Universidad Nacional Autónoma de México
Facultad de Ciencias
Ciencias de la Computación
Semestre 2025-2
Ingeniería de Software

Documentación SISREP: Releases v2.0.X (lastest)

Bernal Esquivel Mariana Morayma	320298787
Comas Castañeda Mauricio Santiago	320215988
Jacome Delgado Alejandro	320011704
Jímenez Sánchez Emma Alicia	320046155
Peña Villegas Diego Eduardo	321260682
Robledo Ramírez Isaac	320140655

El Consejo del Backend

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

23 de mayo de 2025

1. Ambiente de implementación

Release	Repositorio	Link
v2.0.0 (or lastest)	Front-end	Repositorio-frontend
v2.0.0 (or lastest)	Back-end	Repositorio-backend

Concepto	Herramienta	Versión
Lenguaje backend principal	Kotlin	1.9.25
Lenguaje backend auxiliar	Java	JDK-21
Biblioteca para la aplicación web	React	19.0.0
Entorno de ejecución frontend	Node.js	22.14.0
Gestion de dependencias frontend	npm	11.2.0
IDE lógico del negocio	SpringBoot	3.4.4
IDE interfaz de usuario	IntelliJ	2024.3.5
Lógica en vistas	Typescript (vía npm)	5.8.2
Herramienta para pop-ups	Sweetalert2 (vía npm)	11.17.2
Herramienta para optimizar paths	react-router-dom	7.4.1
Herramienta de imagenes	react-bootstrap-icons	1.11.5
Componente react para Leaflet	react-leaflet	5.0.0
Componente react para métodos del DOM	react-dom	19.0.0
Componente react para mapbox-gl	react-map-gl	7.4.0
Manejador de base de datos (imagen para docker)	PostgreSQL	17.4.0
Herramienta de Bases de Datos (recomendable)	DBeaver	25.0.1
Herramienta para construcción	Bootstrap	5.3.3
Herramienta para construcción	Vite	5.3.3
API para el uso de mapas	Leaflet	1.9.4
Herramientas para mapas	mapbox-gl	3.10.0
Control versiones	Git	2.49.0
Diagramador UML (usado, no necesario para ejecución)	Drawio	26.1.0
Motor de documentación	Dokka	2.0.0

2. Historial de cambios

Versión	Repositorio	Fecha	Descripción
v1.1.0	Repositorio-backend-api	Mar 25, 2025	Avance incluyendo API funcional, DB lista para uso, algunas restricciones, parte de identidad visual (logotipo y paleta de colores) documentación de API (OpenAPI), de código, y general del sistema.
v1.1.0	Repositorio-frontend	Mar, 25 2025	Avance de vistas (sobre endpoints), funcionales, con ciertas restricciones de formato en formularios y con ciertos avisos de errores marcados.
v1.1.1	Repositorio-backend-api	Mar 31, 2025	Reparación de fallo sobre etiquetas.
v1.1.2	Repositorio-backend-api	Mar 31, 2025	Reparación de fallo sobre archivos duplicados.
v1.1.3	Repositorio-backend-api	Apr 1, 2025	Cambio de versión del DDL y diagramas, optimizada con roles en lugar de dos tablas (no afecta funcionalidad). Se acompaña de cambios en la documentación general del sistema, con nuevos casos de uso, historial de cambios y el versionado nuevo.
v2.0.0	Repositorio-backend-api	Apr 28, 2025	Se buscó cumplir los Casos de Uso 5-8. Se implementaron nuevos endpoints para su uso en lo agregado. Se modificó el `DDL.sql`.
v2.0.0	Repositorio-backend-api	Apr 28, 2025	Se buscó cumplir los Casos de Uso 5-8. Se agregó el servicio de mapas y sus respectivas adiciones a la iteración. Se adicionan ciertos avances respecto a otras funcionalidades.
Not Defined	Repositorio-backend-api	Mayo 23, 2025	Se agregó documentación in-code extensa con KDoc y OpenAPI.
Not Defined	Repositorio-backend-api	Mayo 23, 2025	Se agregó documentación in-code extensa con TSDoc.

3. Objetivo del proyecto

Crear un sistema de software cuyos objetivos sean los siguientes:

- Automatizar la recolección de datos relacionados con los accidentes o obstáculos viales (fecha, hora, lugar, tipo de accidente, etc.) y manejar dicha información para presentarla a manera de mapa para cualquier usuario.
- El sistema tendrá roles de usuario con y sin sesión y administradores.
- Los usuarios no registrados, solo podrán ver el mapa y los incidentes, y claro registrarse.
- Los usuarios registrados pueden también ver el mapa, y además integrar al sistema nuevos accidentes, reportar hoyos, baches o cualquier dificultad que se presente en el camino, deben ser capaces de registrarse con un usuario, correo y contraseña propios y iniciar sesión.
- Los administradores de zona sumado a lo mencionado en el rol anterior, podrán modificar el status de los incidentes, ya sea que estén reportado, en revisión y resuelto.

4. 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.

5. Planteamiento de necesidades

Se implementó un sistema que cumpliera con las funciones de dar de alta un reporte de accidente vial, donde los usuarios registrados van a poder dar de alta en el sistema mediante un reporte, registrando los datos necesarios del accidente o del problema vial, donde los usuarios registrados pueden comentar el status del reporte. Es necesario contar con usuarios registrados con un correo, usuario y contraseña para tener acceso al sistema. Para el desarrollo del sistema es necesario contar con las funcionalidades esenciales descritas en el documento.

6. Glosario de términos

Término	Definición
SISREP	Sistema de Reporte de Incidentes Urbanos.
IDE	Interfaz de Desarrollo de Entorno.
FrontEnd	Parte del proyecto donde los usuarios interactuan.
Back-end	La lógica del proyecto, no visible para el usuario.
Base de Datos	Estructura para almacenar datos.
Dependencias	Relaciones entre componentes para funcionar correctamente.
API	Aplication Programming Interface. Interfaz que permite la comunicación entre distintos sistemas o componentes.
Endpoint	Ubicación digital donde una API recibe solicitudes de recursos en su servidor.

7. Requerimientos del sistema

Requerimientos funcionales

RF1: Autenticación de Usuarios

- La aplicación permite el acceso de usuarios mediante credenciales únicas: usuario y contraseña.
- Hay autentificación 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.
- **Rendimineto:**
  El sistema deberá soportar al menos 70 usuarios.
  Los datos de incidentes son actualizados en tiempo real.
  El procesado 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 de 20 segundos.
- **Disponibilidad:**
  El sistema deberá estar disponible el 99 % del tiempo.

8. Ejecución

Para ejecutar el proyecto, es recomendable tener un contenedor de Docker con la imagen de Postgres, donde con DBeaver se cargará el `DDL.sql`.

Ya hecho eso, es necesario instalar las dependencias necesarias de npm descritas en el ambiente de implementación (esto se puede hacer en la carpeta propia al repositorio `frontend` mencionada más adelante).

Hay que asegurarse que la Base de datos se encuentra activa y corriendo.

Después de ello, verificar en `src/main/aplication.properties` que tanto el puerto, y nombre de la base de datos, como usuario y contraseña correspondan a las propias.

Ahora, se ejecutará el proyecto:

Ya solo queda crear una carpeta para alojar ambos repositorios.

Creamos la carpeta y dentro de ella ejecutamos

```
$ git clone https://github.com/ingenieria-software-7009-2025-2/frontendcouncil-
    frontend.git
$ cd frontendcouncil-frontend
$ npm install
$ npm run dev
$ cd ..
git clone https://github.com/ingenieria-software-7009-2025-2/backendcouncil-api.git
```

Podemos usar IntelliJ, y solo pulsar Run.

