



Sistema de Gestión de Multas Ciudadanas

## Presentación de Proyecto:



**AlcaldíaTech**  
Transparencia y Agilidad Municipal

2025

**GESTION DE COMPARENDO CIUDADANAS**



## Sistema de Gestión de Multas Ciudadanas

### 00-presentacion

#### Introducción al Proyecto:

Diseñar y poner en marcha un sistema tecnológico integral para la gestión de comparendos policiales en el municipio, que permita registrar, notificar y dar seguimiento a las sanciones impuestas por las autoridades. El propósito es fomentar la transparencia, eficiencia administrativa y responsabilidad ciudadana, fortaleciendo la relación entre los ciudadanos y las autoridades locales.

#### Alcance:

- Construcción de un software integral para el control y manejo de comparendos policiales, con funcionalidades para el registro y seguimiento de sanciones. Este sistema será de acceso público, permitiendo a los ciudadanos consultar información relevante.
- Desarrollo de una interfaz intuitiva y accesible tanto en dispositivos móviles como en computadoras, asegurando una experiencia de usuario satisfactoria y un acceso ágil a los servicios de consulta.
- Incorporación de un módulo administrativo que permita a la inspectora de policía generar automáticamente documentos en PDF



## Sistema de Gestión de Multas Ciudadanas

o a partir de los datos digitados, optimizando así la generación de actas .

- Implementación de una Pasarela de Pago electrónica que permita a los ciudadanos realizar el pago de las multas de forma segura y en línea.

Desarrollo de las funcionalidades necesarias para la gestión del proceso de cobro, incluyendo la emisión de notificaciones de cobro, el seguimiento al cobro coactivo, la etapa prejurídica, la etapa jurídica y la gestión de demandas relacionadas con las multas no pagadas.

### Descripción del Proyecto

La aplicación para la gestión de multas del municipio de Palermo es una plataforma orientada a modernizar y hacer más eficientes los procesos relacionados con la administración de sanciones derivadas de conductas que infringen el Código de Policía de Colombia. Este sistema está enfocado en faltas como agresiones, alteración del orden público, daños a bienes públicos o privados, entre otras que afectan la convivencia ciudadana.

### Principales características:

1. **Consulta de infracciones:** Los ciudadanos podrán verificar las multas asociadas a su comportamiento ingresando su tipo de documento e número de documento. La plataforma detallará el tipo de sanción cometida y el comportamiento que la originó.
2. **Información sobre pagos:** La aplicación especificará los montos de las sanciones e incluirá instrucciones claras sobre cómo efectuar los



## Sistema de Gestión de Multas Ciudadanas

pagos, que podrán realizarse en la Alcaldía Municipal o bancos autorizados.

3. **Pasarela de Pago:** Se integrará un sistema para **pagos electrónicos en línea** que permitirá a los ciudadanos liquidar las multas de manera directa y segura a través de la plataforma.
4. **Gestión de acuerdos de pago:** Los infractores que deseen regularizar su situación encontrarán los pasos requeridos para formalizar acuerdos de pago de forma presencial en la Alcaldía, con la debida orientación.
5. **Notificaciones y recordatorios:** El sistema enviará alertas sobre fechas límite, pagos pendientes y cambios en el estado de las sanciones.
6. **Notificación de Cobro y Proceso Coactivo:** El sistema automatizará la **emisión y envío de notificaciones de cobro** a los infractores con saldos pendientes, y registrará las etapas subsiguientes del proceso de cobro, que incluyen:
  - **Cobro Coactivo:** Registro y seguimiento a los procesos de cobro forzoso.
  - **Prejurídico:** Gestión de la etapa previa a acciones legales.
  - **Jurídico:** Registro de los casos que pasan a instancia legal.
7. **Historial de infracciones:** Los usuarios tendrán acceso a un historial completo de sanciones, diferenciando claramente entre las ya canceladas y las pendientes, promoviendo un seguimiento transparente.
8. **Generación de documentos oficiales:** La inspectora podrá generar automáticamente documentos formales en formato PDF basados en la información que ingrese en el sistema. Esta



## Sistema de Gestión de Multas Ciudadanas

funcionalidad permitirá expedir actas, constancias y oficios de manera precisa, reduciendo el trabajo manual y asegurando la coherencia documental.

### Integrantes del Equipo de Desarrollo:



**Ingrid Yulissa Medina Esquivel**

**Rol:** Líder y Desarrolladora (Base de datos, Frontend, Backend)

### Responsabilidades:

1. Garantizar una comunicación fluida dentro del equipo, promoviendo la colaboración y el trabajo articulado.
2. Supervisar el progreso general del desarrollo, velando por el



## Sistema de Gestión de Multas Ciudadanas

cumplimiento de los plazos establecidos.

- Revisar código y solucionar errores técnicos para asegurar un sistema estable y sin fallos.
- Coordinar la lógica de la base de datos y el backend, asegurando su correcto funcionamiento y la protección de los datos.
- Organizar reuniones periódicas con el equipo para revisar avances y proponer acciones correctivas o de ajuste.

**Contacto:** [ingrid.medina8@soy.sena.edu.co](mailto:ingrid.medina8@soy.sena.edu.co)





## Sistema de Gestión de Multas Ciudadanas

**Daniel Caicedo Trujillo**

**Rol:** Desarrollador (Frontend, Backend)

### Responsabilidades:

- Proponer nuevas soluciones que optimicen el código y enriquezcan la experiencia del usuario, enfocándose en la usabilidad y el rendimiento.
  - Brindar apoyo técnico a otros desarrolladores, asegurando la cohesión en el desarrollo del sistema.
  - Diseñar interfaces amigables que mejoren la experiencia de navegación.
  - Investigar metodologías aplicables al proyecto para adoptar buenas prácticas de desarrollo.
- Realizar pruebas y validaciones en las interfaces, verificando que cumplan con las necesidades del usuario final.





## Sistema de Gestión de Multas Ciudadanas

Camilo Andrés Losada Ramírez

**Rol:** Desarrollador (Base de datos, Frontend, Backend)

### Responsabilidades:

- Diseñar y estructurar la lógica de base de datos para garantizar el flujo eficiente de información en el sistema.
- Supervisar la base de datos, asegurando su estabilidad, precisión .
- Crear interfaces accesibles y funcionales para el usuario final.
- Afinar consultas y procesos internos que favorezcan el desempeño del sistema.
- Establecer mecanismos de respaldo y recuperación de datos para evitar pérdidas ante fallos técnicos.

