



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE COMPUTACIÓN

**DESARROLLO DE UN SISTEMA PARA LA GESTIÓN Y EL MANTENIMIENTO
DEL PARQUE AUTOMOTOR DEL GAD MUNICIPAL DE AZOGUES**

Trabajo de titulación previo a la obtención del
título de Ingeniero en Ciencias de la Computación

AUTORES: DANIEL FERNANDO PINARGO OÑA
JOSÉ LUIS TAPIA VIDAL

TUTOR: ING. ERWIN JAIRO SACOTO CABRERA, PhD.

Cuenca - Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Daniel Fernando Pinargo Oña con documento de identificación N° 1721227179 y José Luis Tapia Vidal con documento de identificación N° 010620255; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

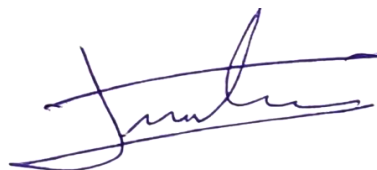
Cuenca, 29 de enero del 2024

Atentamente,



Daniel Fernando Pinargo Oña

1721227179



José Luis Tapia Vidal

010620255

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Daniel Fernando Pinargo Oña con documento de identificación N° 1721227179 y José Luis Tapia Vidal con documento de identificación N° 010620255, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Desarrollo de un sistema para la gestión y el mantenimiento del parque automotor del GAD Municipal de Azogues”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 29 de enero del 2024

Atentamente,



Daniel Fernando Pinargo Oña

1721227179



José Luis Tapia Vidal

010620255

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Erwin Jairo Sacoto Cabrera con documento de identificación N° 0301185229, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN SISTEMA PARA LA GESTIÓN Y EL MANTENIMIENTO DEL PARQUE AUTOMOTOR DEL GAD MUNICIPAL DE AZOGUES**, realizado por Daniel Fernando Pinargo Oña con documento de identificación N° 1721227179 y por José Luis Tapia Vidal con documento de identificación N° 0106620255, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 29 de enero del 2024

Atentamente,

**ERWIN JAIRO
SACOTO
CABRERA** Firmado digitalmente
por ERWIN JAIRO
SACOTO CABRERA
Fecha: 2024.01.30
14:36:51 -05'00'

Ing. Erwin Jairo Sacoto Cabrera, PhD.

0301185229

Dedicatoria y agradecimiento

Agradezco este trabajo a mi amada familia a mi Esposa Johanna, Mamá María, Hermana Ligia y Hermano Vinicio, gracias por ser el pilar y brindarme ese apoyo incondicional durante estos años de estudio.

A mis compañeros, por su colaboración, compañerismo y apoyo mutuo durante este viaje académico.

Daniel Fernando Pinargo Oña

A mis amados padres, Luis Tapia y Fanny Vidal, cuyo amor y apoyo han sido mi mayor inspiración. A todas las personas que han estado a mi lado, brindándome su apoyo y compañía, les agradezco profundamente. Este logro es también de ustedes. Gracias por creer en mí y por estar siempre presentes

José Luis Tapia Vidal

Resumen

El proyecto aborda la problemática del Gobierno Autónomo Descentralizado Municipal del Cantón Azogues (GAD Azogues), enfocado en mejorar la eficiencia en la administración y mantenimiento de su parque automotor. Se reconoce que el desarrollo de un país está intrínsecamente ligado al avance tecnológico de las instituciones públicas, que deben innovar para volverse competitivas. La innovación se caracteriza por la presentación de productos o servicios nuevos y mejorados, impulsados por tecnologías emergentes.

La transformación digital se presenta como clave para optimizar operaciones y crear nuevas oportunidades de negocios. En el contexto ecuatoriano, se destaca que la gestión del mantenimiento vehicular en instituciones como el GAD Azogues se ve afectada por limitaciones en recursos, dificultando la adopción de tecnologías de última generación.

La importancia del proyecto se fundamenta en resaltar que optimizar la gestión de datos y mantenimiento de vehículos tiene un impacto directo en la reducción de costos., el fortalecimiento de la seguridad y el incremento de la eficacia. Para el GAD Azogues, la necesidad de automatizar las tareas cotidianas se convierte en un aspecto crucial para potenciar la optimización de la operación, seguridad y transparencia. Actualmente, la asignación de vehículos y el seguimiento de los mantenimientos se llevan a cabo manualmente, lo que impulsa la sugerencia de crear un sistema completo e integrado.

Los objetivos del proyecto incluyen estudiar los fundamentos de programación relevantes, levantar requisitos, diseñar un sistema informático, y desarrollar módulos específicos para la gestión de datos y reserva de mantenimientos. La metodología Scrum es implementada, contando con un grupo conformado por un Product Owner, Scrum Master y desarrolladores.

En términos prácticos, se propone un diagrama lógico con capas de usuario, presentación, negocio y datos, utilizando tecnologías como IntelliJ IDEA, PostgreSQL y Angular. El cronograma detalla sprints con actividades específicas y se establece un presupuesto que incluye tecnología, servicios e imprevistos.