Lo cual debería correr sin problemas.

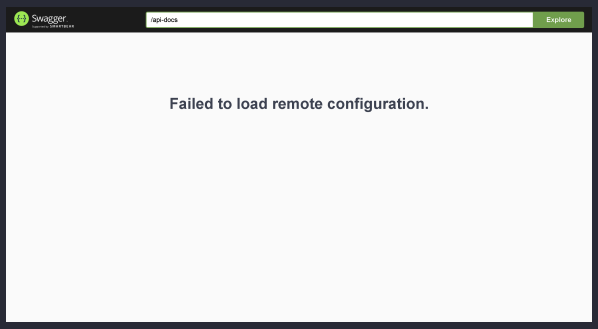
Finalmente queda abrir el navegador y usar dominios `http://localhost:8080/*` para ver la documentación de la API o acceder a sus endpoints, o bien, `http://localhost:5173/` (puertos por defecto).

9. Definición de endpoints

Para documentar los endpoints, utilizamos `OPEN-API v3.0.0`, que al correr el proyecto, podemos checar visitando en el navegador de preferencia

```
http://localhost:8080/api-docs.html
```

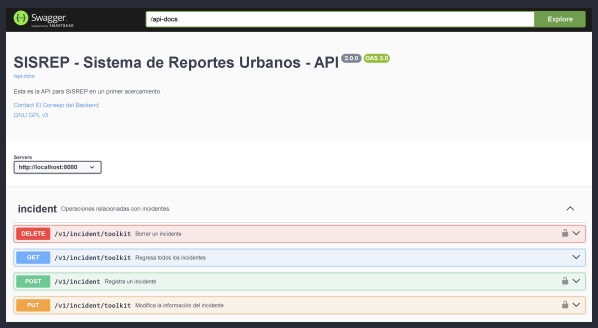
Lo que nos redirigirá a la siguiente página:



Donde en la barra de búsqueda ubicada en la parte superior, colocaremos:

```
/api-docs
```

Lo cual nos llevará a la documentación de los endpoints hasta le momento.



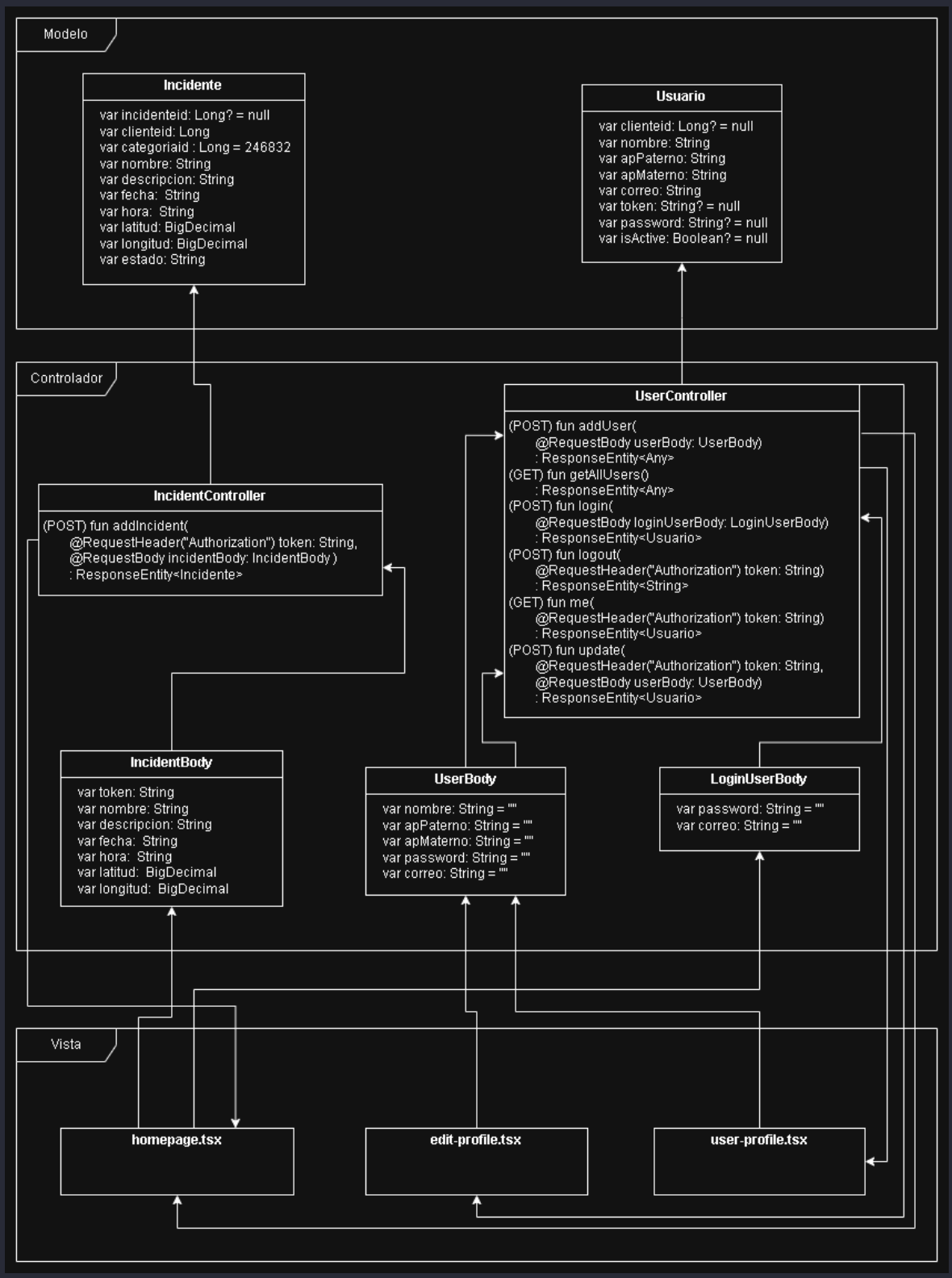
10. Documentación in code

Para el repositorio de la API, se usa KDoc, y se puede visualizar con Dokka.

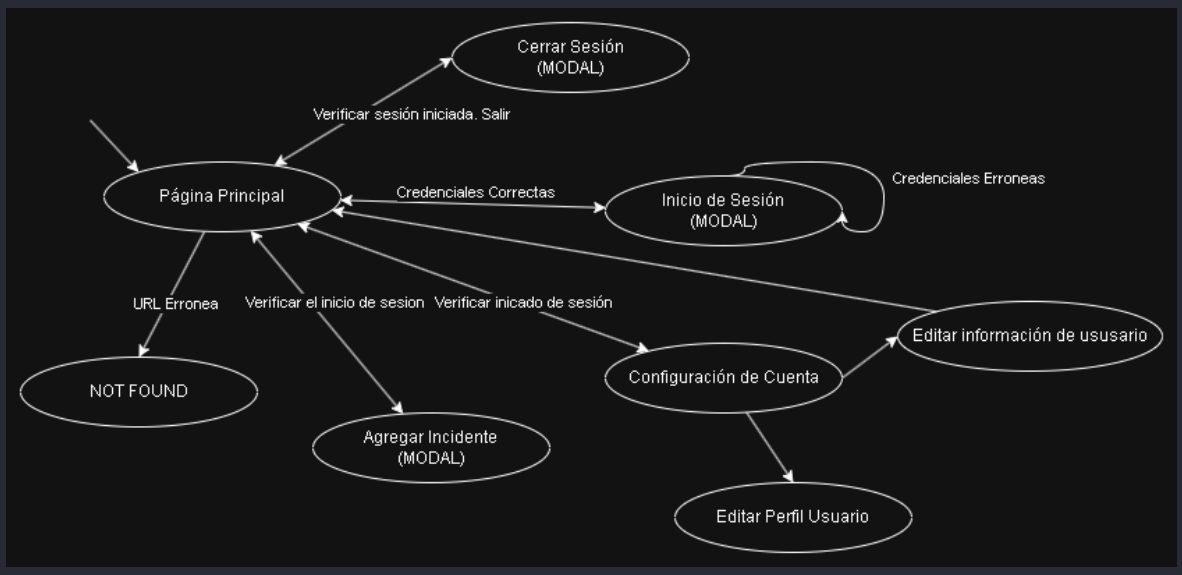
Para visualizar esta documentación en forma de web, es necesario correr el proyecto, y/o correr el *goal* `dokka:dokka` y correr el archivo `target/dokka/index.html`.