**Contacto:** [camilo\\_losada@soy.sena.edu.co](mailto:camilo_losada@soy.sena.edu.co)

### Resumen del Equipo

Nuestro equipo está comprometido con la construcción de un sistema para el **Control de Comparendo Policial**, una herramienta que facilitará el registro, gestión y consulta de comparendos en tiempo real. Con esta solución, buscamos agilizar el trabajo y permitir un control más efectivo de las infracciones. A través de una plataforma robusta y segura, apuntamos a una trazabilidad clara de las multas y a dotar a los funcionarios de una herramienta eficaz para su gestión.

Nuestra apuesta por la calidad nos impulsa a utilizar tecnologías actualizadas que automatizan procesos y reducen los errores en la recolección de datos. Además,



## Sistema de Gestión de Multas Ciudadanas

el sistema contará con un mecanismo para generar automáticamente documentos en PDF con los datos ingresados por la inspectora, lo cual facilitará la documentación oficial de los casos. El sistema estará dividido en dos componentes:

1. **Módulo privado:** Exclusivo para autoridades como la inspectora de policía y el departamento de finanzas. Permitirá la gestión interna y toma de decisiones sobre los comparendos.
2. **Módulo público:** Disponible para los ciudadanos, quienes podrán consultar sus infracciones.

## Objetivos del Equipo

1. Diseñar una solución eficaz y confiable para la gestión de comparendos policiales, garantizando rapidez y seguridad en el tratamiento de datos.
2. Construir una plataforma intuitiva que permita a las autoridades acceder y trabajar con la información de forma ágil.
3. Facilitar a los ciudadanos el acceso a su información mediante un sistema transparente y funcional.
4. Incorporar funcionalidades que permitan a las autoridades generar automáticamente documentos oficiales desde el sistema, reduciendo tiempos y errores manuales.

## Metodología de Trabajo

- **Investigación tecnológica:** Analizar lenguajes de programación y tecnologías como typescripty SqlServer, evaluando sus beneficios



## Sistema de Gestión de Multas Ciudadanas

para el proyecto.

- **Selección y capacitación:** Elegir las herramientas más convenientes y capacitarse en su uso para garantizar su aplicación eficaz.
- **Asignación de tareas:** Distribuir de manera equitativa las funciones del proyecto, asegurando claridad en los roles.
- **Trabajo colaborativo:** Unificar y compartir avances de manera continua, promoviendo la comunicación constante.
- **Pruebas y depuración:** Ejecutar pruebas sistemáticas para detectar y corregir errores, garantizando un rendimiento óptimo del sistema.
- **Evaluación final:** Someter el producto a una revisión integral para confirmar el cumplimiento de los objetivos .

### 01- gestión-proyecto

#### Objetivo General

El objetivo de este proyecto es **diseñar e implementar un sistema integral y tecnológico** para la gestión de comparendos policiales en el municipio. Este sistema permitirá el registro, notificación y seguimiento de las sanciones impuestas por las autoridades locales, promoviendo la **transparencia, eficiencia administrativa y el cumplimiento de la normativa**. A través de esta plataforma, se fortalecerá la relación entre los ciudadanos y las autoridades, contribuyendo a un entorno más organizado y respetuoso con las normas.

#### Comunicaciones y Reuniones



## Sistema de Gestión de Multas Ciudadanas

Para mantener una coordinación adecuada y un desarrollo alineado con los objetivos, se establecieron los siguientes espacios:

1. Reuniones Diarias

Frecuencia: De lunes a viernes.

Propósito: Identificar riesgos, compartir avances y alinear tareas diarias.

2. Reuniones de Revisión

Propósito: Presentar entregables, recibir retroalimentación y ajustar el rumbo del proyecto según las observaciones.

3. Reuniones Técnicas

Propósito: Abordar temas relacionados con la arquitectura, integraciones, resolución de problemas complejos y garantizar la coherencia técnica.

## Cronograma de desarrollo

Hito	Descripción	Fecha de Inicio	Fecha de Fin	Porcentaje de Avance
1. Inicio del Proyecto	Firma del contrato y planificación inicial del proyecto.	01/07/2025	05/12/2025	10%
2. Diseño Conceptual	Entrega del diseño conceptual del sistema, incluyendo los módulos clave.	06/18/2025	15/12/2025	15%
3. Desarrollo del Prototipo	Desarrollo del prototipo funcional para pruebas iniciales.	16/08/2025	30/08/2025	20%
4. Pruebas del	Realización de	2/09/2025	20/09/2026	10%



## Sistema de Gestión de Multas Ciudadanas

Hito	Descripción	Fecha de Inicio	Fecha de Fin	Porcentaje de Avance
Prototipo	pruebas internas, corrección de errores.			
5. Implementación de Funcionalidades Principales	Desarrollo e integración de funcionalidades principales (registro de infracciones, gestión de pagos, etc.).	1/10/2025	15/10/2026	25%
6. Integración de Módulos de Gestión de Cobros y Documentos	Implementación de la pasarela de pago, generación de documentos PDF y gestión de cobros.	16/10/2025	12/11/2025	20%
7. Pruebas Finales	Pruebas integradas del sistema completo. Ajustes y optimización.	20/11/2025	25/11/2025	10%
8. Lanzamiento del Sistema	Entrega del sistema completo, listo para su implementación.	20/12/2025	23/12/2025	10%
G. Expansión y Actualizaciones	Implementación de funcionalidades adicionales	12/01/2026	2/02/2026	5%



