

## **PROYECTO CONTROL DE COMPARENDO POLICIAL**

### **PRESENTADO POR:**

INGRID YULISSA MEDINA ESQUIVEL  
CAMILO ANDRES LOSADA RAMIREZ  
DANIEL CAICEDO TRUJILLO

### **DIRIGIDO A:**

JHON CORREDOR

CENTRO DE LA INDUSTRIA LA EMPRESA Y LOS SERVICIOS NEIVA

- HUILA 2024

# Índice

PROYECTO FOOD MEMORIZE .....	1
1. Introducción .....	3
1.1 Propósito .....	3
1.2 Alcance .....	3
2. Descripción de la Arquitectura .....	3
2.1 Arquitectura General del Sistema .....	3
2.2 Componentes del Sistema .....	3
3. Diseño Detallado .....	4
3.1 Diseño del Frontend.....	4
3.2 Diseño del Backend .....	4
3.3 Diseño de la Base de Datos.....	4
4. Diagrama de Casos de Uso .....	5
4.1 Diagrama UML.....	5
5. Gestión de Errores y Excepciones.....	5
5.1 Manejo de Errores.....	5
5.2 Excepciones .....	5
6. Prototipos de Interfaz de Usuario (UI) .....	6
6.1 Mockups .....	6
7. Diagramas de Flujo .....	6
8. Diagrama De Proceso.....	9
9. Diagrama de Secuencia .....	9
10. Diagrama de clases .....	9

## **1. Introducción**

### **1.1 Propósito**

El Proyecto de Control de Comparendos Policiales tiene como propósito Diseñar un sistema tecnológico integral que permita gestionar de manera eficiente las sanciones impuestas por las autoridades municipales. Este sistema busca mejorar la transparencia y la eficacia en el registro y seguimiento de comparendos, facilitando así el cumplimiento de las obligaciones legales por parte de los ciudadanos.

**2.** Además, la plataforma permitirá el acceso público a la información de comparendos, brindando a los ciudadanos la posibilidad de consultar sus sanciones, recibir notificaciones automáticas sobre infracciones, fechas de vencimiento. También incluirá la opción de gestionar acuerdos de pago de manera digital, eliminando la necesidad de trámites presenciales.

**3.** Con este proyecto, se pretende contribuir a una cultura de cumplimiento normativo y a la creación de un entorno más organizado y respetuoso de las normas dentro de la comunidad, optimizando la relación entre ciudadanos y autoridades mediante el uso de tecnología accesible y eficiente.

### **3.1 Alcance**

Creación de un software integral para el control y manejo de comparendos policiales, que incluirá funcionalidades para el registro y seguimiento de sanciones. Este sistema será de acceso público, permitiendo que los ciudadanos consulten información relevante, gestionen sus sanciones de manera eficiente.

Desarrollo de una interfaz intuitiva y accesible tanto para dispositivos móviles como para computadoras, asegurando una experiencia de usuario óptima y facilitando el acceso a los servicios de consulta y pago.

Generación de documentos oficiales: La inspectora podrá generar automáticamente documentos formales en formato PDF basados en la información que ingrese en el sistema. Esta funcionalidad permitirá expedir actas, constancias y oficios de manera precisa, reduciendo el trabajo manual y asegurando la coherencia documental

---

## 4. Descripción de la Arquitectura

- **Cliente (Aplicación Móvil y/o Web)**
  - Interfaz para la gestión de comparendos.
  - Comunicación con el backend mediante una API.
- **Servidor (Backend en C#)**
  - Procesamiento de datos y lógica del negocio.
  - Exposición de servicios a través de una **API REST**.
  - Gestión de usuarios y seguridad.
- **Base de Datos**
  - Almacenamiento de información de comparendos, usuarios y registros relevantes.

En términos generales, la aplicación móvil o web se conectará a un servidor que manejará las solicitudes y almacenará los datos en una base de datos.

### 4.1 Componentes del Sistema

#### 1. Frontend (Aplicación Móvil o Web)

##### Tecnología:

Angular (v16+): Framework principal para la interfaz web del usuario.

Tailwind CSS: Estilos limpios y responsivos.

TypeScript: Lenguaje tipado para mejor mantenimiento y escalabilidad.

Angular Material : Componentes UI para formularios, modales, tablas y notificaciones.

##### • Funciones principales:

- Interfaz para registrar, consultar y gestionar comparendos.
- Autenticación y autorización de usuarios.
- Notificaciones sobre el estado de los comparendos.

#### 2. Backend

##### • Tecnología:

- React Native: Desarrollo multiplataforma para Android e iOS.
- Expo: Framework para facilitar el despliegue rápido en dispositivos móviles.