Para el repositorio que contiene el front-end, se usa TSDoc, el CSS no ha sido documentado.

11. Diagrama de clases del sistema



12. Diagrama de Navegación del sistema



13. Diagramas para el diseño de la Base de Datos usada en el sistema

Diagrama del Modelo Entidad-Relación

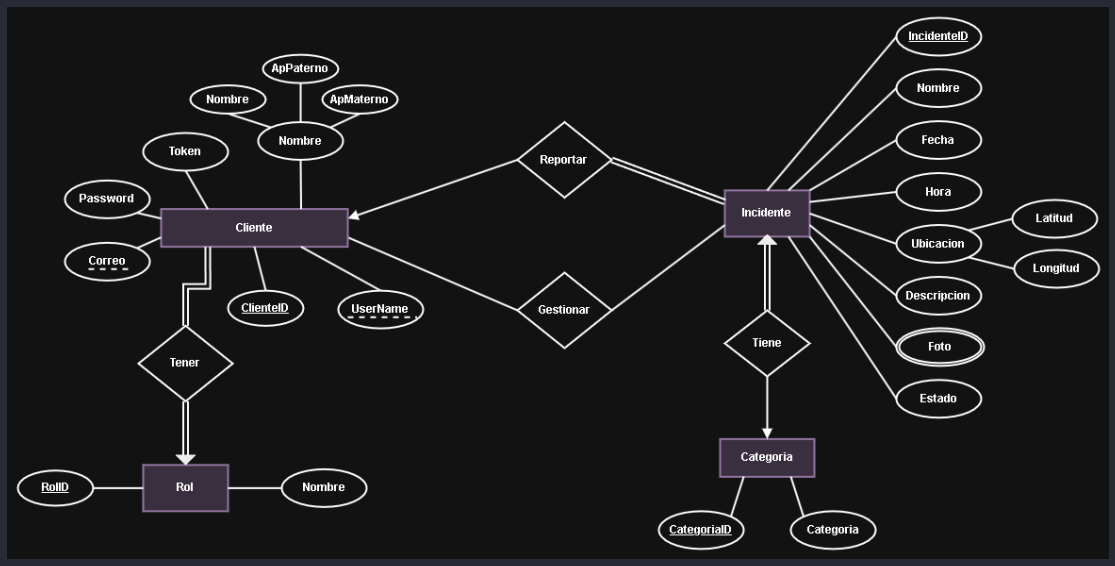
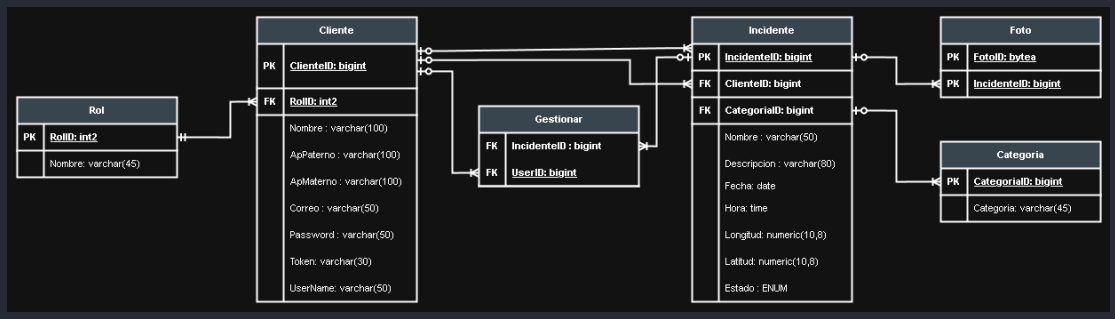


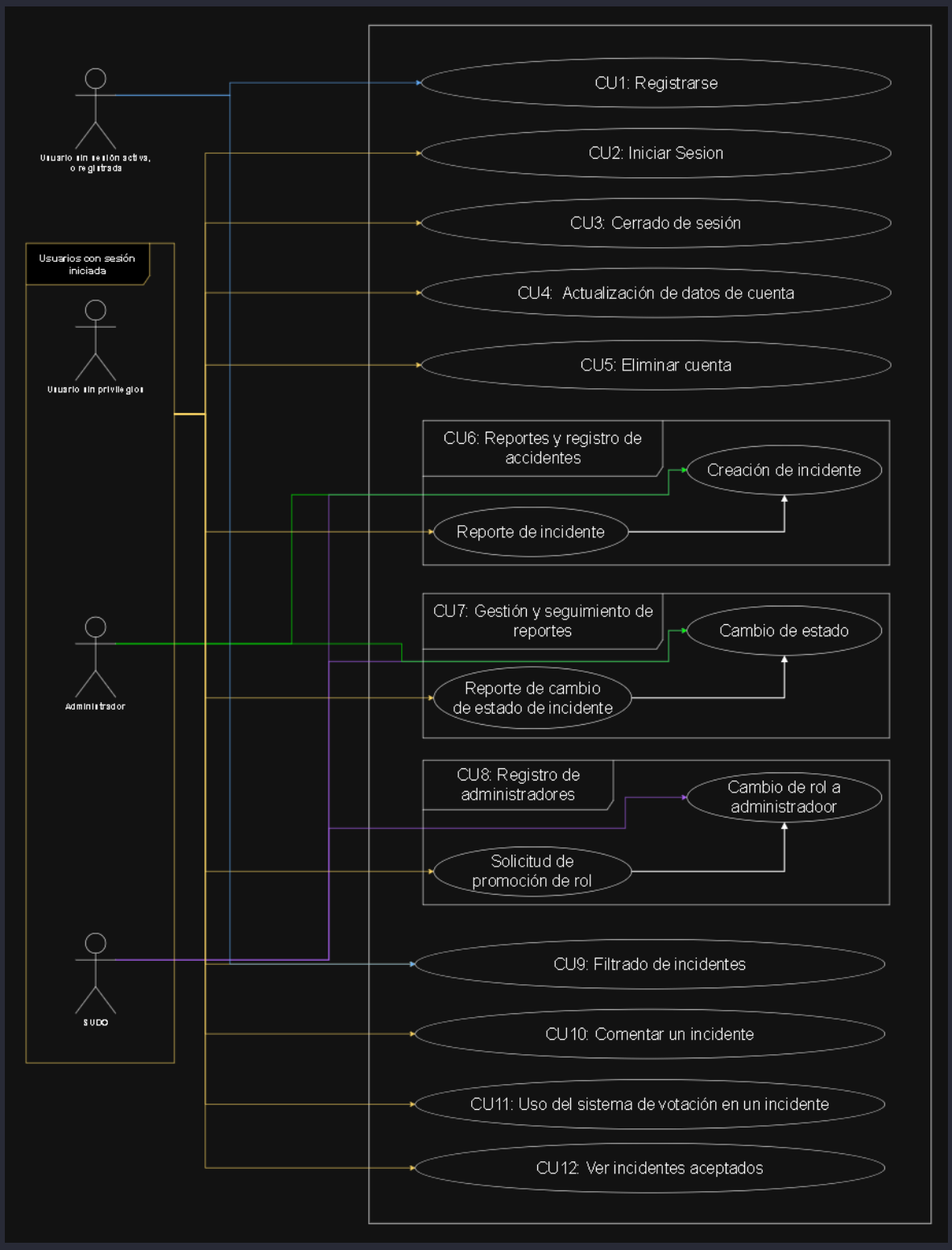
Diagrama del Modelo Relacional



14. Identificación de las funcionalidades del producto de software

Para simplificar este punto, se puede apreciar el Diagrama General de Casos de Uso, donde podemos visualizar las funcionalidades.