## Sistema de Gestión de Multas Ciudadanas

### Matriz de riesgos del software

Riesgo	Probabilidad	Impacto	Nivel de Riesgo	Estrategia de Mitigación	Responsable
1. Retraso en la entrega del prototipo	Alta	Alto	Crítico	Establecer hitos semanales con revisiones periódicas. Identificar cuellos de botella y reasignar recursos si es necesario.	Ingrid (Líder)
2. Desviación del presupuesto	Media	Alto	Alto	Control y seguimiento estricto del presupuesto. Solicitar informes de avance financiero quincenalmente.	Ingrid (Líder)
3. Falta de disponibilidad de personal clave	Media	Alto	Alto	Planificar el equipo con respaldo en tareas clave y asegurar capacitación cruzada para roles críticos.	Ingrid (Líder)
4. Problemas de integración entre módulos	Media	Alto	Alto	Realizar pruebas de integración frecuentes durante el desarrollo. Definir procesos claros de integración y documentar fallos.	Camilo (Desarrollador)
5. Inestabilidad de la plataforma de pago	Baja	Alto	Medio	Realizar pruebas exhaustivas de la pasarela de pago. Monitorear constantemente el rendimiento del sistema de pago.	Daniel (Desarrollador)
6. Cambios en la normativa legal	Baja	Alto	Medio	Mantener comunicación constante con las autoridades locales para asegurar que el sistema cumpla con los requisitos legales actuales.	Ingrid (Líder)
7. Vulnerabilidades de seguridad	Baja	Muy Alto	Crítico	Implementar pruebas de seguridad continuas y	Camilo (Desarrollador)



## Sistema de Gestión de Multas Ciudadanas

Riesgo	Probabilidad	Impacto	Nivel de Riesgo	Estrategia de Mitigación	Responsable
8. Baja adopción por parte de los usuarios	Media	Medio	Medio	Realizar pruebas de usabilidad, recoger feedback de usuarios y ajustar la interfaz según sea necesario.	Daniel (Desarrollador)
9. Falta de escalabilidad del sistema	Baja	Alto	Medio	Realizar pruebas de carga y optimización del sistema desde las primeras etapas del proyecto.	Ingrid (Líder)
10. Errores técnicos durante las pruebas	Media	Medio	Moderado	Realizar pruebas unitarias	Camilo (Desarrollador)

## 02- recoleccion-informacion

### Entrevistas y encuestas

#### Encuesta sobre el Sistema de Control de Comparendos Policiales

Para la recolección de información inicial y el análisis de las necesidades y expectativas de los usuarios, se diseñó y aplicó una encuesta dirigida a los usuarios potenciales del sistema de control de comparendos. Esta encuesta se estructuró con el propósito de obtener opiniones directas sobre las funcionalidades clave que el sistema debe incluir, las dificultades actuales en el proceso de consulta y gestión de comparendos, y las preferencias tecnológicas de los usuarios.

#### Objetivo de la Encuesta

La encuesta fue diseñada con el objetivo de conocer las expectativas de los



## Sistema de Gestión de Multas Ciudadanas

usuarios con respecto al nuevo sistema de control de comparendos. A través de preguntas claras y concisas, buscamos evaluar tanto la experiencia de los usuarios con sistemas previos como su nivel de interés en las características que el nuevo sistema debería tener, como consultas rápidas, notificaciones automáticas y generación de informes detallados. Asimismo, se pretendía medir el grado de importancia que los usuarios le dan a la digitalización y modernización del proceso de gestión de comparendos.

### Metodología de Aplicación

La encuesta fue aplicada de manera digital, lo que permitió un alcance amplio y la recolección de datos en tiempo real. Se distribuyó entre usuarios que ya habían interactuado con sistemas similares o que podrían ser parte de la comunidad objetivo (ciudadanos, inspectores). Las preguntas cubrieron diversas áreas, como la frecuencia de consulta de comparendos, la facilidad de uso de plataformas anteriores, las dificultades comunes y las funcionalidades más valoradas.

El formulario fue diseñado utilizando Google Forms, una herramienta accesible y fácil de utilizar, que permitió recopilar respuestas de manera automática y organizada. La plataforma también ofreció la opción de garantizar la confidencialidad de las respuestas, un factor clave en un tema tan sensible como el de las sanciones y comparendos.

### Percepción de los Clientes

Los resultados obtenidos de los encuestados revelaron interés significativo en las funcionalidades propuestas para el nuevo sistema, destacando la consulta rápida de comparendos, las alertas automáticas sobre vencimientos de pagos y la posibilidad de generar informes detallados sobre el historial de infracciones y pagos.

Sin embargo, también se identificaron áreas de mejora, como la lentitud en los sistemas actuales y la dificultad para encontrar información clara. Estas respuestas han sido clave para ajustar el diseño del nuevo sistema y enfocarse en la mejora de la usabilidad y la optimización de tiempos de respuesta.



## Sistema de Gestión de Multas Ciudadanas

### Elección de la Herramienta

Se eligió Google Forms como plataforma para la recolección de datos debido a su accesibilidad y facilidad de uso. La herramienta permite crear formularios personalizados, gestionar las respuestas de manera sencilla y obtener análisis preliminares de los datos a través de gráficos automáticos. Además, ofrece la posibilidad de garantizar la confidencialidad de las respuestas, lo cual es fundamental en este tipo de encuestas.

El uso de esta herramienta fue una decisión estratégica para facilitar la participación de los usuarios sin generar barreras tecnológicas, permitiendo que cualquier persona con acceso a internet pudiera completar la encuesta desde su dispositivo móvil o computadora. También proporcionó una forma rápida y eficiente de procesar los datos para su posterior análisis.

link de encuesta [https://docs.google.com/forms/d/1XxuK\\_Nap-OyffzLZrlTHhjepveFP5aKWBGtp-DiaUIE/edit](https://docs.google.com/forms/d/1XxuK_Nap-OyffzLZrlTHhjepveFP5aKWBGtp-DiaUIE/edit)

### Benchmark o análisis de mercado

#### Objetivo del Análisis de Mercado

El objetivo principal de este análisis de mercado es identificar las mejores prácticas y características clave de los sistemas existentes en el mercado que gestionan comparendos, multas o sanciones, tanto en Colombia como en otros países. A través de este benchmarking, buscamos comparar las soluciones



## Sistema de Gestión de Multas Ciudadanas

disponibles y ajustar el diseño del sistema de control de comparendos para satisfacer las necesidades actuales y futuras de los usuarios.

### Metodología de Análisis de Mercado

Para realizar este análisis, se seleccionaron plataformas similares que ofrecen gestión de comparendos o sistemas de sanciones, con un enfoque en las siguientes áreas:

**Consultas de comparendos en línea:** Cómo los usuarios consultan sus infracciones y qué facilidades tienen para acceder a la información.

