

Universidad Nacional Autónoma de México

Facultad de Ciencias (2025-2)



# Ingeniería de Software

## Elaboración del Documento de Requerimientos del Proyecto

Martínez Martínez Geovani 320141384  
Martínez Jiménez Maitreyi 320099773  
Ortiz Menez Victor Gael 320101124  
Flores Linares Oscar Daniel 320208591  
Tinoco Miguel Laura Itzel 316020189

15 de Febrero de 2025

# 1. Introducción

## 1.1. Propósito del Sistema

El sistema tiene como propósito principal facilitar la participación ciudadana en la identificación y seguimiento de problemas urbanos mediante una plataforma web colaborativa (OGPS-Reports). Esta herramienta permitirá a los ciudadanos reportar, documentar y dar seguimiento a incidentes urbanos como baches, luminarias descompuestas y obstáculos en la vía pública.

## 1.2. Alcance

El sistema abarcará:

- Registro y gestión de incidentes urbanos con geolocalización.
- Documentación fotográfica de incidentes.
- Seguimiento del estado de los incidentes.
- Visualización interactiva de incidentes en un mapa.
- Validación comunitaria de la resolución de incidentes mediante evidencia fotográfica.

## 1.3. Definiciones y Abreviaciones

- **OGPS-Reports:** Online Geospatial Public System - Reports
- **Incidente:** Cualquier problema en la infraestructura urbana que requiera atención.
- **Usuario:** Persona que puede acceder al sistema (puede ser usuario regular o administrador).
- **Administrador:** Usuario con privilegios especiales que puede gestionar reportes, validar o rechazar actualizaciones de estado. También puede gestionar contenido inapropiado y supervisar la calidad de los reportes.
- **Mapa Interactivo:** Representación visual que permite la geolocalización de incidentes.
- **Estado del Incidente:** Representa el progreso de resolución del problema (*Reportado, En proceso, Resuelto*).
- **Prueba:** Una o varias fotos del accidente reportado por el usuario.

# 2. Requerimientos del Sistema

## 2.1. Requerimientos Funcionales

### 2.1.1. RF1: Autenticación de Usuarios

- El sistema permitirá el acceso mediante credenciales únicas (correo electrónico y contraseña).
- Se implementarán dos niveles de acceso:

- **Usuarios regulares:** Pueden reportar incidentes, actualizar estados y validar resoluciones.
- **Administradores:** Tienen acceso a funciones administrativas y gestión de usuarios.
- Las contraseñas deberán cumplir con requisitos mínimos de seguridad:
  - Mínimo 8 caracteres.
  - Al menos una letra mayúscula.
  - Al menos un número.
  - Al menos un carácter especial.
- El sistema bloqueará la cuenta temporalmente después de 3 intentos fallidos de inicio de sesión.
- El sistema mantendrá un registro de todos los intentos de inicio de sesión para fines de auditoría.

### 2.1.2. RF2: Gestión de Usuarios

- El sistema permitirá el registro, edición y eliminación de usuarios.
- Los usuarios podrán iniciar sesión de forma segura.
- Los usuarios podrán editar su perfil (Alias y Contraseña).

### 2.1.3. RF3: Registro de Incidentes

- Los usuarios podrán marcar la ubicación exacta del incidente en un mapa.
- Se permitirá subir hasta 5 fotografías por incidente.
- Los usuarios deberán categorizar el tipo de incidente (baches y desperfectos, luminaria, accidentes vehiculares, obstáculos).
- Se requerirá una descripción detallada del problema (límite 500 caracteres).
- El sistema registrará automáticamente la fecha y hora del reporte.

### 2.1.4. RF4: Actualización de Incidentes

- Cualquier usuario podrá actualizar el estado de un incidente.
- Se requerirá evidencia fotográfica para marcar un incidente como resuelto.
- Se mantendrá un historial de actualizaciones.
- Se notificará al creador del reporte sobre las actualizaciones.

### 2.1.5. RF5: Visualización de Incidentes

- Mostrar incidentes en un mapa interactivo.
- Filtrar incidentes por categoría, estado y fecha.
- Permitir búsqueda por dirección o coordenadas.

### **2.1.6. RF6: Validación Comunitaria de Resolución**

- Cuando un usuario marca un incidente como resuelto, este no se actualizará automáticamente.
- Otros usuarios podrán confirmar o disputar la resolución.
- Si hay disputas, el incidente permanecerá .<sup>en</sup> revisión” hasta que un administrador lo valide.
- Se notificará a los usuarios involucrados sobre el resultado de la validación.