Nótese que podemos ver a los actores con los roles del sistema. Donde La primera y más clara clasificación, es si está logeado en una cuenta, lo cual permite el uso de muchas funcionalidades; y después los permisos que el usuario tenga.

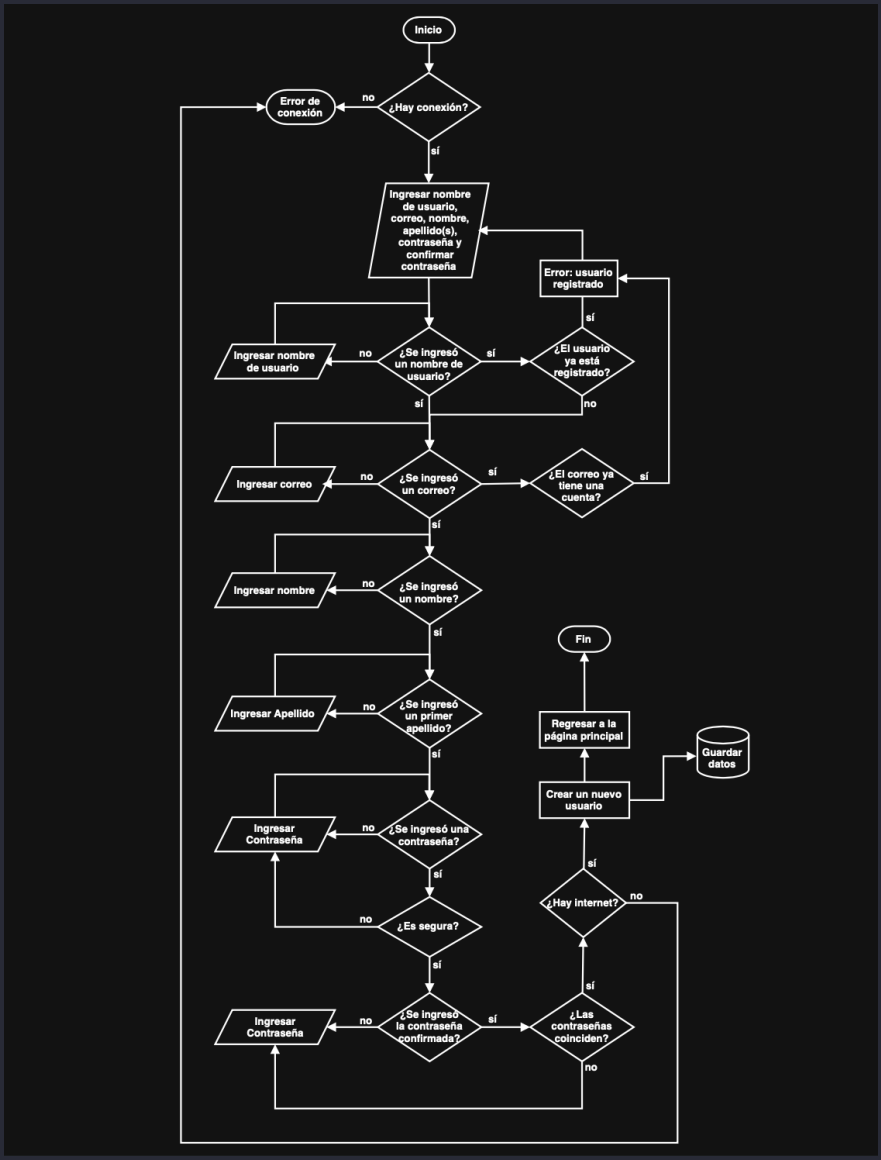

Usuario sin sesión activa, o registrada
Usuarios con sesión iniciada
Usuario sin privilegios
Administrador
SUDO
CU1: Registrarse
CU2: Iniciar Sesion
CU3: Cerrado de sesión
CU4: Actualización de datos de cuenta
CU5: Eliminar cuenta
CU6: Reportes y registro de accidentes
Creación de incidente
Reporte de incidente
CU7: Gestión y seguimiento de reportes
Cambio de estado
Reporte de cambio de estado de incidente
CU8: Registro de administradores
Cambio de rol a administradoor
Solicitud de promoción de rol
CU9: Filtrado de incidentes
CU10: Comentar un incidente
CU11: Uso del sistema de votación en un incidente
CU12: Ver incidentes aceptados

15. Casos de Uso Normales

A partir del diagrama de funcionalidades, se realiza una explicación de una manera más detallada.

CU1: Registro de cuenta

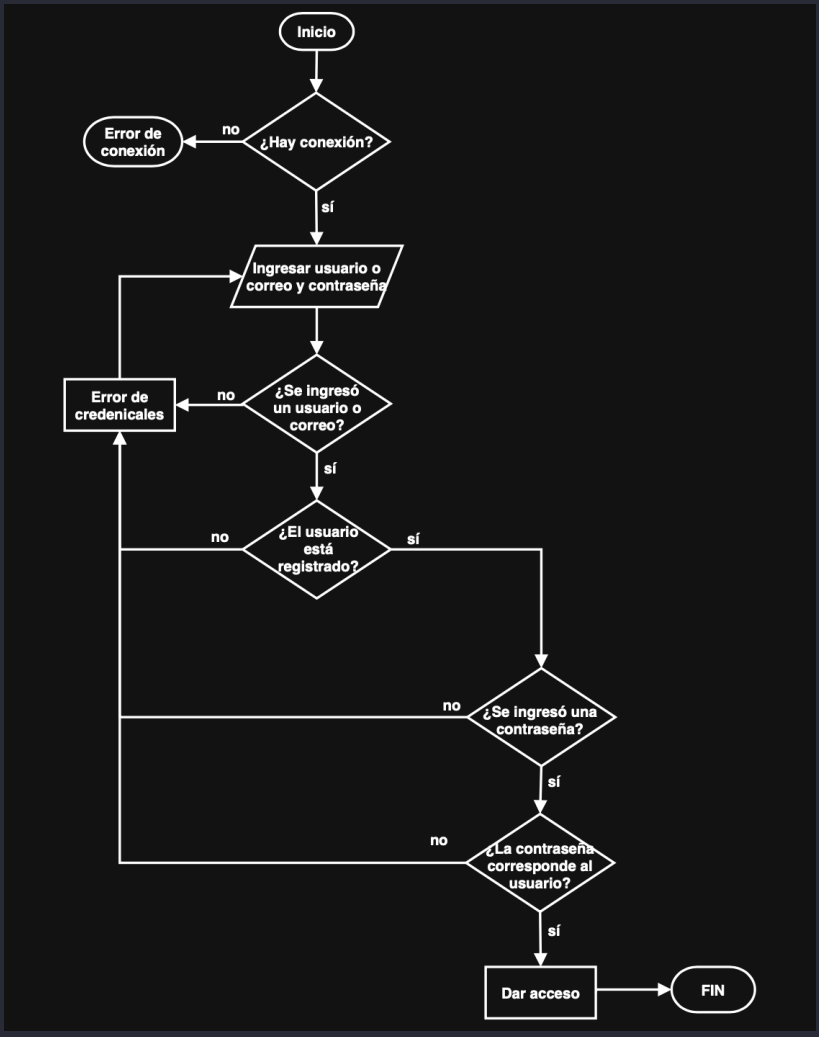
- Descripción
  Los usuarios nuevos al sistema deberán de registrarse con un correo, usuario y contraseña para poder autentificarlos y dar autorización que entren al sistema.
- Precondiciones
  Tener un correo y usuario únicos en el modelo correspondiente. Una contraseña segura (una mayúscula, un número y un carácter especial y al menos 8 de longitud).
- Postcondiciones
  Poder acceder a más funcionalidades como configuración de la cuenta o creación de reportes al sistema.
- Casos de uso alternativos
  - El usuario ya tiene una cuenta registrada, debe de mostrar un mensaje de error y se sugiere iniciar sesión ó bien, si los de campos de contraseña y su confirmación difieren y/o no es una contraseña segura, se indicará un error de que se ingreso datos incorrectos.
  - El formulario contiene campos vacíos o formato inválido, debe de notificar al usuario con mensajes de validación específicos.
- Casos de uso extraordinarios
  - Si no hay internet o no hay luz, los datos no se guardan y deberá de llenarlos otra vez.