**Integración de pasarelas de pago:** Soluciones que permiten a los ciudadanos realizar pagos electrónicos de sus comparendos de manera sencilla y segura.

**Funcionalidades adicionales:** Incluyendo generación de informes detallados, notificaciones automáticas y seguimiento de casos.

**Las plataformas analizadas incluyen:**

**SIMIT** (Sistema Integrado de Información de Multas de Colombia): Un sistema utilizado a nivel nacional para la consulta y pago de multas de tránsito. Permite a los usuarios consultar el estado de sus infracciones y gestionar el pago de las mismas.

**Sistema de Control de Comparendos de la Secretaría de Movilidad de Bogotá:** Plataforma que ofrece consulta de infracciones de tránsito y opciones para generar acuerdos de pago. Incluye funciones de notificación y seguimiento de pagos.

**LICO** (Sistema de Liquidación de Infracciones de Convivencia): Implementado en algunas ciudades, permite la gestión de sanciones por infracciones relacionadas con el orden público y la convivencia ciudadana, integrando la notificación de



## Sistema de Gestión de Multas Ciudadanas

cobros y la gestión de acuerdos de pago.

### Datos recopilados del cliente o usuario

#### Objetivo de la Recolección de Datos

El objetivo principal de la recolección de datos fue identificar las necesidades y expectativas de los usuarios del sistema de control de comparendos, así como comprender cómo se gestionan actualmente las infracciones y sanciones en la alcaldía. A través de las entrevistas y encuestas, buscamos mejorar el diseño y la funcionalidad del nuevo sistema para asegurar que sea eficiente, intuitivo y adaptado a las condiciones reales del proceso.

#### Metodología para la Recopilación de Datos

La recolección de datos se realizó mediante dos métodos principales:

**Entrevistas directas con funcionarios de la alcaldía:** El equipo de desarrollo se desplazó a la alcaldía para realizar entrevistas con los funcionarios responsables de la gestión de comparendos. Esto permitió obtener información de primera mano sobre los procesos actuales, las herramientas utilizadas y las dificultades encontradas.

**Encuestas en línea a usuarios potenciales:** Se distribuyó una encuesta digital para obtener una visión más amplia de las necesidades de los ciudadanos, inspectores y otros usuarios potenciales del sistema. Las preguntas cubrieron desde la frecuencia de uso de plataformas digitales hasta las funcionalidades más valoradas.



## Sistema de Gestión de Multas Ciudadanas

### Descubrimiento en las Entrevistas Directas con la Alcaldía

Durante las entrevistas con los funcionarios de la alcaldía, nos enteramos de que el sistema actual de gestión de comparendos se realiza principalmente utilizando archivos Excel, lo que genera varios problemas operativos, como:

**Falta de centralización de la información:** La gestión de datos de los infractores y las multas está dispersa en múltiples hojas de Excel, lo que dificulta el seguimiento de los casos y la actualización de información.

**Riesgo de errores humanos:** El uso manual de Excel aumenta la probabilidad de errores en el registro de datos y duplicación de información, afectando la precisión del sistema.

**Limitación en la generación de reportes:** Los funcionarios deben elaborar informes de forma manual, lo que consume tiempo y no permite obtener informes detallados y automatizados de manera eficiente.

**Falta de integración:** No existe un sistema integrado que permita a los funcionarios gestionar comparendos, pagos y notificaciones de manera centralizada y en tiempo real.

### Datos Recopilados a Través de la Encuesta

Las respuestas de los usuarios reflejaron las necesidades y expectativas más comunes, como:

- Interfaz intuitiva y fácil de usar.
- Consultas rápidas de comparendos y la opción de ver el historial completo de sanciones.



## Sistema de Gestión de Multas Ciudadanas

- La importancia de un sistema de pago en línea seguro y accesible.

### Razón para Elegir Este Sistema de Recolección de Datos

Se optó por realizar entrevistas directas con la alcaldía porque queríamos comprender profundamente el contexto operativo actual y los desafíos específicos que enfrentan los funcionarios al gestionar los comparendos. La interacción cara a cara permitió una conversación más fluida y detallada, lo que nos proporcionó una visión clara de las deficiencias del sistema actual basado en Excel.

Las encuestas en línea, por su parte, se utilizaron para obtener un panorama más amplio de las necesidades de los usuarios finales (ciudadanos, inspectores, funcionarios de finanzas) y evaluar la percepción general sobre las plataformas digitales existentes.

## 03- requisitos

### Requisitos funcionales y no funcionales

#### Funcionales

1. El sistema debe permitir a los usuarios de finanzas registrarse e iniciar sesión de manera segura para gestionar sus comparendos.

2. Debe permitir a los ciudadanos consultar sus comparendos mediante múltiples criterios de búsqueda como tipo de documento e número de cédula.



## Sistema de Gestión de Multas Ciudadanas

3. La inspectora debe poder registrar comparendos con todos los datos necesarios, incluyendo evidencia fotográfica.
4. se integrará una funcionalidad para que, al momento en que la inspectora digite los datos, tenga la posibilidad de generar automáticamente un documento en formato PDF con la información ingresada.
5. Se debe permitir la generación automática de notificaciones y alertas sobre vencimientos de pagos y actualizaciones de estado de comparendos.
6. El sistema debe integrar herramientas de comunicación para que los ciudadanos puedan realizar consultas y solicitudes de revisión de comparendos.
7. Se implementará un control de acceso basado en roles, asegurando que cada usuario solo pueda acceder a las funciones correspondientes a su perfil (ciudadano, inspectora, Finanzas).
8. El sistema debe estar optimizado para dispositivos móviles y computadoras, garantizando accesibilidad y facilidad de uso.