## **2.2. Requerimientos No Funcionales**

### **2.2.1. RNF1: Seguridad**

- Implementación de autenticación.
- Uso de autenticación segura para evitar modificaciones no autorizadas.
- Encriptación de datos sensibles (como contraseñas).
- Validación de inputs para prevenir inyecciones.
- Control de acceso basado en roles.
- Política de grupo:
  - No todos los grupos tienen acceso a la base de datos.
  - Solo usuarios registrados pueden registrar incidentes.
  - Solo administradores pueden eliminar cualquier incidente.
  - Solo el usuario dueño del incidente puede borrarlo.

### **2.2.2. RNF2: Rendimiento**

- Tiempo de respuesta máximo de 3 segundos.
- Soporte para 100 usuarios simultáneos.
- Optimización de imágenes automática.

### **2.2.3. RNF3: Disponibilidad**

- El sistema deberá estar disponible al menos el 99 % del tiempo.
- Recuperación ante desastres.

### **2.2.4. RNF4: Usabilidad**

- No se puede modificar un incidente sin pruebas.
- Diseño intuitivo y accesible.
- Navegación simple con número reducido de pasos para completar tareas.

### 3. Casos de Uso

#### 3.1. CU1: Registro de Usuario

**Actor Principal:** Usuario nuevo

##### Flujo Principal

1. Usuario selecciona “Registrarse”.
2. Sistema muestra formulario de registro.
3. Sistema informa al usuario el registro fue exitoso.

##### Flujos Alternativos

- **Correo electrónico ya registrado:** Sistema muestra un mensaje de error.
- **Contraseña débil:** Sistema solicita una contraseña que cumpla con los requisitos de seguridad.

#### 3.2. CU2: Registro de Nuevo Incidente

**Actor Principal:** Usuario registrado

##### Flujo Principal

1. Usuario selecciona “Reportar Incidente”.
2. Sistema muestra mapa interactivo.
3. Usuario marca ubicación del incidente.
4. Usuario completa formulario con detalles.
5. Usuario sube fotografías.
6. Sistema valida y almacena el incidente.
7. Sistema confirma registro exitoso.

##### Flujos Alternativos

- **Descripción detallada:** Habrá un campo extra en el registro de incidentes en donde si el usuario lo desea podrá agregar una descripción más detallada con texto.
- **Fallo en carga de imágenes:** Sistema permite reintento.

#### 3.3. CU3: Actualización de Estado de Incidente

**Actor Principal:** Usuario registrado

### **Flujo Principal**

1. Usuario selecciona incidente existente.
2. Usuario elige “Actualizar Estado”.
3. Usuario selecciona nuevo estado.
4. Usuario sube evidencia fotográfica.
5. Sistema valida actualización.
6. Sistema notifica al creador original.

### **Flujos Alternativos**

- **Evidencia insuficiente:** Sistema rechaza actualización.
- **Error en procesamiento:** Sistema mantiene estado anterior.

## **3.4. CU4: Visualización de Incidentes en Mapa**

**Actor Principal:** Usuario registrado

### **Flujo Principal**

1. Usuario accede a vista de mapa.
2. Sistema carga incidentes cercanos.
3. Sistema actualiza markers en mapa.
4. Usuario selecciona incidente específico.
5. Sistema muestra detalles completos.

## **3.5. CU5: Validación Comunitaria de Resolución**

**Actor Principal:** Usuario registrado

### **Flujo Principal**

1. Usuario revisa incidente marcado como resuelto.
2. Usuario verifica evidencia fotográfica.
3. Usuario confirma o disputa resolución.
4. Sistema registra validación.
5. Sistema actualiza estado según consenso.

## Flujos Alternativos

- Disputa de resolución: Sistema notifica a administrador.
- Evidencia contradictoria: Sistema mantiene estado en revisión.

## 4. Requisitos de Hardware y Software

### 4.1. 4.1 Tecnologías Seleccionadas

#### Frontend

- **React:** Seleccionado por su robusto ecosistema y capacidad para crear interfaces dinámicas.
- **Mapas:** Uso de API como Google Maps, OpenStreetMap o Leaflet.

#### Backend

- **Kotlin:** Seleccionado por su seguridad de tipos y compatibilidad con Java.
- **Spring Boot:** Framework para desarrollo rápido y seguro.
- **PostgreSQL:** Base de datos elegida por:
  - Manejo de grandes cantidades de datos.
  - Buen rendimiento de consultas.
  - Código abierto y gratuito.
  - Herramientas para hacer copias de seguridad fácilmente.

**Permitir que usuarios sin registrarse usen la app** pero solo podrían visualizar los incidentes, no podrían registrar un nuevo incidente ni actualizar el estado del incidente.