

# Análisis De Requerimientos

Aldo Javier Ángeles Sánchez

Arlet Pinacho Báez

Juvenal Guzmán Condado

Ricardo Iván Martínez Cano

Ricardo Maximiliano Victoria Morales

11/2/25

## 1. Introducción

El objetivo de este documento es definir claramente los requerimientos funcionales y no funcionales necesarios para el desarrollo de un sistema destinado a la gestión de incidentes urbanos. Durante el desarrollo del proyecto se busca llevar a cabo un método incremental que permita entregar en cada fase resultados parciales que puedan ser validados por el cliente.

### 1.1. Propósito del sistema.

Se busca desarrollar una página web para el registro y gestión de incidentes urbanos por medio de la participación activa de los ciudadanos. Esto con el fin de a largo plazo hacer uso de los datos recopilados para la optimización de la resolución y seguimiento brindado por las autoridades gubernamentales.

### 1.2. Alcance del sistema.

Dado que el proyecto busca fomentar la participación de la ciudadanía, la funcionalidad prioritaria será permitir a los usuarios (ciudadanos de la CDMX) registrarse en la plataforma web, lo que les habilitará para crear, actualizar y visualizar incidentes relacionados con la vía pública. Los registros de incidentes incluirán información esencial como la ubicación y una descripción del problema, permitiendo un seguimiento adecuado.

### 1.3. Definiciones y abreviaciones.

- Salting: Técnica criptográfica que consiste en añadir un valor aleatorio único (llamado salt) a cada contraseña antes de calcular su hash.
- Hashing: Forma de cifrar las contraseñas a través de una función hash criptográfica. Esto para tener más seguridad.
- Estado de un accidente: Puede ser Reportado", .<sup>En</sup> proceso.º Resuelto". Se refiere a etiquetas asociadas a los accidentes registrados.

- MojaraDrive: Nombre de la página web a implementar.

## **2. Requerimientos del Sistema**

### **2.1. Requerimientos funcionales**

- Creación de cuentas para los usuarios de la página.

El sistema debe permitir a cada nuevo usuario crear una cuenta que lo identifique como ciudadano y resguardar su información para posteriores ingresos a la plataforma.

- Registro de incidentes.

Se debe permitir a los usuarios llevar un registro de los incidentes que reportan. Cada incidente deberá incluir la ubicación, al menos una fotografía, un estado (Reportado, En proceso, Resuelto) y una descripción. Además, por cuestiones de administración de la información, se debe contar con una forma de clasificar los incidentes y poder identificar al usuario que los reporta.

- Clasificación de incidentes.

Además de tener asociado un estado (Reportado, En proceso, Resuelto), cada incidente debe clasificarse por tipo (por ejemplo: bache, luminaria descompuesta, obstáculo en vía pública, entre otros).

- Actualización del estado de los incidentes.

Es importante que el sistema permita a los usuarios actualizar el estado de los incidentes (aunque no hayan sido ellos quienes los registraron), siempre y cuando adjunten al menos una fotografía que respalde el cambio de estado.

- Visualización y filtrado de incidentes.

La página debe permitir a los usuarios marcar la ubicación de los incidentes que reportan en un mapa interactivo. Además, todos los usuarios deben poder ver los incidentes reportados en la página y filtrarlos por tipo y estado.

### **2.2. Requerimientos no funcionales**

- Delimitación de roles

El sistema no debe de permitir a un usuario que no cuenta con los permisos necesarios realizar operaciones que solo usuarios privilegiados (Administradores, Etc.) puedan realizar.

- Delimitación de usuarios

El sistema no puede permitir que un usuario que no es el dueño de un reporte pueda realizar acciones como la eliminación del reporte, cambio de ubicación de este, cambio de descripción del reporte, cambio de clasificación del incidente.

- Inicio de sesión seguro

El sistema debe de usar tecnología de inicio de sesión segura, ya sea utilizar un sistema de inicio de sesión popular (Inicio de sesión con cuenta de Google, Facebook) o bien utilizar un sistema de inicio de sesión desarrollado propiamente. El cuál no permita el acceso a cuentas no registradas, no permita duplicidad de cuentas, es decir, una sola

cuenta puede estar asociada solo con un correo electrónico. Por último el sistema debe de ser capaz de solo aceptar correos electrónicos válidos.