## No Funcionales

1. La interfaz de usuario debe ser atractiva y fácil de usar, garantizando una experiencia intuitiva para todos los usuarios, incluyendo ciudadanos, inspectora y administradores del sistema.
2. El sistema debe ser capaz de manejar múltiples usuarios simultáneamente sin comprometer el rendimiento, permitiendo el acceso concurrente
3. La aplicación debe ser compatible con múltiples dispositivos y navegadores,



## Sistema de Gestión de Multas Ciudadanas

permitiendo el acceso desde computadoras y teléfonos móviles

### **Historias de usuario o casos de uso**

#### **Historias de usuario**

- Como ciudadano, quiero recibir una notificación cuando se me imponga un comparendo, para estar informado y tomar acciones a tiempo.
- Como ciudadano, quiero poder consultar mis comparendos y los detalles de cada multa, para conocer mi historial y los pagos pendientes.
- Como inspector, quiero registrar comparendos de manera rápida y precisa en el sistema, para llevar un mejor control de las infracciones.
- Como inspector, quiero acceder a notificaciones en tiempo real sobre las multas impuestas, para analizar tendencias y mejorar la gestión.
- Como funcionario de finanzas, quiero gestionar los pagos y aplicar descuentos según la normativa, para garantizar una correcta administración de los recursos.

#### **Casos de uso**

1. Registro e inicio de sesión → Los usuarios(inspectora, finanzas) inician sesión en el sistema.



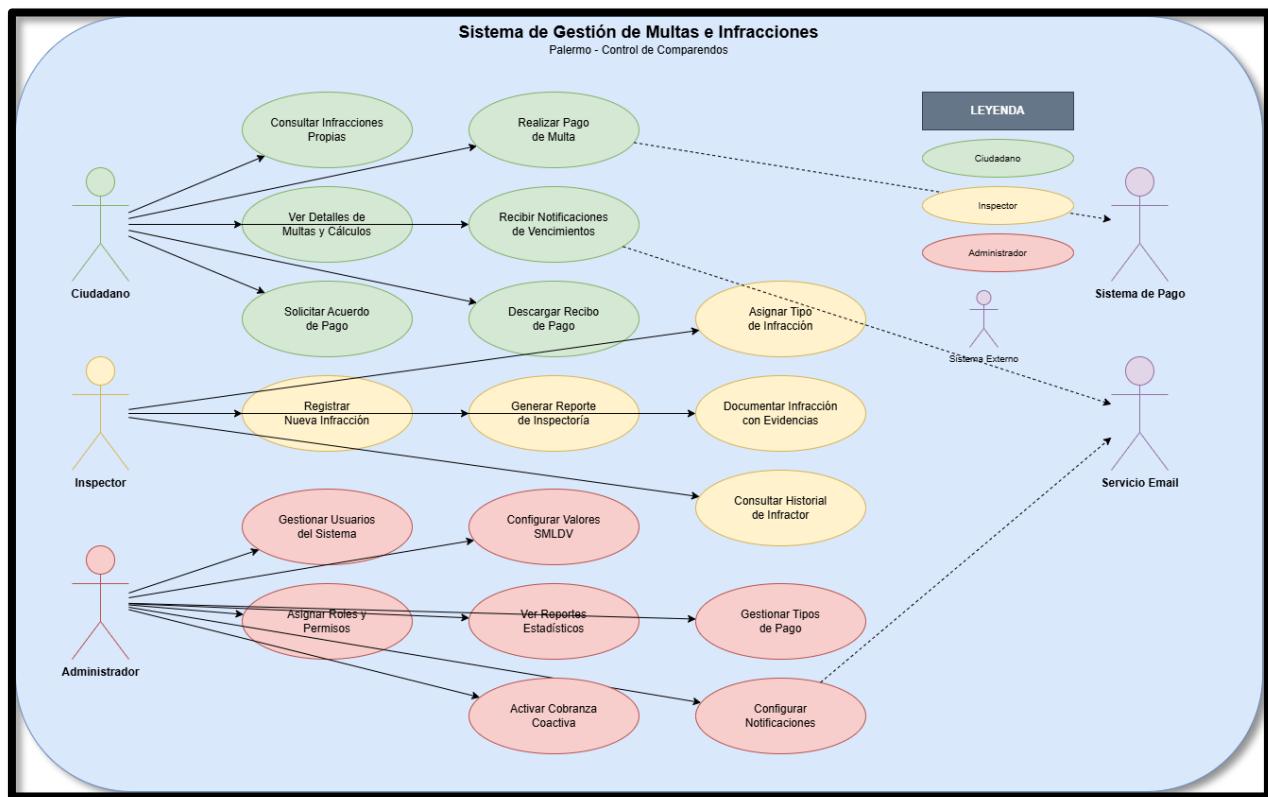
## Sistema de Gestión de Multas Ciudadanas

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).



## Sistema de Gestión de Multas Ciudadanas

### Diagramas de requisitos



05-diseño



## Sistema de Gestión de Multas Ciudadanas

### Wireframes, mockups, prototipos

#### IMAGENES DE PROTOTIPO FIGMA

##### 1 LOGIN



#### PARA MIRAR MAS DEL PROTOTIPO:

<https://www.figma.com/design/N86xp7ZdLuLv3Re1rdkEdg/Untitled?node-id=1063-427&p=f&t=7ylbuLQJALje4GuG-0>

#### Guía de Estilos - Palermo Portal

- Stack Tecnológico:
  - ❖ Framework: Angular 19.0.0
  - ❖ UI Library: Primen 19.1.4
  - ❖ Tema: @primeng/themes/aura



## Sistema de Gestión de Multas Ciudadanas

- ❖ Estilos: SCSS + Tailwind CSS
- Dependencias UI:
  - ❖ PrimeNG, TailwindCSS, SweetAlert2, y más.
  - ❖ Sistema de Colores
- Tema Claro:
  - ❖ Fondo principal: var(--p-surface-100) (Color de fondo claro)
  - ❖ Color de texto: var(--p-text-color) (Color de texto principal)
  - ❖ Fondo de tarjetas: var(--p-content-background) (Color de fondo para tarjetas)
- Colores personalizados del proyecto:
  - ❖ Acento coral: #ff7f50 (Color cálido, usado en botones y detalles)
  - ❖ Fondo de código: var(--p-surface-900) (Color oscuro para áreas de código)
  - ❖ Texto de código: var(--p-surface-200) (Color claro para texto de código)

### Tipografía

- Fuentes principales:
  - ❖ Cuerpo general: Roboto, Helvetica Neue (Fuentes sans-serif).
  - ❖ Código: Monaco, Consolas (Fuentes monoespaciadas).
- Tamaños recomendados:
  - ❖ Base: 14px
  - ❖ Títulos H1: 35px, H2: 28px, H3: 24.5px
  - Componentes PrimeNG
- Botones:
  - ❖ Variantes: primary, secondary, success, etc.
  - ❖ Tamaños: small, large.
- Cards y Select/Dropdown: Estilos personalizados con configuraciones de opciones y tamaños.
- Otros Componentes:



## Sistema de Gestión de Multas Ciudadanas

- ❖ Table, Date Picker, Toast, Message Service para notificaciones.

