

SENA – Centro de Industria y Construcción Tecnólogo en Análisis y Desarrollo de Software

# FlotaxAGC

Desarrollo de un Sistema Integral para el Control y Mantenimiento de Vehículos

##### Presentado por:

Edwar Farid Gomez Sanchez

##### Bajo la dirección de:

Ing. Cesar Arturo Esquivel Cortes

Ibagué, Tolima 05 de mayo de 2025

### Contraportada

Este proyecto fue desarrollado como parte del proceso formativo del programa Tecnólogo en Análisis y Desarrollo de Software del Servicio Nacional de Aprendizaje – SENA, Regional Tolima.

La finalidad del sistema es ofrecer a los ciudadanos una solución tecnológica que facilite la gestión personal de su flota vehicular. El sistema permite a los usuarios registrar, consultar y dar seguimiento a los mantenimientos y documentación legal de sus vehículos, incluyendo motos, automóviles, buses, taxis y servicios de transporte como Uber.

Este documento y el sistema desarrollado están destinados exclusivamente con fines educativos. La reproducción, distribución o modificación sin autorización expresa del autor o de la entidad educativa está prohibida.

Todos los derechos reservados.

SENA – 2025

1. **Tabla de contenidos**

[FlotaxAGC 1](#_bookmark0)

1. [Contraportada 2](#_bookmark1)
2. [Introducción 4](#_bookmark2)
3. [Descripción del Problema 5](#_bookmark3)
4. [Contenido del proyecto 6](#_bookmark4)
   1. [Análisis 6](#_bookmark5)
   2. [Diseño 9](#_bookmark6)
   3. [Codificación 13](#_bookmark7)
   4. [Pruebas 13](#_bookmark8)

## Introducción

En la actualidad, muchas personas cuentan con vehículos de uso personal o laboral que requieren un seguimiento constante en cuanto a su mantenimiento, documentación legal y otros aspectos clave para su funcionamiento seguro y eficiente. Sin embargo, el olvido de fechas importantes como el vencimiento del SOAT, la tecnomecánica o el próximo cambio de aceite puede generar consecuencias legales, económicas e incluso de seguridad vial.

Con base en esta problemática, surge el proyecto **"Sistema Integral para la Gestión de Flota Vehicular Personal"**, el cual tiene como objetivo brindar a los propietarios de vehículos una herramienta intuitiva, automatizada y accesible para gestionar toda la información relevante de sus automóviles, motos, taxis o servicios de transporte como Uber y buses.

Este sistema permitirá registrar cada vehículo, asociar documentos con fechas de vencimiento, programar mantenimientos, recibir alertas automatizadas y visualizar reportes. Todo esto contribuirá a un mayor orden y control del estado legal y técnico de los vehículos, reduciendo el riesgo de sanciones, accidentes o gastos innecesarios por descuidos.

La solución propuesta se enfoca principalmente en usuarios de la ciudad de Ibagué, pero tiene potencial de ampliación a nivel nacional. Su diseño contempla una interfaz amigable, validaciones, gestión de sesiones seguras y una base de datos bien estructurada para garantizar eficiencia y usabilidad.

## Descripción del Problema

En la actualidad, gran parte de los propietarios de vehículos particulares enfrentan dificultades para realizar un seguimiento adecuado y organizado del estado legal y técnico de sus medios de transporte. A pesar del crecimiento de la tecnología y la accesibilidad a herramientas digitales, aún es común que muchas personas olviden realizar gestiones esenciales como la renovación oportuna del Seguro Obligatorio de Accidentes de Tránsito (SOAT), la revisión técnico-mecánica o el pago de impuestos y comparendos, lo que conlleva riesgos importantes tanto para su seguridad como para su economía.

Este problema no solo radica en el desconocimiento o desinterés, sino también en la ausencia de sistemas centralizados y accesibles que brinden alertas personalizadas y recordatorios automáticos. Muchas veces, los propietarios dependen de su memoria o de recordatorios físicos poco eficientes, lo que puede resultar en el uso de vehículos que no cumplen con las normativas vigentes, exponiéndose a sanciones legales, inmovilización del vehículo o accidentes por fallas mecánicas prevenibles.