- Fetch: Consumo de servicios REST desde dispositivos móviles.
- **Funciones principales:**
  - Procesamiento de datos y validación de reglas de negocio.
  - Exposición de para la gestión de comparendos.
  - Registro y auditoría de acciones.

### **3. Base de Datos**

- **Tecnología: SQL Server con Entity Framework Core.**
  - **Funciones principales:**
    - Almacenamiento de comparendos, usuarios, roles y registros de auditoría.
    - Optimización de consultas y relaciones entre tablas.
- 

## **5. Diseño Detallado**

### **5.1 Diseño del Frontend**

- Angular (v16+): Framework principal para la interfaz web del usuario.
- Tailwind CSS: Estilos limpios y responsivos.
- TypeScript: Lenguaje tipado para mejor mantenimiento y escalabilidad.
- Angular Material : Componentes UI para formularios, modales, tablas y notificaciones.

### **Mobile App**

- React Native: Desarrollo multiplataforma para Android e iOS.
- Expo: Framework para facilitar el despliegue rápido en dispositivos móviles.
- Fetch: Consumo de servicios REST desde dispositivos móviles.

### **5.2 Diseño del Backend**

- C# .NET 7 / .NET 8: Motor principal del backend, estructurado en capas (Controladores, Servicios, Repositorios).
- Swagger: Documentación y prueba de las API REST.
- Entity Framework Core: para comunicación con base de datos SQL Server.
- AutoMapper: Mapeo entre entidades y DTOs para separar lógica de dominio y presentación.
- JWT (JSON Web Tokens): Autenticación y autorización segura para los roles (ciudadano, inspectora, administrador).
- CORS y Middleware de logging: Seguridad y trazabilidad de las solicitudes.

### **5.3 Diseño de la Base de Datos**

- SQL Server: Almacén principal para toda la información del sistema (comparendos, pagos, usuarios, acuerdos, etc.).
-

## **6. Prototipos de Interfaz de Usuario (UI)**

### **6.1 Mockups**

para la realización de mockups utilizaos la herramienta de figma el cual nos ayuda a visualizar gráficamente como quedara el aplicativo

Incluir los prototipos de las pantallas principales:

- **Pantalla de bienvenida** (Registro/Iniciar sesión).
- **Pantalla de consulta de infracciones** (donde el usuario ve su historial de multas).
- **Pantalla de reportes administrativos** (visualización de comparendos y pagos procesados).
- **Pantalla de Anexar multas** (ingresa la información requerida de la infracción y se anexara la multa, además se le informara por medio de un correo al usuario)
- **Pantalla de acuerdo de pago** (donde se crearan los acuerdos de pago a las multas que se encuentren sin subsanar)

## **6. Diagrama de Casos de Uso**

Diagrama de casos de uso que mostrara los principales flujos de interacción del usuario,

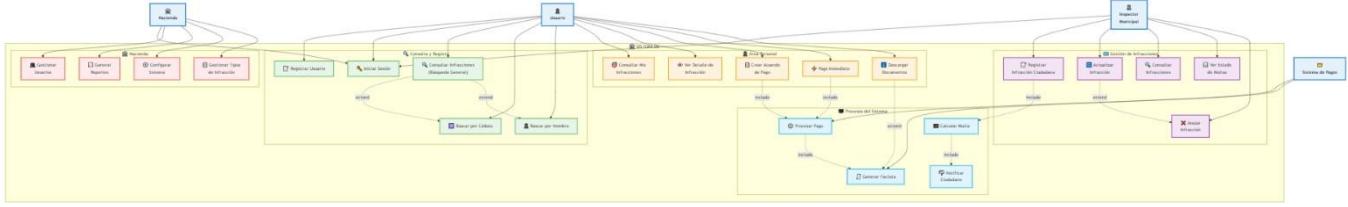
1. **Registro e inicio de sesión** → Los usuarios (inspectora, finanzas) inician sesión en el sistema.
2. **Registrar comparendo** → Una inspectora crea un nuevo comparendo en el sistema.
3. **Consultar comparendos** → Un usuario puede ver el estado de sus comparendos.
4. **Pagar comparendo** → El ciudadano puede realizar el pago en línea.
5. **Generar reportes** → Los administradores pueden obtener estadísticas.

Los errores comunes deben ser manejados con mensajes amigables para el usuario y logs en la consola para los desarrolladores.

Credenciales incorrectas (al iniciar sesión).

Problemas de conexión con el servidor.

Errores de validación (ejemplo: datos incompletos).



## 5.2 Excepciones

- En caso de errores críticos como fallos en la base de datos o problemas en el backend, se implementará un sistema de alerta para que los administradores puedan responder rápidamente.