CU2: Inicio de sesión

- Descripción
  Permite al usuario entrar a la cuenta ya registrada en el sistema.
- Precondiciones
  Contar con una cuenta registrada e ingresar correo o usuario y contraseña a los campos correspondientes.
- Postcondiciones
  El usuario será capaz de enviar reportes, y levantar un reporte de cambio de estado, o en el caso de tener el rol de administrador, gestionar reportes.

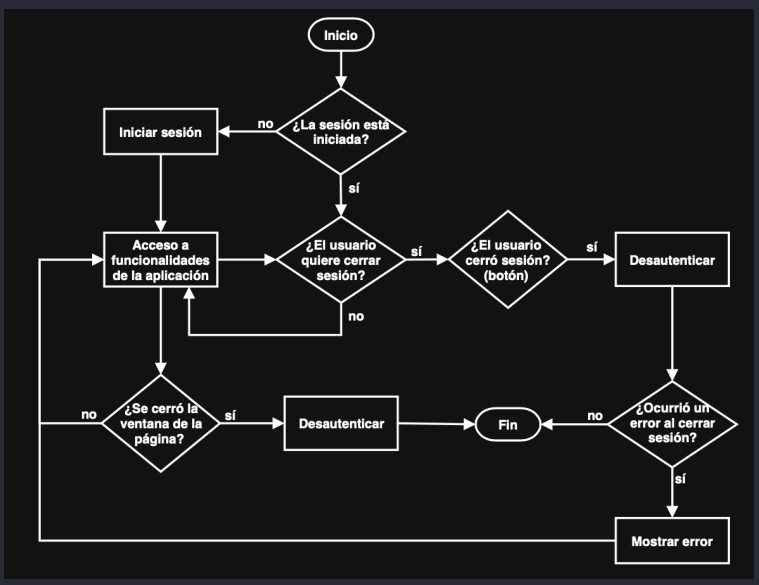
- Casos de uso alternativos
  - Si las credenciales son incorrectas, entonces el sistema muestra un mensaje de error e impide el acceso.
- Casos de uso extraordinarios
  - Si no hay internet no se permite el inicio de sesión.
  - Si hay problemas con la conexión de la base de datos entonces no se permite iniciar la sesión.



CU3: Cerrado de sesión

- Descripción
  Permite terminar y caducar las credenciales con las que tiene acceso al sistema.

- Precondiciones
  Tener una sesión abierta.
- Postcondiciones
  El usuario/administrador no tiene acceso a las funcionalidades del sistema correspondientes.
- Casos de uso alternativos
  - En caso de cerrar la pestaña en la que esta iniciada la sesión se tomara como un cierre de sesión en la aplicación.
- Casos de uso extraordinarios
  - Si hay un error al cerrar sesión, no se cierra, y al solucionarlo, se muestra la sesión iniciada.



CU4: Actualización de datos del usuario

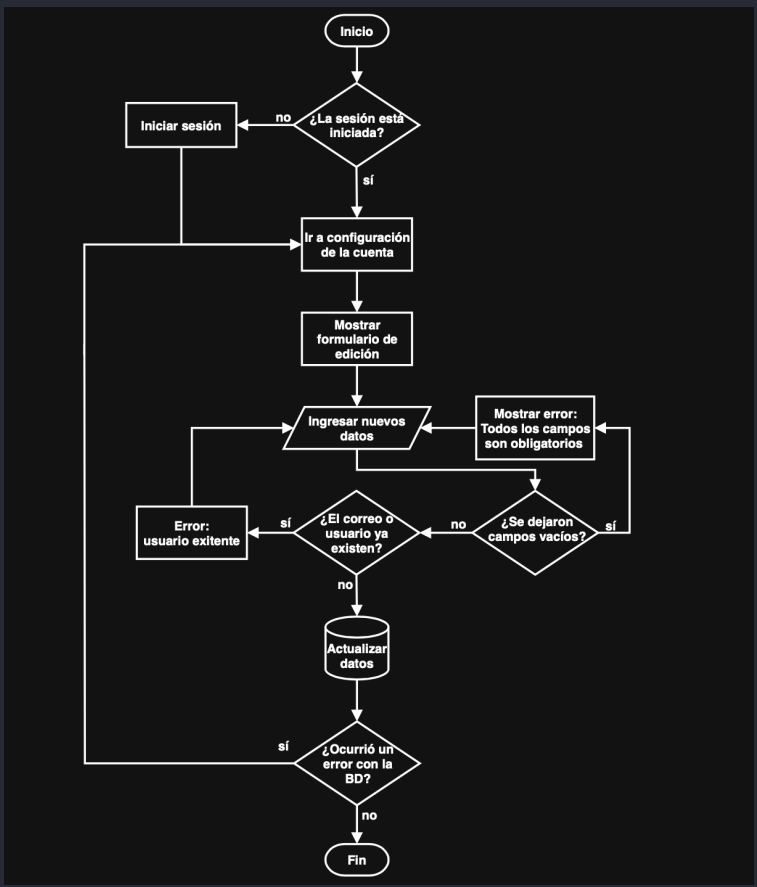
- Descripción
  El usuario/administrador cambia datos como nombre, apellido paterno y materno, usuario, y correo. O bien, contraseña.
- Precondiciones
  Tener una cuenta activa.
- Postcondiciones
  Cambio de datos.

- Casos de uso alternativos

  - Si el nuevo correo/usuario ya existe en el sistema, se muestra un error y se solicita ingresar otro diferente, recontestando el formulario.