En ciudades como Ibagué, donde el parque automotor continúa creciendo y muchos ciudadanos utilizan vehículos como parte de su actividad económica (por ejemplo, en servicios de transporte tipo Uber, taxis o motos de reparto), la falta de una herramienta tecnológica que centralice la gestión documental y mecánica representa una oportunidad desaprovechada. Además, el manejo manual de registros, facturas de mantenimiento o fechas de vencimiento genera desorden, pérdida de información y dificultades para mantener una gestión clara de los vehículos.

La ausencia de un sistema integral y automatizado que ayude al usuario a recibir alertas anticipadas, almacenar documentación digital, registrar mantenimientos y visualizar el historial completo de su vehículo representa una brecha significativa en la gestión moderna de la movilidad urbana y personal. Por lo tanto, es necesario implementar una solución que aproveche las tecnologías de la información para facilitar estas tareas, reducir el margen de error humano y mejorar tanto la experiencia del usuario como la seguridad vial en general.

## Contenido del proyecto

### Análisis

##### Planificación Scrum (JIRA)

Para el desarrollo del sistema se utilizó la metodología ágil Scrum, la cual permite organizar el trabajo en iteraciones cortas y enfocadas llamadas *sprints*. Este enfoque facilita la entrega continua de funcionalidades, así como la adaptación rápida a cambios durante el proceso de desarrollo.

El equipo estuvo conformado por un *Scrum Master*, quien se encargó de garantizar que el marco Scrum fuera entendido y aplicado correctamente, un *Product Owner*, encargado de representar al cliente y priorizar los requerimientos, y un equipo de desarrollo, compuesto por programadores que ejecutaron las tareas técnicas del sistema.

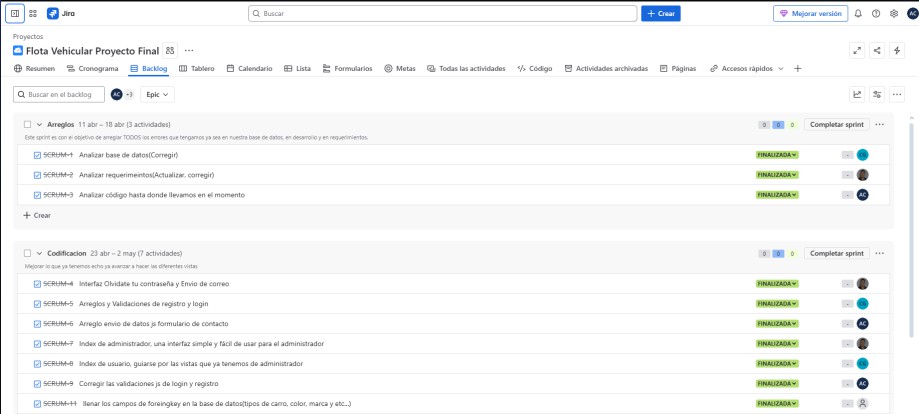
La planificación del proyecto se gestionó mediante la herramienta JIRA, donde se configuró un proyecto tipo *Scrum Software*. En esta plataforma se definieron las épicas principales, como el registro de vehículos, gestión de documentos (SOAT, tecnomecánica, impuestos), control de mantenimientos, generación de alertas, reportes, gestión de usuarios y seguridad. A partir de estas épicas se desglosaron historias de usuario que se organizaron en el *Product Backlog*.

Se estableció una duración de dos semanas por sprint, con una planificación inicial conocida como *Sprint 0*, en la cual se configuró el entorno de desarrollo, la base de datos, el repositorio y el diseño general de la arquitectura del sistema.

Posteriormente, los *Sprints 1 al 4* abarcaron el desarrollo de las funcionalidades principales, y el *Sprint 5* se enfocó en pruebas finales, ajustes y despliegue del sistema.

Durante cada sprint se realizaron ceremonias Scrum como la planificación de sprint, reuniones diarias (dailies), revisión de sprint y retrospectivas. Estas actividades permitieron mantener una comunicación constante, detectar bloqueos y asegurar una entrega progresiva de valor.

La planificación en JIRA fue esencial para el seguimiento de avances, la visualización de tareas pendientes y completadas, y la documentación de decisiones tomadas durante el ciclo de desarrollo.



