



# Universidad Nacional Autónoma de México

## Facultad de Ciencias

Ciencias de la Computación  
Semestre 2025-2

Ingeniería de Software

---

### Requerimientos del proyecto

---

BERNAL ESQUIVEL MARIANA MORAYMA	320298787
COMAS CASTAÑEDA MAURICIO SANTIAGO	320215988
JACOME DELGADO ALEJANDRO	320011704
JÍMENEZ SÁNCHEZ EMMA ALICIA	320046155
ROBLEDO RAMÍREZ ISAAC	320140655

KARLA SOCORRO GARCÍA ALCÁNTARA	PROFESOR
DULCE JULIETA MORA HERNÁNDEZ	AYUDANTE
LUIS GERARDO BERNABÉ GÓMEZ	AYUD. LAB.
BRENDA AYALA FLORES	AYUDANTE
DIANA LAURA LUCIANO PANIAGUA	AYUDANTE

## 1. Introducción

### ▷ Propósito

Este documento especifica los requerimientos funcionales y no funcionales del Sistema de Reporte de Incidentes Urbanos (o abreviado: SisRep), que permitirá a cualquier usuario registrado en el sistema reportar incidentes de carácter ciudadano, tales como; el mal funcionamiento de alumbrado público, daños a la vía pública (por ejemplo baches), y problemas similares.

### ▷ Alcance

El Sistema de Reporte de Incidentes Urbanos le dará a los ciudadanos una forma fácil y rápida de reportar incidentes que posiblemente aún no sean detectados por las autoridades locales, lo que permitirá que las autoridades correspondientes se hagan cargo de atender y solucionar los reportes de forma mas rápida.

### ▷ Definiciones

- ▷ SisRep: Sistema de Reporte de Incidentes Urbanos .
- ▷ Usuario: cualquier ciudadano que este registrado en el SisRep.
- ▷ Reporte: registro de cualquier incidente que cumpla con los requisitos de registro colocado en el SisRep.

## 2. Requerimientos del sistema

### ▷ Requerimientos funcionales

#### **RF1: Autenticación de Usuarios**

- ▷ La aplicación permite el acceso de usuarios mediante credenciales únicas: usuario y contraseña.
- ▷ Hay autenticación para acceder como usuario (ciudadano) o administrador.

#### **RF2: Gestión de Usuarios**

- ▷ Los usuarios podrán crear, editar y eliminar su perfil.
- ▷ Los administradores podrán eliminar las cuentas de los usuarios y ver sus datos. A su vez estos tendrán su propia cuenta.

#### **RF3: Visualización de Incidentes**

- ▷ Los incidentes tendrán 3 estados: reportado, en revisión y resueltos
- ▷ Los incidentes se verán en el mapa interactivo según su ubicación, estos tendrán un icono correspondiente a su tipo.
- ▷ Los usuarios pueden hacer clic en el ícono de un incidente y ver detalles sobre este.

#### **RF4: Gestión de Incidentes**

- ▷ El sistema debe permitir a los usuarios registrar un incidente, para esto se pedirá la siguiente información obligatoriamente:
  - ▷ Tipo del incidente, esta será una selección de los tipos predeterminados: baches, luminarias descompuestas, obstáculos en la vía pública, accidentes automovilísticos, otro.
  - ▷ Descripción del incidente.
  - ▷ Fotos de prueba del incidente.
  - ▷ Ubicación, esta se podrá registrar manualmente (de forma escrita o seleccionando el punto en el mapa), o utilizando la ubicación en tiempo actual del usuario.
- ▷ Usuarios que mande pruebas de dicho incidente esté resuelto, el administrador podrá decidir, con ese o más reportes si cambiar el estado del incidente como resuelto.
- ▷ Los incidentes se borrarán después de cierto tiempo cuando ya estén en el estado resuelto.
- ▷ Los administradores deben recibir notificaciones de nuevos incidentes registrados.

#### **RF5: Filtro de incidentes**

- ▷ Los usuarios podrán filtrar los incidentes en el mapa por su estado (Reportado, en revisión, resuelto) o por su tipo, fecha o estado.

#### **▷ Requerimientos no funcionales**

##### **Seguridad:**

Definición de protocolo http, acceso restringido a la base de datos, protección de la rama main, los usuarios que reporten incidentes deben estar obligatoriamente registrados en la aplicación, solo administrador puede eliminar y cambiar el estado de cualquier incidente.

##### **Usabilidad:**

Regulación de contenido, interfaz intuitiva y amigable con el usuario.

##### **Rendimiento:**

El sistema deberá soportar al menos 70 usuarios.

Los datos de incidentes son actualizados en tiempo real.

El procesamiento y subida de incidentes tarda 5 segundos en promedio.

Filtrar incidentes por tipo o tiempo demora aproximadamente de 10 segundos.

Encontrar incidentes cercanos tarda alrededor de 20 segundos.

##### **Disponibilidad:**

El sistema deberá estar disponible el 99 % del tiempo.

### **3. Casos de uso**

▷ **CU1: Reporte de Accidentes:**

**Actores:** Ciudadano

**Flujo Principal:**

- ▷ Abre la aplicación web.
- ▷ Le damos click en el botón de “+” para poder agregar el accidente.
- ▷ Llenamos la información requerida sobre el accidente.
- ▷ Se deberá de indicar correctamente la dirección del accidente.
- ▷ Cuando se termine de llenar la información, se deberá de presionar agregar el accidente.
- ▷ El accidente deberá de aparecer en el mapa.

▷ **CU2: Gestión y Seguimiento de Reportes:**

**Actores:** Administradores y Autoridades

**Flujo Principal:**

- ▷ Un administrador entra a la aplicación.
- ▷ Debe de acceder a la sección correspondiente de los reportes.
- ▷ Filtra los reportes por grado de importancia y tiempo de publicación.
- ▷ Puede actualizar el status del reporte: *en trámite*, *en proceso* y *solucionado*

▷ **CU3: Estadísticas de Accidentes Automovilísticos:**

**Actores:** Ciudadano, Autoridades y Administradores

**Flujo Principal:**

- ▷ Entrar a la aplicación
- ▷ Ir al apartado de estadísticas
- ▷ Poder encontrar una gráfica la cuál pueda describir cuantos accidente ha habido en la semana, mes y año.
- ▷ Salir de la aplicación.

▷ **CU4:Notificación de Accidentes cercanos:**

**Actores:** Ciudadano

**Flujo Principal:**

- ▷ Cada que ocurra una accidente en una vialidad o calle cerca de la ubicación del usuario ó en avenidas importantes, se mandará una notificación informando al usuario para que tome sus precauciones.



## 4. Requisitos de Hardware y Software

### ▷ Hardware

Para poder almacenar la Base de Datos, y alojar el sitio web.

Laptop de 16 RAM, y procesador 6 cores (mínimo) para alojar los servicios y que estos funcionen de manera óptima.

Bases de datos almacenadas en un disco de estado sólido 20 GB (recomendable).

### ▷ Software

- ▷ Frontend: React

- ▷ Backend: Kotlin / Java.

- ▷ Manejo de base de datos: PostgreSQL o SQLite.

- ▷ Mapas: API de Google Maps

- ▷ Herramientas a utilizar: Spring Boot, Notion.