- Casos de uso extraordinarios

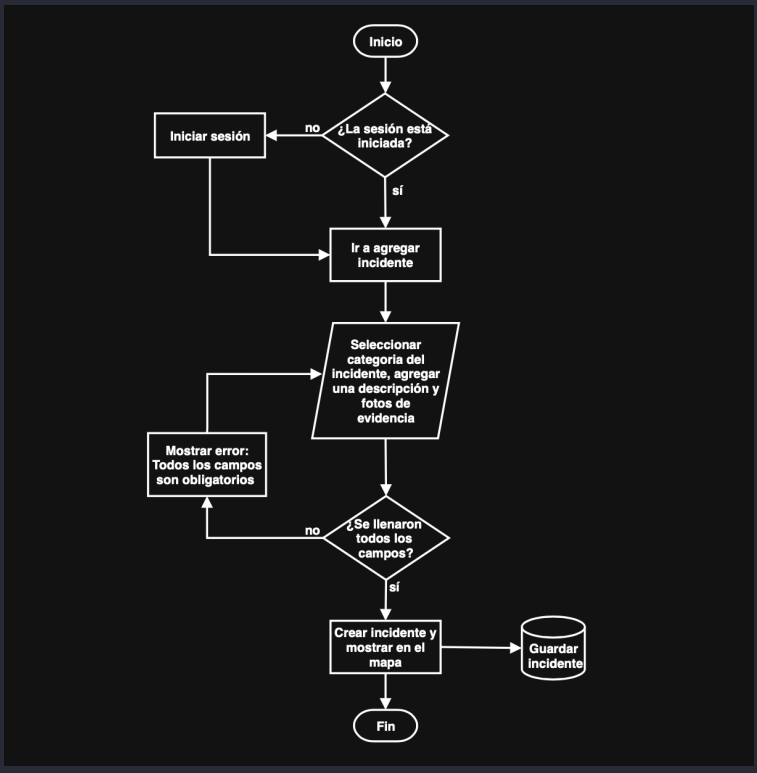
  - Ante cualquier error, se tendrá que volver a enviar el formulario.
  - Si el SUDO, cambia su rol por error, solamente se puede regresar accesando directamente a la DB.



CU5: Reporte de accidentes y registro de incidentes

- Descripción
  Un usuario/administrador envía un reporte (aparecerá como incidente reportado o similares), un administrador recibe dicho reporte y al ver varios del mismo incidente o suficientes pruebas, dado su criterio podrá cambiar el estado del incidente a (en revisión/en progreso o similares).

- Precondiciones
  Un usuario/administrador y un administrador (preferentemente diferente) con cuentas registradas, activas al momento de realizar las acciones correspondientes.
- Postcondiciones
  Agregado de un pin/marca al mapa, representando un incidente con estado en revisión/en progreso.
- Casos de uso alternativos
  - Si el reporte carece de información o datos necesarios, el sistema solicita completar los campos obligatorios.
- Casos de uso extraordinarios
  - Si el servicio de mapas no está disponible, debido a un error del servidor o la falta de conexión, entonces no se podrá enviar la información correcta a la base de datos. Se tendrá que enviar otra vez los datos, cuando el servicio vuelva a ser estable.

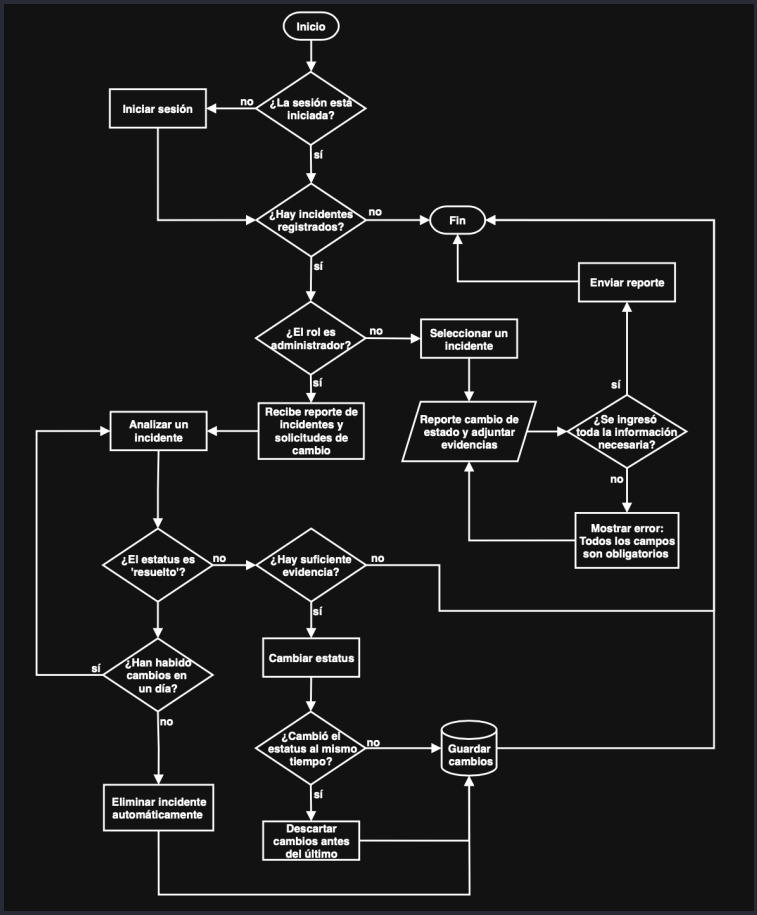


CU6: Gestión y seguimiento de reportes

- Descripción
  Si un usuario se percata que el estado del incidente debe cambiarse (en la vida real), envía un reporte de cambio de estado. Posteriormente un administrador

dadas las pruebas y/o cantidad de reportes, según su criterio decide si cambiar o no el estado del incidente (estado resuelto o similares). Si el incidente tiene el estado de resuelto, la chincheta desaparece del mapa o se mantiene indicando su nuevo status.

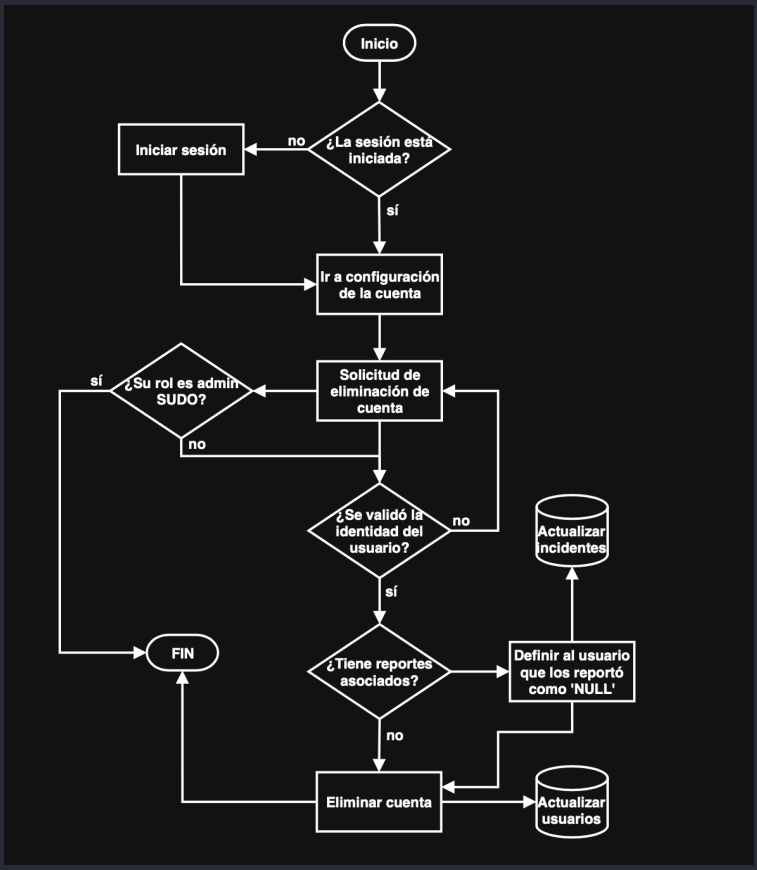
- Precondiciones
  Incidente existente, además de usuario y administrador con cuentas registradas y activas en el movimiento.
- Postcondiciones
  Cambio de estado de dicho incidente.
- Casos de uso alternativos
  - Si hay un conflicto de edición, como dos administradores modifican el estado simultáneamente, entonces el sistema prioriza el último cambio.
  - Si un incidente con estado de como resuelto recibe nuevos reportes, el administrador deberá de checar los reportes para reabrir la revisión.
- Casos de uso extraordinarios
  - Un administrador intenta cambiar el estado de un incidente, pero durante el proceso, la conexión a internet falla abruptamente, ningún dato se modifica, y se tendrán que rehacer dichos cambios.