##### Requerimientos funcionales (RF) RF-01. Registro de usuarios:

El sistema permitirá a nuevos usuarios registrarse mediante un formulario que solicitará nombres, apellidos, número telefónico, correo electrónico y la creación de una contraseña segura.

##### RF-02. Inicio de sesión:

Los usuarios registrados podrán iniciar sesión ingresando su nombre, apellido y contraseña previamente registrada.

##### RF-03. Recuperación de contraseña:

El sistema ofrecerá la opción de recuperación de contraseña en caso de olvido, mediante correo electrónico o verificación por número de teléfono.

##### RF-04. Registro de vehículos:

El usuario podrá registrar uno o más vehículos, indicando información como tipo de vehículo, marca, modelo, placa, cilindraje y fecha de mantenimiento.

##### RF-05. Información de mantenimientos:

El sistema permitirá registrar, consultar y editar información relacionada con mantenimientos realizados, incluyendo revisiones, cambios de aceite, revisión de llantas y reparaciones generales.

##### RF-06. Gestión de alertas:

El software generará alertas automáticas visibles en la interfaz del usuario relacionadas con documentos importantes como SOAT, tecnomecánica, licencia de

conducción, pico y placa, entre otros.

##### RF-07. Historial de alertas:

El sistema mostrará un historial completo de alertas generadas, mantenimientos realizados, reparaciones, y fechas programadas para futuros mantenimientos.

##### RF-08. Notificaciones de mantenimiento:

Los usuarios recibirán notificaciones anticipadas cuando sus vehículos estén próximos a necesitar mantenimiento o renovación de documentos importantes.

##### RF-09. Notificación por correo y SMS:

El sistema enviará recordatorios y alertas mediante correo electrónico y mensajes SMS, según los datos de contacto registrados por el usuario.

##### RF-10. Alertas de vencimiento o documentos expirados:

Se notificará a los usuarios cuando un documento esté próximo a vencer o ya se encuentre vencido, con mensajes específicos indicando la acción necesaria.

##### RF-11. Consulta de multas:

El sistema proporcionará al usuario un resumen de las multas de tránsito asociadas a su documento, permitiendo su consulta y seguimiento.

##### RF-12. Reporte de problemas de notificación:

El sistema dispondrá de una sección donde el usuario podrá reportar problemas relacionados con la falta de recepción de notificaciones por correo o SMS.

##### Requerimientos no funcionales (RNF) RNF-01. Seguridad de la información:

El sistema debe proteger los datos personales y sensibles de los usuarios mediante protocolos de seguridad como cifrado de contraseñas, validación de sesiones, protección contra inyecciones SQL y uso de HTTPS para la transmisión de datos.

##### RNF-02. Usabilidad:

El sistema debe contar con una interfaz amigable, intuitiva y accesible para usuarios con conocimientos técnicos básicos. Debe incluir validaciones visuales en formularios y mensajes de error claros.

##### RNF-03. Rendimiento:

El tiempo de respuesta del sistema no debe superar los tres segundos para la mayoría de las operaciones, tanto en consultas como en el registro de información.

##### RNF-04. Disponibilidad:

El sistema debe estar disponible el 99% del tiempo, garantizando acceso estable y continuo para los usuarios. Se implementarán respaldos automáticos para evitar pérdida de datos.

##### RNF-05. Escalabilidad:

El sistema debe estar diseñado para permitir la inclusión de nuevas funcionalidades y soporte para mayor cantidad de usuarios sin afectar su rendimiento.

##### RNF-06. Compatibilidad:

La aplicación debe ser funcional en los principales navegadores modernos (Google Chrome, Mozilla Firefox, Microsoft Edge) y en dispositivos móviles de distintas resoluciones.

##### RNF-07. Mantenibilidad:

El código fuente debe estar estructurado siguiendo buenas prácticas de programación, como separación entre HTML/PHP, CSS y JavaScript, uso de rutas absolutas, y nombramiento coherente de archivos y variables sin tildes ni caracteres especiales.