### Layout y Espaciado

- **Base Layout:**
  - ❖ Usamos Tailwind CSS para flexibilidad.
  - ❖ Grid y Flexbox para la disposición de los elementos.
- **Responsive:**
  - ❖ Breakpoints definidos para dispositivos móviles, tabletas y escritorios grandes.
- **Iconografía:**
  - ❖ Iconos básicos: pi pi-home, pi pi-user, pi pi-search, etc.
  - ❖ Estilos de iconos: Tamaños y colores ajustables con Tailwind CSS.
- **Animaciones**
  - ❖ Transiciones suaves con duración definida en las variables CSS.
  - ❖ Focus states configurados para accesibilidad.
  - ❖ Responsive Design.
- **Estrategia Mobile First:**
  - ❖ Diseño optimizado para dispositivos pequeños y escalable a grandes pantallas.
- **Accesibilidad**
  - ❖ Contraste de color y suavizado de fuentes.
  - ❖ Focus visible con un sistema de focus-ring para mejorar la navegación.
- **Librerías Adicionales**
  - ❖ SweetAlert2: Para modales y alertas personalizadas.
  - ❖ Chart.js: Gráficos y visualizaciones.



## Sistema de Gestión de Multas Ciudadanas

- ❖ SignalR: Comunicación en tiempo real.

## 06-estandares-y-normativa

### Convenciones de Código

#### 1.1 Nomenclatura de Clases y Archivos

Cumple Parcialmente: La mayoría de las clases usan correctamente la convención Pascal Case, pero algunos archivos como departmentController.cs y documentTypeController.cs usan minúscula inicial, lo cual debe corregirse.

#### 1.2 Nomenclatura de Variables y Propiedades

Cumple Parcialmente con Inconsistencias: Se usa \_camelCase con guion bajo para campos privados correctamente, pero hay inconsistencias con las propiedades públicas como email (debe ser Email en PascalCase).

#### 1.3 Estructura y Organización de Código

Cumple Adecuadamente: Se sigue correctamente la Clean Architecture, con capas bien definidas para la presentación, lógica de negocio, acceso a datos y modelos de dominio.

#### 1.4 Uso de Comentarios



## Sistema de Gestión de Multas Ciudadanas

Cumple Parcialmente: Se utilizan comentarios XML en algunas funciones críticas, pero falta documentación en muchos controladores y métodos públicos. También hay código comentado sin eliminar que debe ser revisado.

### 1.5 Formateo y Espaciado

Cumple: Se utiliza un formato de código limpio y consistente, con buena indentación y organización de imports.

## 2. Normas de Documentación

### 2.1 Documentación XML en Código

Cumple Parcialmente: Se usa documentación XML en algunas funciones clave, pero faltan detalles en los métodos públicos y parámetros. Es necesario documentar todos los controladores y métodos públicos.

### 2.2 Documentación Técnica

Cumple Excelentemente: La documentación de las pruebas unitarias es detallada y completa, con una guía clara para el uso de herramientas como xUnit y Moq.

### 2.3 Documentación de API

Cumple Parcialmente: Aunque se usa Swagger para la integración de la API, faltan atributos OpenAPI en algunos endpoints que deberían incluirse para una mejor comprensión de los desarrolladores.



## Sistema de Gestión de Multas Ciudadanas

### 3. Buenas Prácticas de Seguridad

#### 3.1 Autenticación y Autorización

Cumple Excelentemente: Se utiliza JWT con Refresh Tokens para la autenticación, y los controladores están protegidos con el atributo [Authorize].

#### 3.2 Validación de Entrada

Cumple Adecuadamente: Se usa FluentValidation y Google reCAPTCHA para validar la entrada de los usuarios, lo cual es una buena práctica para proteger la API de entradas maliciosas.

#### 3.3 Protección contra Ataques Comunes

Cumple: Se han implementado buenas prácticas para mitigar ataques comunes como CSRF, XSS y CSRF, con CORS y protección HTTPS configurados correctamente.

#### 3.4 Logging

Cumple: Se ha integrado un sistema de logging eficiente con el uso de ILogger y el nivel de log adecuado para las diferentes situaciones.



## Sistema de Gestión de Multas Ciudadanas

### 07- desarrollo

#### Bitacora de Desarrollo

##### Click up

###### Descripción del Proceso de Desarrollo:

Para gestionar el progreso del proyecto, utilizamos ClickUp como herramienta principal para asignar y hacer seguimiento a las tareas de cada integrante del equipo. A través de este sistema, cada miembro del equipo recibe una tarea específica, que se asigna y visualiza en el tablero de ClickUp.

Realizamos sprints diarios (daily stand-ups), donde cada integrante del equipo informa sobre el avance de su tarea asignada. Durante estos sprints, se discute si alguna tarea ha presentado errores o si no ha sido completada dentro del plazo estipulado. En caso de que un miembro no logre terminar su tarea, se reasignara. Además, llevamos un seguimiento constante de los errores y de cualquier impedimento que se presente durante el desarrollo. De esta manera, aseguramos que el proyecto avanza de manera eficiente y que todos los miembros se mantienen alineados con los objetivos y plazos establecidos.

Movil - React	3	+ ○	Mejorar alertas	≡	D	9/12/25	Normal
●	implementacion de movil con el backend	≡	D	9/19/25	Alta		
○	Resolver probela movil no reconoce tipo de d...	≡	D	Iun.			



## Sistema de Gestión de Multas Ciudadanas

### Capturas o logs de pruebas locales

#### Descripción del Proceso de Desarrollo:

En nuestro proceso de desarrollo, implementamos logs de seguimiento para registrar y monitorear los errores que puedan surgir cuando añadimos nuevos campos o funcionalidades al código. Esta práctica nos permite tener un registro detallado de las fallas, facilitando la identificación y resolución de problemas rápidamente.