Diagramas de Flujo

- **Iniciar del sistema**

- El usuario de finanza inicia sesión al sistema: Finanzas
- Si los datos son correctos, accede al sistema; si no, se muestra un error.

- **Gestión de Infracciones**

- El inspector selecciona el tipo de infracción.
- Se registra la fecha , tipo de infraccion , tipo de documento , numero de documento e correo para mandar informacion de la multa anexada.

- **Consulta de Comparendos**

- El usuario puede revisar su historial de infracciones.
- Si hay comparendos pendientes, se muestra el detalle.

- **Gestión de Pagos**

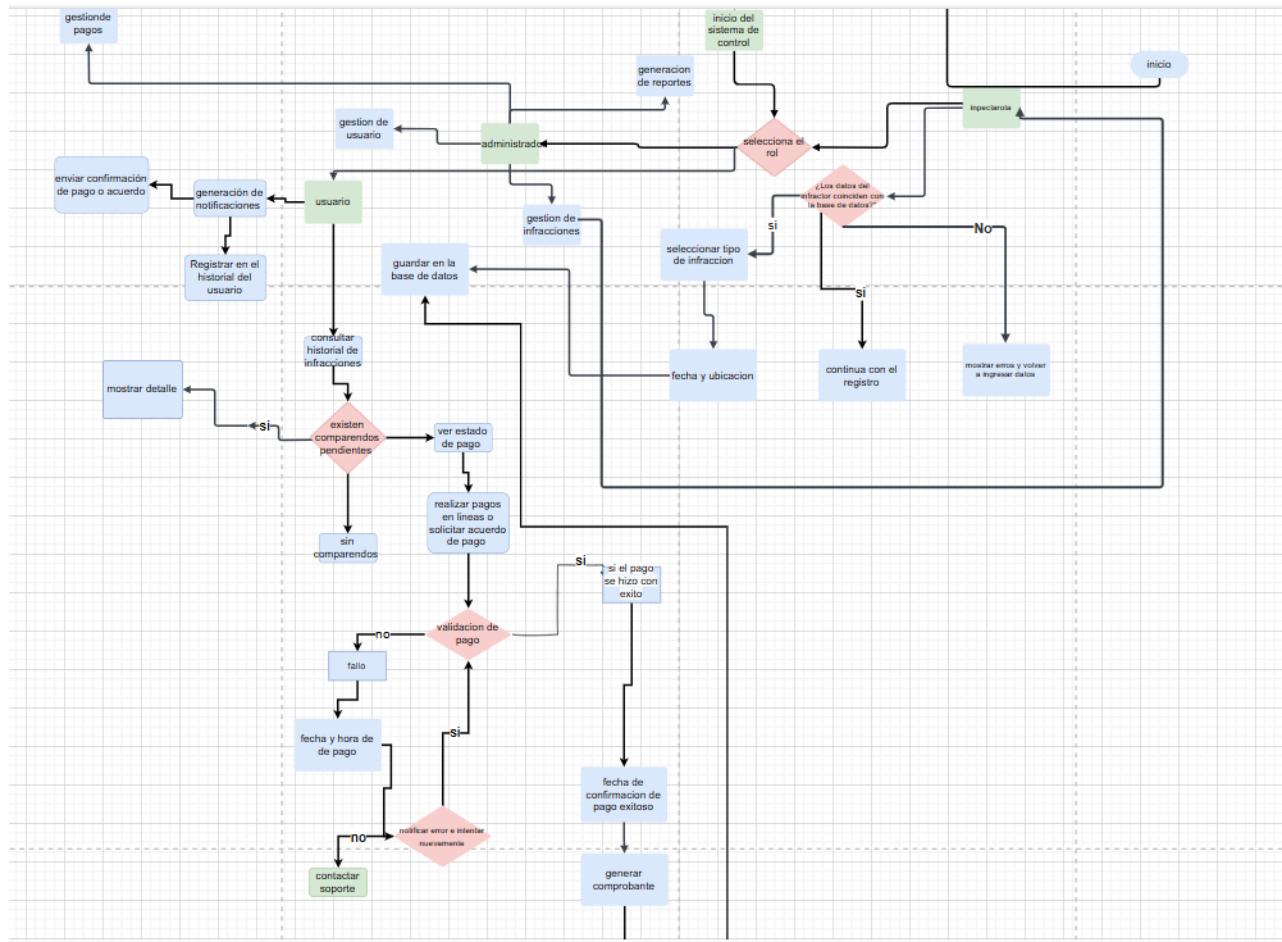
- Se verifica el estado del pago.

- **Notificaciones y Confirmaciones**

- Se envían notificaciones al usuario sobre pagos y acuerdos.
- Se actualiza el historial del usuario.