##### RNF-08. Portabilidad:

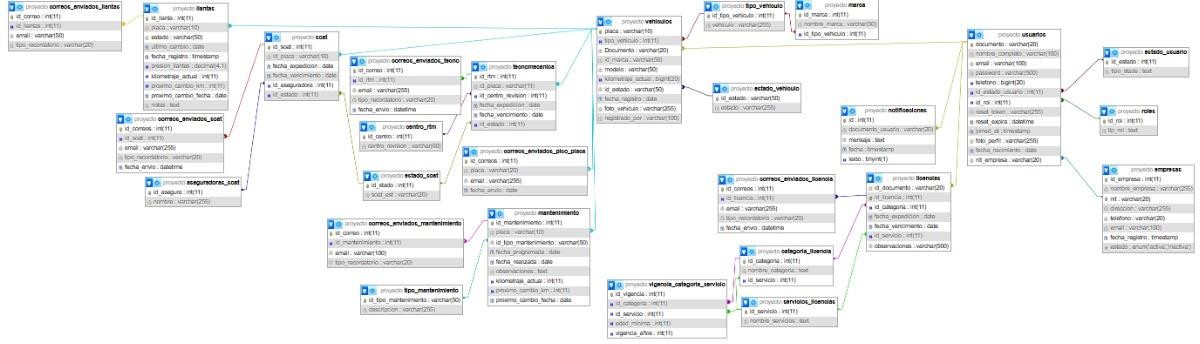
El sistema debe ser fácilmente migrable a diferentes entornos de servidor (por ejemplo, local en XAMPP o producción