CU7: Borrado de cuenta

- Descripción
  Usuario/administrador decide eliminar su cuenta.
- Precondiciones
  Usuario con cuenta activa.
- Postcondiciones
  Si el usuario no acepta eliminar la cuenta, se cancela la operación. O en caso contrario, un cerrado de sesión y eliminación de acceso al sistema por la eliminación en la base de datos.
- Casos de uso alternativos
  - Si el usuario/administrador ha creado reportes, estos tendrán una referencia nula..
- Casos de uso extraordinarios

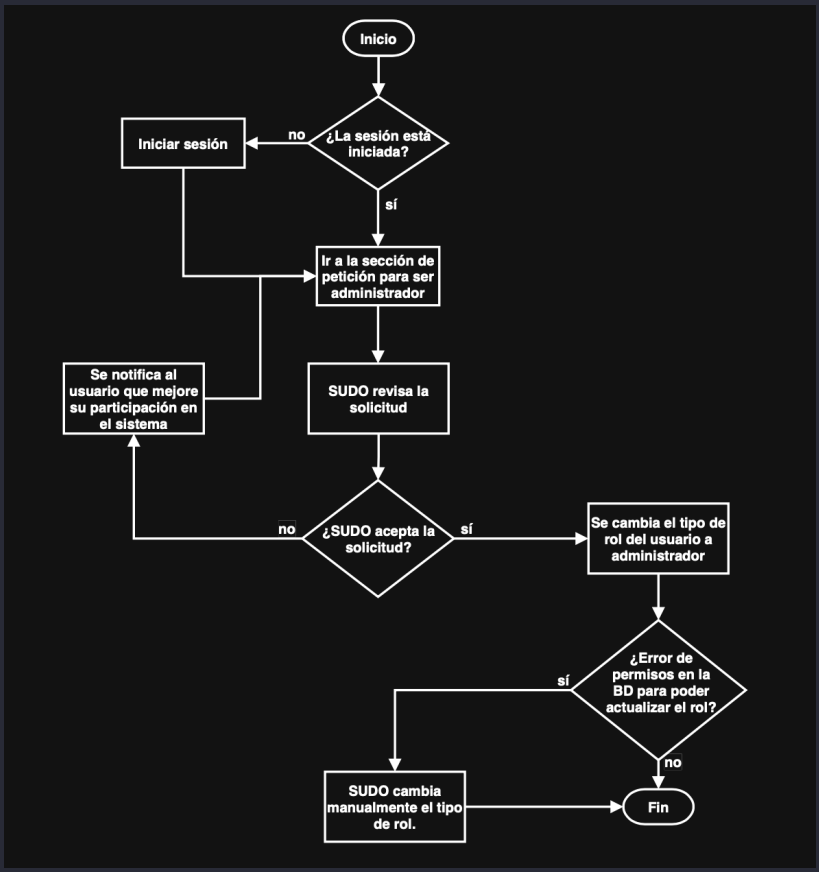
- Si hay un error, la cuenta no será eliminada, hasta repetir el proceso si errores.
- Si el usuario falla al autenticarse, no se permitirá la eliminación de la cuenta.



CU8: Registro de administradores

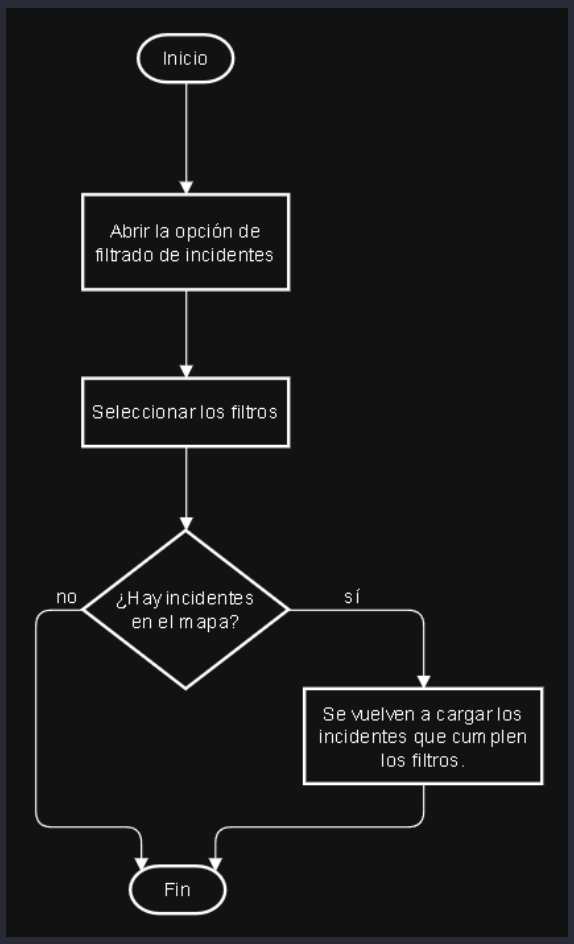
- ▷ Descripción
  Un usuario de registra para ser administrador en la sección correspondiente. El SUDO dependiendo el historial de reportes y su criterio, decide si cambiarle el rol de usuario a administrador.
- ▷ Precondiciones
  Un usuario con cuenta registrada y activa al momento de la solicitud y el SUDO con cuenta activa.
- ▷ Postcondiciones
  Cambio de rol al usuario o rechazo.
- ▷ Casos de uso alternativos
  - Si el SUDO rechaza la solicitud, el sistema notifica al usuario.
- ▷ Casos de uso extraordinarios

- Si la base de datos no puede actualizar el rol, ya sea por error de permiso, el usuario SUDO debe realizar la acción manualmente.



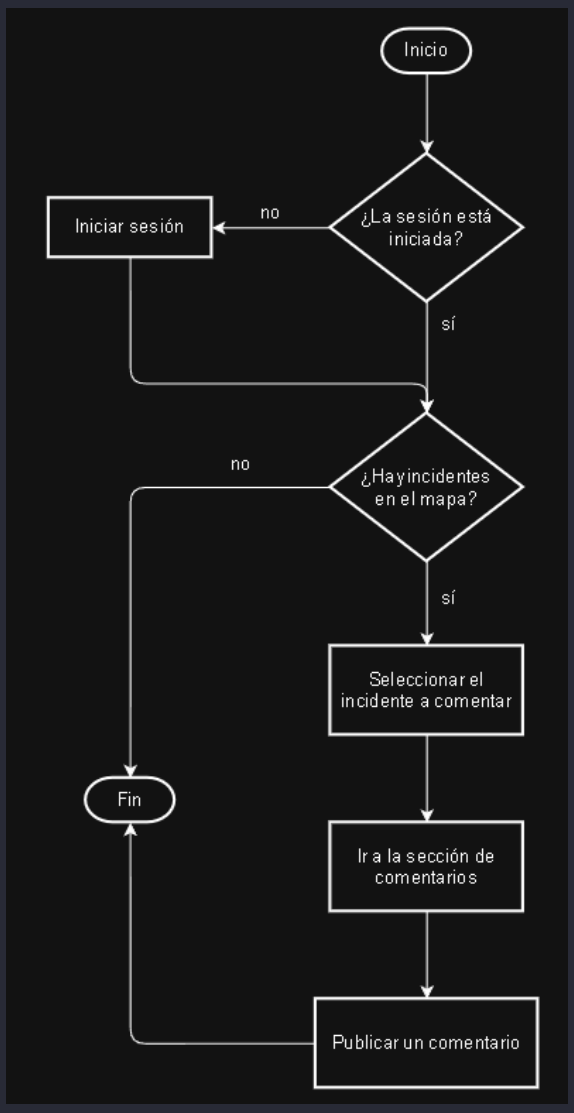
CU9: Filtrado de incidentes

- Descripción
  Un usuario al estar en la visualización del mapa, puede filtrar los incidentes en el mapa con las categorías disponibles.
- Precondiciones
  Un usuario dentro de la página e incidentes previamente reportados.
- Postcondiciones
  Incidentes ocultos en la vista si no cumplen las condiciones en los filtros.
- Casos de uso alternativos
  - Si no hay incidentes, el mapa continuará igual.
- Casos de uso extraordinarios
  - Si ocurre un error al momento del filtrado, volverá a un estado inicial.