- **Contraseñas seguras**

Al momento de hacer el registro de un nuevo usuario, el sistema debe de revisar si la contraseña introducida por el usuario cuenta con la mínima seguridad indispensable (Por lo menos una letra mayúscula, Un caracter especial, Por lo menos un número, etc). En caso de no contar con esta información, se debe de indicar al usuario que debe de corregir antes de poder registrar a su cuenta.

- **Validación de campos**

El sistema debe de validar los campos que el usuario debe de introducir para evitar situaciones de seguridad como la inyección de código.

- **Protección de contraseñas**

Las contraseñas de usuario deben de estar guardadas con un estándar de cifrado usado comúnmente y seguro, de la misma manera se debe de usar un proceso de "salting" para proveer más seguridad a los usuarios.

### **3. Casos de Uso**

#### **CU1: Registrar un Incidente**

**Actores:** Usuario

**Flujo Principal:**

1. El usuario inicia sesión en la aplicación web MojarraDrive.
2. Selecciona la opción Reportar incidente".
3. Ingresar la ubicación del incidente en el mapa interactivo.
4. Adjunta una o más fotografías del incidente.
5. Clasifica el incidente según sus características
6. Escribe una breve descripción del incidente.
7. Confirma el registro del incidente.
8. El sistema guarda y verifica la información.
9. Una vez se haya confirmado la validez del incidente se muestra en el mapa de la aplicación.

**Flujo Alternativo:**

- Si se detecta algún error en la información o falta información obligatoria, el sistema notificará al usuario y solicitará completar o corregir los campos faltantes. De no completarlo correctamente, el incidente no se mostrará en el mapa.

## **CU2: Actualizar Estado de un Incidente**

**Actores:** Usuario

### **Flujo Principal:**

1. El usuario inicia sesión en la aplicación web MojarraDrive.
2. Busca un incidente en el mapa o en la lista de reportes.
3. Selecciona el incidente y elige la opción "Actualizar estado".
4. Agrega el nuevo estado del incidente.
5. Adjunta evidencia fotográfica del nuevo estado del incidente.
6. Confirma la actualización del estado.
7. El sistema cambia el estado del incidente y almacena el historial de modificaciones.

### **Flujo Alternativo:**

- Si el usuario no adjunta pruebas fotográficas, el sistema rechazará la actualización del estado.

## **CU3: Visualizar Incidentes**

**Actores:** Usuario

### **Flujo Principal:**

1. El usuario accede a la aplicación web MojarraDrive.
2. Visualiza un mapa interactivo con incidentes reportados.
3. Aplica filtros por tipo de incidente, estado y lugar.
4. Selecciona un incidente para ver detalles y fotografías adjuntas.

## **CU4: Administración de Incidentes**

**Actores:** Administrador

### **Flujo Principal:**

1. El administrador inicia sesión en la aplicación web MojarraDrive.
2. Accede al panel de administración.

3. Filtra incidentes por tipo o estado.
4. Modifica, elimina o marca incidentes como revisados.
5. Genera estadísticas sobre incidentes registrados.

## 4. Requisitos de Hardware y Software

### 4.1. Hardware

- Servidor con al menos 8 GB de RAM y procesador Intel® Core™ i5-6300U, o alguno más potente. Para tener buena velocidad y rendimiento.
- Sistema con almacenamiento SSD o HDD para alojar a la base de datos. Esto para tener más flexibilidad.

### 4.2. Software

- **Lenguajes de programación:** La implementación se realizará mediante Kotlin debido a que permite que la escritura del código sea más intuitiva conservando las ventajas de Java.
- **Base de datos:** Se usará PostgreSQL debido a que es la herramienta que conoce mayormente el equipo de desarrollo.
- **Framework web:** Se utilizará Spring Boot debido a su facilidad de uso y su extensa comunidad que deriva en un mayor soporte. Además, se hará uso de Angular.

## 5. Consideraciones finales

Este documento engloba el análisis de requerimientos de nuestro sistema de gestión de incidentes urbanos.

Debido a que se está usando el modelo incremental, nos reservamos el derecho de modificar los requerimientos posteriormente. Aunque claramente sin afectar gravemente la aplicación y las posibles implementaciones ya hechas durante alguna iteración.

Cualquier cambio debe ser aprobado por los interesados.