en Hostinger) sin necesidad de cambios estructurales complejos.

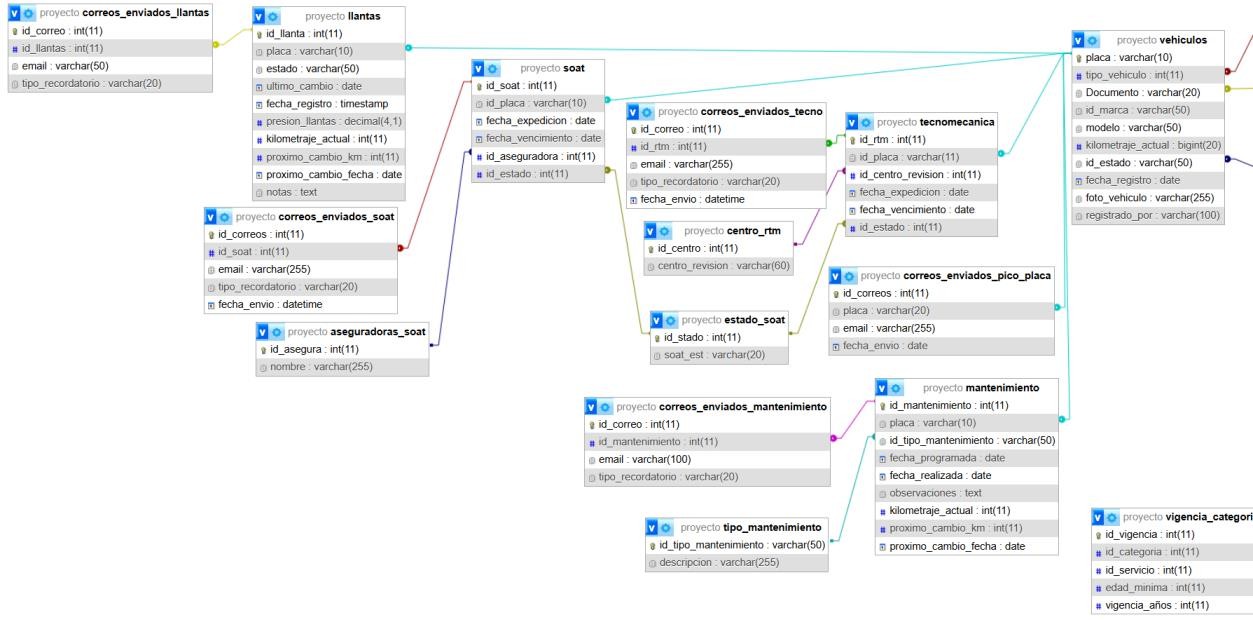
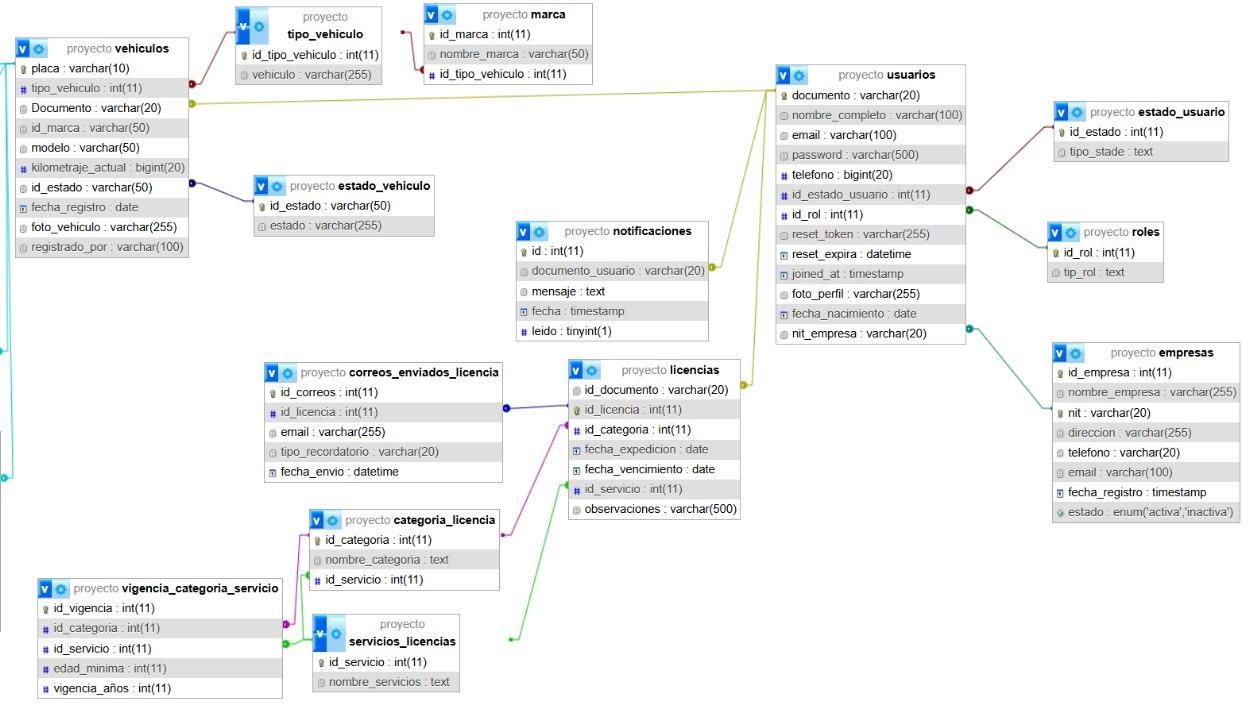
##### RNF-09. Accesibilidad:

La interfaz debe estar optimizada para ser utilizada en diferentes tamaños de pantalla, garantizando que se mantenga funcional y legible desde dispositivos móviles, tablets o computadores de escritorio.

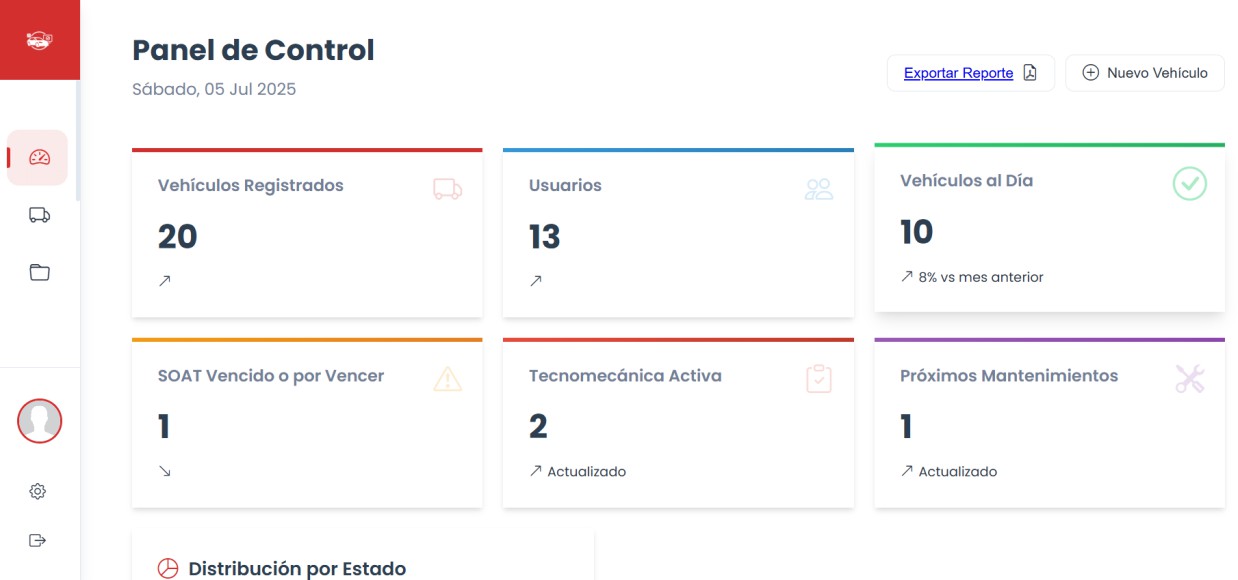
### Diseño

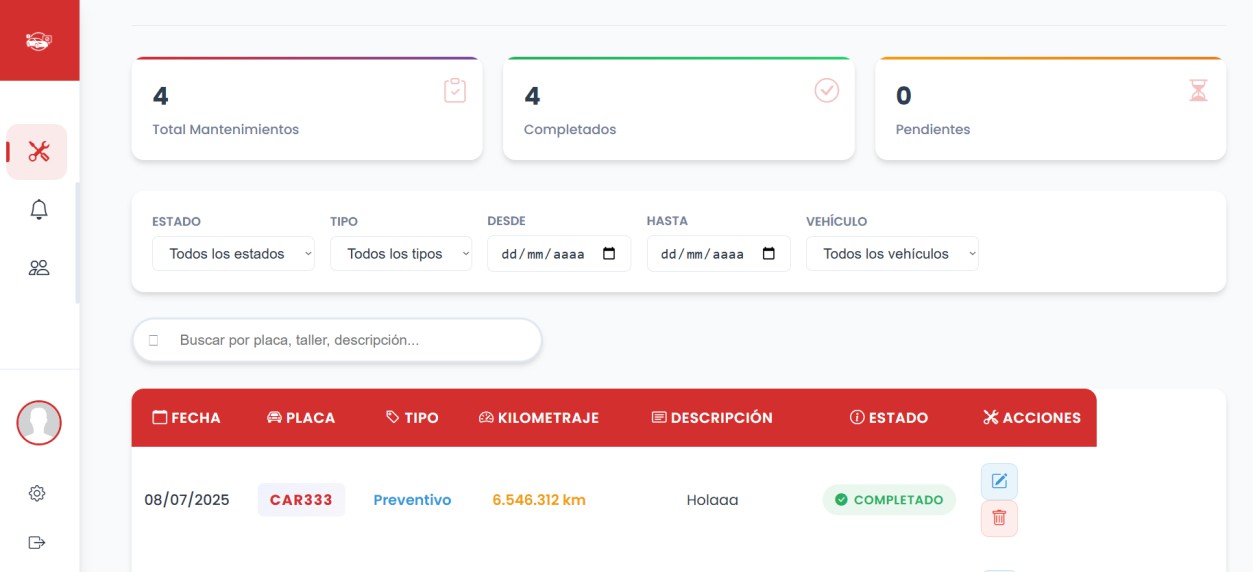