- **Manejo de Errores**

- Credenciales incorrectas → Mensaje de error.
- Fallo en la base de datos → Registro en consola.
- Problemas de conexión → Intento de reconexión.



## **7. Diagrama de Secuencia**

### **. Inicio de Sesión y Consulta de Historial de Infracciones**

- 1. El usuario ingresa sus credenciales.**
- 2. El sistema valida las credenciales:**
  - Si son correctas, asigna el rol correspondiente.
  - Si son incorrectas, muestra un mensaje de error.
- 3. El usuario solicita su historial de infracciones.**
- 4. El sistema verifica el rol:**
  - **Si es administrador**, se muestra el historial de todos los usuarios.
  - **Si es un usuario normal**, se muestra solo su historial.

### **. Proceso de Pago y Generación de Acuerdos**

---

### **3. Generación de Reportes**

- **El usuario de finanzas anexa una nueva multa**
- **El sistema recupera la información de la base de datos.**
- **El sistema genera el pdf.**
- **El administrador puede descargarlo .**
- **Manejo de Errores**

### **4. Manejo de Errores**

- Credenciales incorrectas → Mensaje de error.
- Fallo en la base de datos → Registro en consola.
- Problemas de conexión → Intento de reconexión.

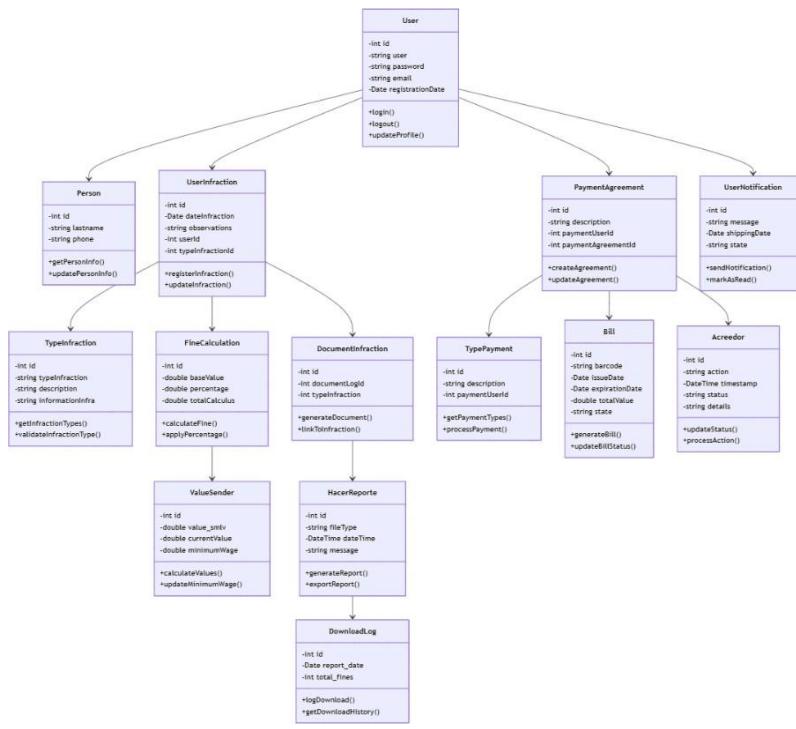
## 8. Diagrama de clases

El sistema se compone de varias clases principales:

- **Usuario:** Almacena datos de acceso y estado.
  - **Infracción:** Contiene información sobre la multa y su estado.
  - **Notificación:** Envía alertas sobre pagos y sanciones.

## 5. Manejo de Errores

- **Errores de autenticación:** Si las credenciales son incorrectas, se muestra un mensaje de error.
  - **Errores en la conexión:** Si hay fallas en el sistema, se notifica al usuario.
  - **Errores críticos:** Fallos en la base de datos se registran en logs para análisis.



## 9. Diagrama de proceso

### 1. Inicio de sesión:

- El usuario o administrador ingresa credenciales.
- Si los datos son incorrectos, se muestra un mensaje de error y se solicita un nuevo intento.
- Si son correctos, se asignan roles.

### 2. Consulta de infracciones:

- El usuario accede a su historial de infracciones.
- Puede seleccionar una infracción.

### 3. Consulta administrativa:

- Los administradores pueden revisar históricos de usuarios e estado de pago.
- 

## 5. Manejo de Errores

- **Errores de autenticación:** Si las credenciales son incorrectas, se muestra un mensaje de error.
- **Errores en la conexión:** Si hay fallas en el sistema, se notifica al usuario.
- **Errores en el pago:** Si la transacción falla, se informa al usuario y se ofrece soporte.
- **Errores críticos:** Fallos en la base de datos se registran en logs para análisis.