Cada vez que se implementa una nueva función, el sistema genera un log con información sobre el error ocurrido, como en el siguiente ejemplo:

```
[ERROR] [ERROR] [documentTypeApi.obtenerTiposDocumento] Error al obtener tipos de documento [Error: La solicitud al servidor excedió el tiempo de espera. Intenta nuevamente.]  
{"component": "documentTypeApi", "function": "obtenerTiposDocumento"}  
[ERROR] [ERROR] [documentTypeApi.getDocumentTypeID] Error al buscar ID para abreviatura 'cc' [Error: La solicitud al servidor excedió el tiempo de espera. Intenta nuevamente.]  
{"component": "documentTypeApi", "function": "getDocumentTypeID"}
```

### Referencias técnicas

**Descripción del Proceso de Desarrollo:** Generalmente, recurrimos a la inteligencia artificial como una primera línea de apoyo para resolver problemas durante el desarrollo. Sin embargo, también consultamos videos de YouTube y otros recursos en línea para asegurarnos de que las soluciones que estamos implementando sean correctas y estén alineadas con las mejores prácticas. Además, ponemos en práctica lo aprendido en clase, aplicando los conceptos y soluciones a los problemas reales que encontramos. Para encontrar soluciones más efectivas, dialogamos con otros compañeros e instructores, ya que compartir experiencias nos permite saber si ellos han enfrentado problemas similares o si ya han encontrado una solución que podamos aplicar.

### 08- implementacion

### Guía de instalación



## Sistema de Gestión de Multas Ciudadanas

### Propósito

Esta sección documenta el proceso de despliegue y configuración del proyecto Palermo API, incluyendo la infraestructura como código, configuración de entornos y scripts de despliegue automatizado.

### Arquitectura de Despliegue

El proyecto utiliza una arquitectura basada en contenedores Docker con soporte para múltiples entornos:

- ❖ Develop - Entorno de desarrollo
- ❖ QA - Entorno de pruebas de calidad
- ❖ Staging - Entorno de pre-producción
- ❖ Production - Entorno de producción

### Prerrequisitos

Antes de comenzar el despliegue, asegúrese de tener instalado:

- ❖ Docker Engine (versión 20.10 o superior)  
`docker -version`



## Sistema de Gestión de Multas Ciudadanas

- ❖ Docker Compose (versión 1.26 o superior)

```
docker compose version
```

- ❖ .NET SDK 8.0 (para desarrollo local)

```
dotnet -version
```

- ❖ Git (para control de versiones)

```
git -version
```

### Instalación Local con Docker

#### Paso 1: Clonar el repositorio

```
git clone  
https://github.com/MultasPalermo/Palermo-  
api.git  
cd Palermo-api
```

#### Paso 2: Crear la red de Docker compartida

```
docker network create alcaldiafetch_network
```

#### Paso 3: Levantar base de datos



## Sistema de Gestión de Multas Ciudadanas

```
| cd alcaldiaFetch-DB  
| docker compose -f docker-compose-db.yml up -  
| d
```

### Paso 4: Configurar variables de entorno

```
# Editar archivo .env según el entorno  
deseado  
cd ../taller/DevOps/develop  
# Configurar valores en .env
```

### Paso 5: Levantar la API

```
docker compose up -d --build
```

### Paso 6: Verificar despliegue

```
docker ps  
# Debe mostrar los contenedores en ejecución
```

## Configuración de servidores o entorno



## Sistema de Gestión de Multas Ciudadanas

### Variables de Entorno

Cada entorno requiere configuración específica en su archivo .env:

```
# Configuración de la API
ASPNETCORE_ENVIRONMENT=Development
ASPNETCORE_URLS=http://+:8080

# Base de datos
ConnectionString__DefaultConnection=Server=alc
sqlserver-develop;Database=AlcaldiaDB;User
Id=sa;Password=Admin123!;TrustServerCertificate

# JWT
JWT__Secret=tu_clave_secreta_jwt_aqui
JWT__Issuer=AlcaldiaFetch
JWT__Audience=AlcaldiaFetchAPI
JWT__ExpirationMinutes=60

# Email
SMTP__Host=smtp.gmail.com
SMTP__Port=587
SMTP__Username=tu_email@gmail.com
SMTP__Password=tu_password_app

# ReCaptcha
ReCaptcha__SiteKey=tu_site_key
ReCaptcha__SecretKey=tu_secret_key
```

### Puertos por Entorno



## Sistema de Gestión de Multas Ciudadanas

Entorno	API Port	DB Port	Contenedor API	Contenedor DB
Develop	5100	1438	alcaldiafetch-api-develop	alcaldiafetch-sqlserver-develop
QA	5200	1436	alcaldiafetch-api-qa	alcaldiafetch-sqlserver-qa
Staging	5300	1437	alcaldiafetch-api-staging	alcaldiafetch-sqlserver-staging
Production	5400	1435	alcaldiafetch-api-prod	alcaldiafetch-sqlserver-prod

## Scripts de Despliegue



## Sistema de Gestión de Multas Ciudadanas

```
#!/bin/bash

ENVIRONMENT=$1

if [ -z "$ENVIRONMENT" ]; then
    echo "Uso: ./deploy.sh [develop|qa|staging|prod]"
    exit 1
fi

echo "🚀 Desplegando entorno: $ENVIRONMENT"

# Crear red si no existe
docker network create alcaldiafetch_network 2>/dev/null || true

# Levantar base de datos
echo "🕒 Levantando base de datos..."
docker compose -f alcaldiaFetch-DB/docker-compose-db.yml up -d

# Esperar a que la base de datos esté lista
echo "🕒 Esperando base de datos..."
sleep 10

# Desplegar API
echo "🌐 Desplegando API..."
cd taller/DevOps/$ENVIRONMENT
docker compose up -d --build

echo "✅ Despliegue completado"
echo "🌐 Estado de contenedores:"
docker ps --filter "label=com.alcaldiafetch.environment=$ENVIRONMENT"
```