* + 1. **Mer**

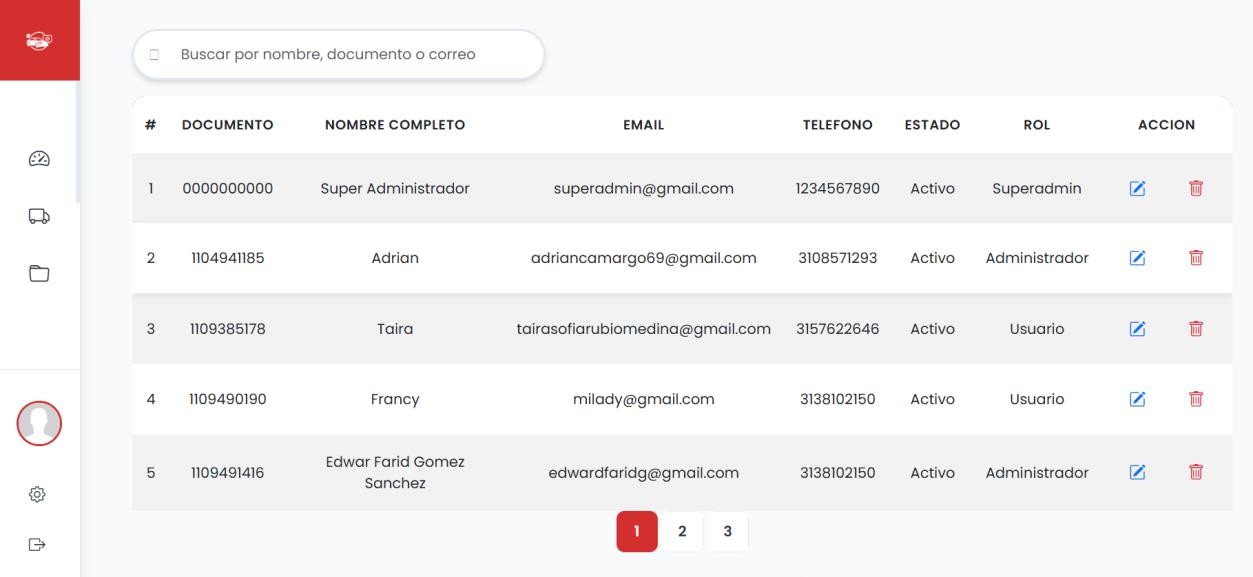


****

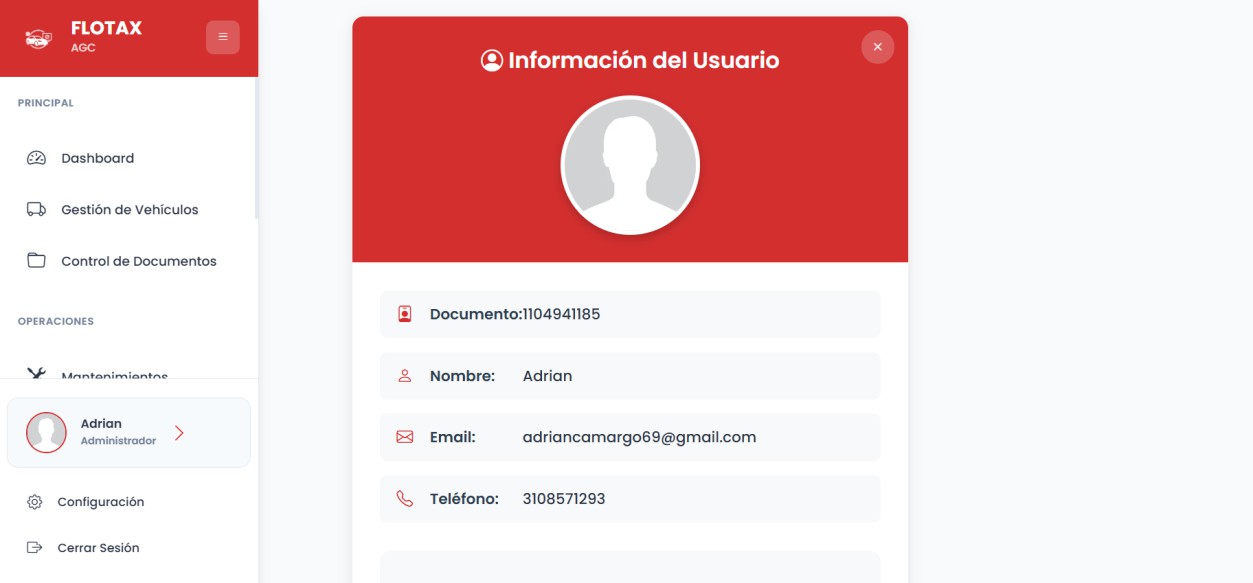
**6.2.3 Algunas interfaces administrador**

****



****



****

### Codificación

* Manual de desarrollador (Programador)

#### Pruebas

Ejemplos con validaciones

