

# Links de los repositorios:

Aplicación móvil: <https://github.com/Erdf404/aplicacion-de-rastreo/tree/main>

API: [https://github.com/Max-FD/Api\\_app\\_TSJ/tree/main](https://github.com/Max-FD/Api_app_TSJ/tree/main)

Página web: <https://github.com/Erdf404/pagina-de-administrador>

link del mapa de uso  
gratuito(openstreetmap): <https://www.openstreetmap.org/#map=18/20.702506/-103.473546>

## Lista de errores, mejoras y sugerencias del sistema

### Errores detectados en la página web

1. **Visualización incorrecta del menú para el rol Encargado**  
El usuario con rol de Encargado puede ver en el menú todas las pestañas del sistema, aunque no tenga permisos para acceder a ellas.(corregir permisos del menú desplegable)
2. **Falla en la carga de asignación de rutas para el Encargado**  
Al ingresar al módulo de asignar rutas con un usuario Encargado, no se cargan los datos de las rutas existentes ni la información necesaria para realizar la asignación, impidiendo completar la acción desde la interfaz.
3. **Falta de indicador por rutas incompletas**  
Si un guardia envía una ronda sin completar todos los puntos, el sistema guarda y muestra únicamente los puntos registrados, pero no existe una marca visual o estado que indique claramente que la ronda está incompleta.

### Mejoras a implementar en la página web

1. **Asignación de múltiples rondas en un mismo día y horario**  
Permitir asignar varias rondas para el mismo guardia en un mismo día y hora sin que

el sistema muestre mensajes de validación innecesarios, siempre que las rutas sean distintas.

**2. Tipo de ruta mixta (GPS + QR)**

Agregar un nuevo tipo de ruta “Mixta” que combine validación por GPS y QR dentro de una misma ronda, aumentando la seguridad y reduciendo la posibilidad de fraude.

**3. Visualización de mapas en el resumen de rondines**

En la página de Rondines, en la opción de “Mostrar resumen de rondines” incluir un mapa que muestre la ruta realizada por el guardia, con los puntos marcados, similar a la visualización utilizada en el módulo de creación de rutas.

**4. Indicador explícito de estado de ronda**

Añadir un indicador visible para las rondas, por ejemplo: Ronda completada, Ronda incompleta. Esto para facilitar la revisión por parte de administradores y encargados.

**5. Recuperar contraseña**

Completar la función de recuperar la contraseña.

**6. Estadísticas de rondas**

Incluir gráficos con métricas de las rondas realizadas y el desempeño de los guardias

**7. Ciberseguridad**

Implementar mecanismos adicionales de seguridad de ciberseguridad, como tokens de sesión más robustos y autenticación de dos pasos.

**8. Historial de auditorías y cambios**

Registrar quién crea, modifica o elimina rutas, puntos y asignaciones para mayor trazabilidad.

**9. Actualizar mapa de la institución**

Agregar más detalles de la institución en OpenStreetMap.(Canchas o edificios futuros)

---

## Errores detectados en la aplicación móvil

**1. Falta de visualización de puntos en rondas exteriores**

Actualmente, la aplicación no muestra los puntos que el guardia debe registrar durante una ronda exterior, por lo que el administrador o encargado debe comunicar esta información de forma verbal, lo cual puede generar errores de comunicación.

## 2. Actualización incorrecta del estado de rondas

Al finalizar una ronda, el tablero de rondas pendientes y realizadas no se actualiza automáticamente.

Es necesario cerrar sesión y volver a iniciarla para que los cambios se reflejen correctamente.

# Mejoras a implementar en la aplicación móvil

## 1. Actualización en tiempo real del estado de rondas

Implementar una actualización automática de las tarjetas de rondas pendientes y rondas realizadas, sin necesidad de reiniciar sesión.

## 2. Mapa interactivo para rondas exteriores

Agregar un botón que muestre un mapa con los puntos asignados a la ronda exterior, tanto al seleccionarla como durante su ejecución.

Se recomienda el uso de **OpenStreetMap** por ser una solución gratuita.

## 3. Registro de ruta en rondas exteriores

Implementar el seguimiento de la ruta recorrida por el guardia durante la ronda, permitiendo una verificación más precisa del recorrido realizado.

## 4. Riesgo de modificación de la hora

Si el guardia cambia manualmente la hora del teléfono, podría alterar la coherencia temporal de los registros.

## 5. Sistema de anotaciones en rondas

Permitir que el guardia agregue comentarios u observaciones durante la ronda.

Esta mejora requiere modificaciones también en:

- a. Base de datos
- b. API
- c. Página web

## 6. Registro de ubicación en rondas interiores (QR)

Además del escaneo del código QR, guardar también la ubicación GPS del dispositivo para verificar que el guardia se encontraba dentro del edificio o área asignada al momento del registro.

## 7. Falta de cifrado en el almacenamiento local

Las rondas guardadas en SQLite no están cifradas.

Si un tercero obtiene acceso físico al dispositivo, podría extraer información sensible.

## 8. Iniciar sesión con datos biométricos(face id o huella digital)

Si se puede y con supervisión de alguien con conocimientos en ciberseguridad, implementar la posibilidad de iniciar sesión con face id o huella digital.

---

## Errores detectados en la api

### 1. Falta de limitación de solicitudes

La API no restringe el número de peticiones que puede enviar un cliente, lo que la hace vulnerable a ataques de saturación o envío masivo de datos.

### 2. Ausencia de validación de identidad del cliente

La API acepta información sin verificar la autenticidad del emisor, lo que permitiría que un tercero, conociendo la estructura del JSON, pueda enviar rondas falsas desde fuera de la aplicación móvil.

### 3. Falta de validación lógica de datos

La API no valida que los datos recibidos sean coherentes, por ejemplo:

- horas fuera de rango.
- rondas con duración negativa.
- coordenadas inválidas.

### 4. Falta de versionado de la API

La API no cuenta con versión (/v1, /v2), lo que puede generar incompatibilidades al actualizar la aplicación móvil o la web.

### 5. Ausencia de registros de auditoría (logs)

No se cuenta con un sistema de logs detallado para rastrear errores, accesos indebidos o fallos durante la sincronización.

## Mejoras a implementar en la api

### 1. Implementación de rate limiting

LIMITAR el número de solicitudes por usuario o IP, así como el tamaño máximo del payload, para prevenir abusos y ataques de denegación de servicio.

### 2. Seguridad mediante autenticación

Integrar:

- a. Tokens JWT
- b. Verificación de origen
- c. Restricción de endpoints a usuarios autenticados

Esto garantiza que solo clientes válidos puedan interactuar con la API.

### 3. registro de logins

Implementar un registro de los logins de los usuarios para evitar varios tipos de problemas.