CU10: Comentar un incidente

- Descripción
  Un usuario puede comentar un incidente en el mapa.
- Precondiciones
  Un usuario con cuenta registrada y activa al momento y un incidente en el mapa.
- Postcondiciones
  Comentario visible en el incidente.
- Casos de uso alternativos
  - Sin cuenta registrada y activa es posible ver los comentarios, más no comentar.
- Casos de uso extraordinarios
  - El comentario no es registrado ante cualquier error, no se guardan los cambios.



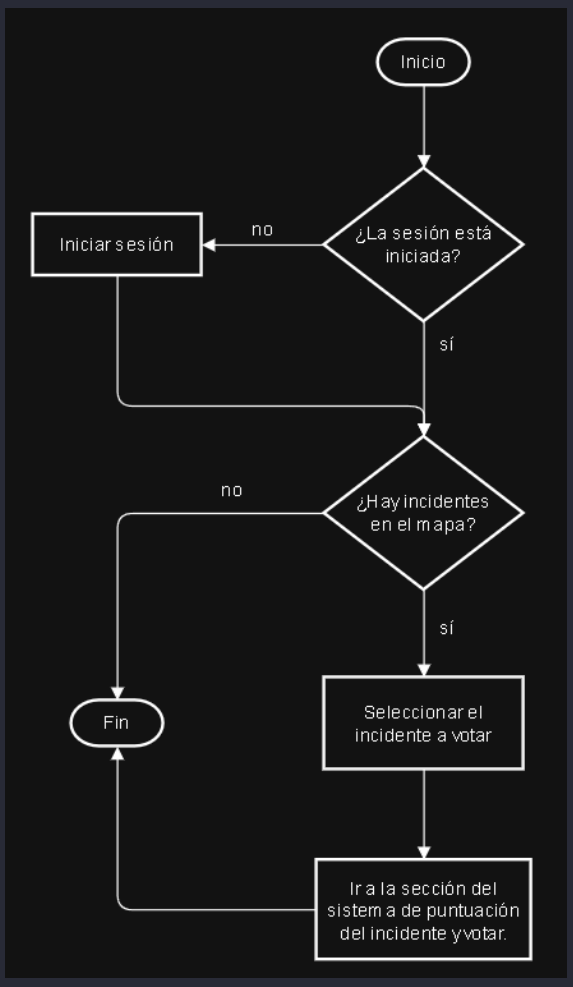
CU11: Interacciones de puntuación con incidentes

- Descripción
  Un usuario puede dar like a un incidente en el mapa.
- Precondiciones
  Un usuario con cuenta registrada y activa al momento y un incidente en el mapa.
- Postcondiciones
  Actualización del sistema de puntuación en el incidente.
- Casos de uso alternativos
  - El incidente mismo puede ser eliminado por un administrador si éste recibe una puntuación negativa considerable.

- No se puede interactuar en un incidente que no se encuentre en el mapa.

▷ Casos de uso extraordinarios

- Ante errores, la votación no es registrada, no se guardan los cambios.



CU12: Ver incidentes del cliente

▷ Descripción
Un usuario puede ver los incidentes que un administrador ha aprobado.

▷ Precondiciones
Un usuario con cuenta registrada y activa al momento.

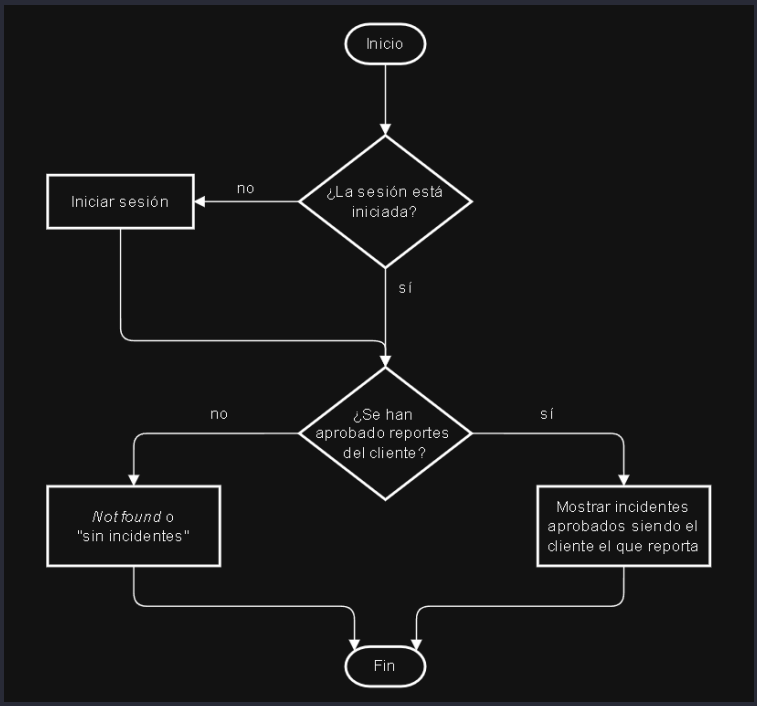
▷ Postcondiciones
Una vista que muestra los incidentes propios.

▷ Casos de uso alternativos

- Si el usuario no le han aceptado ningún incidente, se debe de mostrar una vista o mensaje con lo mismo.

▷ Casos de uso extraordinarios

- Ante errores, se debbe mostrar un 404, o la vista “sin incidentes”, mientras se arreglan los errores.



16. Casos de Uso Alternativos

CUAlt1: Datos Incorrectos

- Descripción: Este escenario ocurre cuando el usuario introduce datos incorrectos.
- Acciones para llevar acabo:
  - Validar los datos ingresados en cada campo.
  - Mostrar un mensaje de error para cuando los datos ingresados son incorrectos.

CUAlt2: Errores de Validación

- Descripción: Ocurre cuando los datos ingresados no cumplen con las reglas de validación.
- Acciones para llevar acabo:
  - Comprobar la integridad y formato de los datos ingresados.
  - Mostrar mensajes específicos por cada error de validación detectado.

17. Casos de Uso Extraordinarios

CUExt1: Desconexión del servidor

- Descripción: El servidor se desconecta o deja de estar disponible durante un periodo de tiempo determinado cuando el usuario lo estaba ocupando.
- Acciones a llevar acabo:
  - Detectar la pérdida de conexión y notificar al usuario con un mensaje de error.
  - Intentar la reconexión automática en intervalos regulares.
  - Tener planes de contigencias en servidores de respaldo o en contenedores.

CUExt2: Fallo de equipo o sistema no identificado

- Descripción: Se presenta cuando el sistema falla por un error inesperado.
- Acciones a llevar acabo: Contactar con los programadores, suspender el servicio lo antes posible para evitar problemas mayores, y actuar según el plan de contingencia (como reiniciar), o solucionar el sistema con un hotfix hasta que se encuentre una solución que contemple de la mejor manera la problemática, contactar a los usuarios, para que si alguno tuvo un problema de pérdida de datos, o un cambio de rol, etc, se pueda solucionar.