### Script para Windows PowerShell:



## Sistema de Gestión de Multas Ciudadanas

```
# deploy.ps1
param(
    [Parameter(Mandatory=$true)]
    [ValidateSet('develop','qa','staging','prod')]
    [string]$Environment
)

Write-Host "❗ Desplegando entorno: $Environment" -ForegroundColor Green

# Crear red si no existe
docker network create alcaldiafetch_network 2>$null

# Levantar base de datos
Write-Host "🕒 Levantando base de datos..." -ForegroundColor Yellow
docker compose -f alcaldiaFetch-DB/docker-compose-db.yml up -d

# Esperar a que la base de datos esté lista
Write-Host "⏳ Esperando base de datos..." -ForegroundColor Yellow
Start-Sleep -Seconds 10

# Desplegar API
Write-Host "🌐 Desplegando API...." -ForegroundColor Yellow
Set-Location "taller/DevOps/$Environment"
docker compose up -d --build

Write-Host "✅ Despliegue completado" -ForegroundColor Green
docker ps --filter "label=com.alcaldiafetch.environment=$Environment"
```

## 09- testing-y-calidad

**Link del archivo:**

<https://drive.google.com/file/d/11Vu-8VU26v06NDTGKXkY3S2QOXdSpBYA/view?usp=sharing>

## 10- manuales

**Link del archivo Manual de usuario:**

<https://docs.google.com/document/d/1bilj4lv4yc63ntAK07Ds53AGlMrCT6s6/edit?usp=sharing&suid=105323076446767140821srtpof=true>

**Link del archivo Manual técnico :**



## Sistema de Gestión de Multas Ciudadanas

[https://docs.google.com/document/d/1tN3MIGBC\\_kGVfxKaL2nl621xFuswNcZW/edit?usp=sharing&Coid=105323076446797140821&rtpof=true&csd=true](https://docs.google.com/document/d/1tN3MIGBC_kGVfxKaL2nl621xFuswNcZW/edit?usp=sharing&Coid=105323076446797140821&rtpof=true&csd=true)

### 11- costos

**Propósito:** Esta sección describe los **costos financieros** involucrados en el desarrollo y mantenimiento del proyecto gestión de multas y comparendos, con un enfoque en **presupuesto económico**.

#### 1. Presupuesto Estimado del Proyecto

Categoría	Costo Aproximado (COP)
Recursos Humanos	\$1,500,000 - \$4,500,000 COP
Licencias de Software	\$3,000,000 - \$1,000,000 COP
Infraestructura en la Nube	\$1,000,000 - \$3,000,000 COP
Servicios de API	Variable (por transacción)
Herramientas de Desarrollo	\$0
Seguridad	\$150,000 - \$500,000 COP
Mantenimiento y Soporte	\$200,000 - \$800,000 COP

#### 2. Costos de Recursos, Licencias, etc.



## Sistema de Gestión de Multas Ciudadanas

### Recursos Humanos:

Desarrolladores Backend (C# / .NET): Aproximadamente \$2,500,000 - \$4,000,000 COP mensuales por desarrollador (equivalente a \$30,000,000 - \$48,000,000 COP anuales por desarrollador).

Desarrolladores Frontend (Angular / React): Aproximadamente \$2,000,000 - \$3,500,000 COP mensuales por desarrollador (equivalente a \$24,000,000 - \$42,000,000 COP anuales por desarrollador).

QA / Tester: Aproximadamente \$1,500,000 - \$2,500,000 COP mensuales por tester (equivalente a \$18,000,000 - \$30,000,000 COP anuales por tester).

### Licencias de Software:

Visual Studio / JetBrains: Si usas versiones gratuitas o de estudiantes, los costos son \$0 - \$500,000 COP anuales. Las licencias para estudiantes o equipos pequeños suelen estar disponibles de manera gratuita.

Docker / Kubernetes: \$0 si se usa la versión gratuita.

### Infraestructura en la Nube:



## Sistema de Gestión de Multas Ciudadanas

**Servicios en la nube (AWS, Azure, Google Cloud):** Con uso moderado y plan gratuito, puedes obtener créditos de hasta \$100 por cada nuevo proyecto de estudiante. Los costos aproximados pueden ser de \$2,000,000 - \$5,000,000 COP anuales si se elige un plan de pago, pero se puede reducir a \$0 si se usan los planes gratuitos de plataformas como AWS Educate o Azure for Students.

### **Servicios de API:**

**MercadoPago, Stripe, etc.:** Los costos de pasarelas de pago generalmente dependen de las transacciones. Si el proyecto es pequeño, los costos de pasarela de pago serían mínimos, 2.9% + \$0.30 USD por transacción.

### **Seguridad:**

**SSL/TLS:** Usar certificados gratuitos como Let's Encrypt o servicios educativos. Los costos de implementación son \$0 si se utilizan estas opciones.

**Autenticación JWT:** el costo es \$0.

### **Mantenimiento y Soporte:**



## Sistema de Gestión de Multas Ciudadanas

Los costos de mantenimiento anuales representan entre 10-15% del costo total de desarrollo, lo que podría equivaler a \$5,000,000 - \$10,000,000 COP anuales para cubrir pequeñas actualizaciones o soporte.

### 12- investigación

**Propósito:** Esta sección tiene como objetivo proporcionar la base teórica y explorar el soporte técnico y científico que respalda el proyecto Palermo API. Se incluye un análisis comparativo con otras soluciones similares, así como estudios previos que han abordado problemáticas o soluciones similares a las que se presentan en este proyecto.

#### 1. Referencias Bibliográficas

Las referencias bibliográficas sirven para fundamentar el desarrollo de Palermo API desde una perspectiva técnica, científica y académica. Algunas de las fuentes clave son:

1. Gamma, E., Helm, R., Johnson, R., C Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

**Aplicación:** El diseño de la API se basa en patrones de diseño como GRASP, MVC y Repository, que son fundamentales para crear aplicaciones sostenibles y escalables.



## Sistema de Gestión de Multas Ciudadanas

2. Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.

**Aplicación:** Se aplican los principios de código limpio para asegurar que el código sea legible, mantenible y testable a largo plazo.

3. Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.

**Aplicación:** El diseño de la API sigue los patrones de arquitectura empresarial, como Service Layer y Data Transfer Object (DTO), para lograr una separación clara entre las diferentes capas del sistema.

4. Microsoft. (2020). *Building Secure Web Applications with ASP.NET Core*. Microsoft Press.

**Aplicación:** La referencia proporciona técnicas y mejores prácticas para construir aplicaciones web seguras con ASP.NET Core, que es el framework utilizado en Palermo API. Los temas incluyen la autenticación JWT, protección contra ataques CSRF, y encriptación de datos.