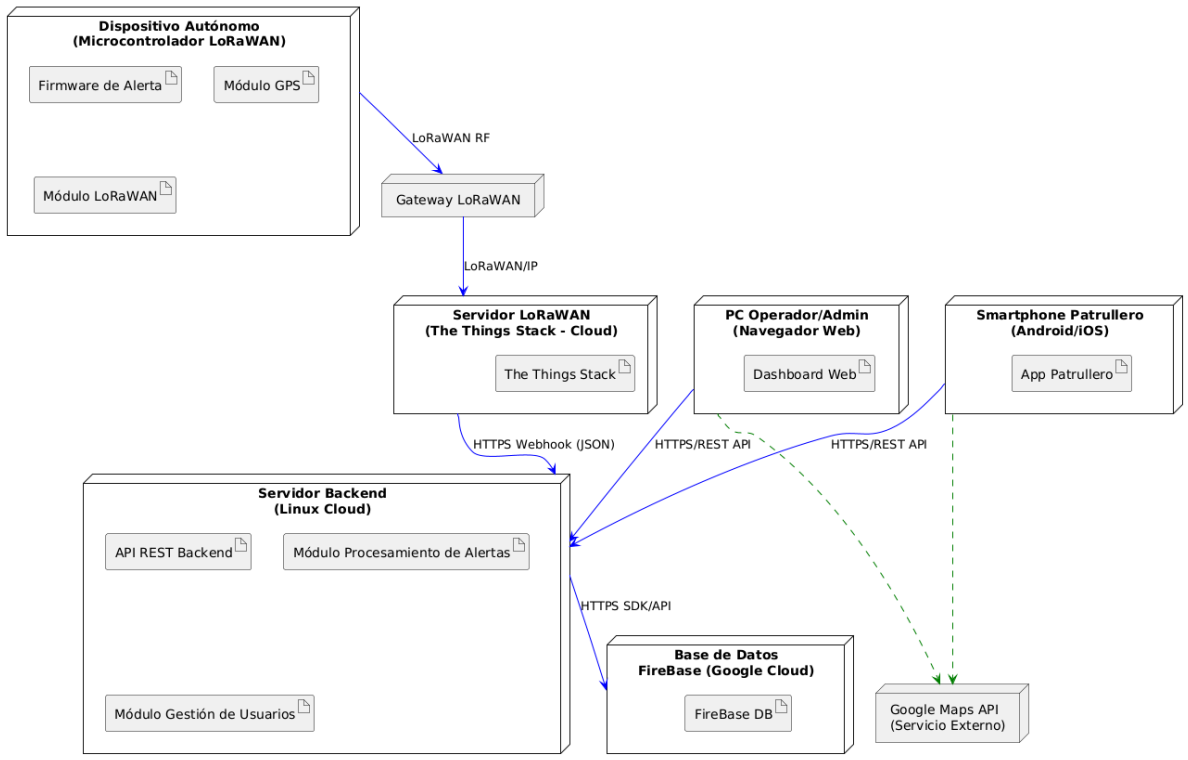
## 3.4. Vista de procesos

### 3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)



### 3.5. Vista de Despliegue (vista física)

#### 3.5.1. Diagrama de despliegue




https://drive.google.com/file/d/1f5_Vn0NR-CGB4JAm58xlfFNsxgm-Cxnr/view?usp=sharing

## 4. Atributos de calidad de software

El sistema de alerta ha sido diseñado considerando diversos atributos de calidad de o implica alta disponibilidad de los servicios backend y de la base de datos, así como resistencia a fallos en la transmisión LoRaWAN y tolerancia a errores software, los cuales se reflejan directamente en sus requerimientos no funcionales. Estos atributos son fundamentales para garantizar que el sistema no solo cumpla con su propósito funcional, sino que también sea confiable, eficiente, seguro y sostenible a largo plazo.

Fiabilidad:
El sistema debe garantizar un funcionamiento continuo y sin fallos, especialmente en situaciones de emergencia. Están los dispositivos. La fiabilidad se asegura mediante

mecanismos de reintentos automáticos, monitoreo del estado del sistema y la robustez de los componentes electrónicos seleccionados para el dispositivo autónomo.

Rendimiento:
Uno de los principales atributos de calidad es la capacidad del sistema para transmitir y procesar alertas en tiempo real, minimizando la latencia entre la activación del botón y la notificación a las unidades de respuesta. Se establece como meta que la alerta se procese y despacha en menos de 15 segundos, y que las ubicaciones de las patrullas se actualicen frecuentemente para una respuesta óptima. Además, el sistema debe soportar la atención de múltiples alertas simultáneamente sin degradar el rendimiento.

Usabilidad:
El sistema está orientado a usuarios en situaciones de alto estrés y vulnerabilidad, por lo que la interfaz del dispositivo debe ser simple e intuitiva. La activación de la alerta se da mediante una única pulsación, y la confirmación visual (LED) asegura al usuario que su solicitud fue enviada. Las aplicaciones móviles y el dashboard web presentan interfaces claras, con mapas e iconos fácilmente interpretables, facilitando la toma de decisiones y la gestión eficiente de emergencias.

Seguridad y Privacidad:
El sistema maneja información sensible, como datos personales y ubicaciones. Por ello, se han implementado medidas de seguridad como la autenticación de dispositivos, el control de acceso basado en roles y el cifrado de datos en tránsito y en reposo. Esto garantiza que solo personal autorizado pueda acceder a la información, cumpliendo con la legislación vigente sobre protección de datos personales en Perú.

Mantenibilidad y Escalabilidad:
La arquitectura modular del sistema, junto con una clara estructuración y documentación del código, facilita su mantenimiento y la incorporación de nuevas funcionalidades a futuro. Además, el uso de tecnologías escalables en la nube permite que el sistema pueda crecer en número de usuarios y dispositivos sin afectar su desempeño, adaptándose a nuevas necesidades o a un despliegue en mayor escala.

Portabilidad:
El sistema ha sido diseñado para ser portable y adaptable a diferentes entornos. La elección de tecnologías abiertas y multiplataforma (como Flutter para la aplicación móvil y React para el dashboard web) facilita su implementación en distintos tipos de

dispositivos y sistemas operativos, asegurando una experiencia homogénea para todos los usuarios.

**Escenario de Funcionalidad**