Palabras claves: Avance tecnológico, Competitividad, Innovación, Tecnología emergente, Recursos, Gestión de datos, Mantenimiento predictivo, Sustentabilidad, Rentabilidad, Operatividad, Confiabilidad, Metodología Scrum, Product Owner, Scrum Master, Scrum Team, Base de datos SQL, Java, PostgreSQL, Angular, IntelliJ IDEA.

Abstract

This project addresses the Municipal GAD (Government Autonomous Decentralized) issue in Azogues, focusing on the efficiency of managing and maintaining its vehicle fleet. It recognizes that a country's development is intrinsically linked to the technological advancement of public institutions, which must innovate to become competitive. Innovation entails the implementation of fresh and enhanced products or techniques prompted by emerging technologies.

Digital transformation emerges as a key element to optimize operations and create new business opportunities. In the Ecuadorian context, it is emphasized that vehicle maintenance management in institutions such as the GAD Azogues is hindered by resource limitations, hampering the adoption of state-of-the-art technologies.

The project justifies its relevance by highlighting that efficiency in data management and vehicle maintenance directly impacts cost reduction, improved safety, and increased productivity. For Azogues's GAD, automating daily processes becomes a crucial factor in enhancing operational efficiency, safety, and transparency. Vehicle reservations and maintenance control are carried out manually, motivating the proposal to develop a comprehensive system.

The project objectives include studying relevant programming fundamentals, gathering requirements, designing a computer system, and developing specific data management and maintenance reservations modules. The Scrum methodology is adopted, with a team consisting of a Product Owner, Scrum Master, and developers.

In practical terms, a logical diagram is proposed with user, presentation, business, and data layers, utilizing technologies such as IntelliJ IDEA, PostgreSQL, and Angular. The schedule details sprints with specific activities, and a budget is established, encompassing technology, services, and unforeseen circumstances.

Keywords: Technological Advancement, Competitiveness, Innovation, Emerging Technology, Resources, Data Management, Predictive Maintenance, Sustainability, Profitability, Operability, Reliability, Methodology, Scrum, Product Owner, Scrum Master, Scrum Team, SQL Database, Java, PostgreSQL, Angular, IntelliJ IDEA.

Índice de Contenido

1.	Introducción	14
2.	Problema	18
2.1	Antecedentes	18
2.2	Importancia y Alcances.....	18
2.3	Delimitación.....	19
3.	Objetivos Generales y Específicos.....	20
3.1	Objetivo General.....	20
3.2	Objetivos Específicos.....	20
4.	Revisión de la literatura o fundamentos teóricos.....	21
4.1	IntelliJ IDEA.....	21
4.2	PostgreSQL.....	24
4.3	Angular	27
5.	Marco metodológico.	31
5.1	Metodología De Desarrollo Scrum	31
5.1.1	Principales características de Scrum:.....	31
5.1.2	Actividades a Desarrollar.....	33
6.	Resultados	36
6.1	Levantamiento de requisitos.	36
6.2	Diseño del sistema informático.....	37

6.2.1	BackEnd.....	37
6.2.2	PostgreSQL.....	42
6.2.3	FrontEnd.....	44
6.3	Implementación del BackEnd.....	46
6.4	Diagrama lógico.....	49
6.5	Funciones del Sistema.....	50
6.5.1	Gestión de Vehículos y Empleados.....	50
6.5.2	Desarrollo del sitio Web.....	52
7.	Cronograma.....	60
8.	Presupuesto.....	62
9.	Conclusiones.....	63
10.	Recomendaciones.....	65
11.	Referencias bibliográficas.....	67
12.	Anexos.....	70
	Anexo 1.....	70
	Anexo 2.....	84
	Anexo 3.....	102

Índice de Figuras

Figura 1 Arquitectura del Backend.....	38
Figura 2 Estructura de tablas de datos	42
Figura 3 Entidad.....	47
Figura 4 Repositorio	48
Figura 5 Controlador.....	49
Figura 6 Diagrama Lógico.....	50
Figura 7 Clave Secreta (Jasyp)	51
Figura 8 Simplificado con Lombok.....	52
Figura 9 Pantalla de Inicio.....	53
Figura 10 Pantalla de Inicio de Sesión.....	53
Figura 11 Pantalla de bienvenida.....	54
Figura 12 Pantalla de las Personas.....	55
Figura 13 Pantalla de Vehículos	56
Figura 14 Pantalla de Reservas.....	56
Figura 15 Pantalla de Sugerencias.....	57
Figura 16 Pantalla de Mantenimientos	58
Figura 17 Pantalla de Repuestos.....	59

Índice de Tablas

Tabla 1 Sprint 1	34
Tabla 2 Sprint 2	34
Tabla 3 Sprint 3	34
Tabla 4 Sprint 4	35
Tabla 5 Sprint 5	35
Tabla 6 Cronograma	60
Tabla 7 Presupuesto	62

1. Introducción

La Transformación Digital ha surgido como un fenómeno empresarial y social de suma importancia en la era actual, fomentando transformaciones de considerable importancia en cuanto a cómo las organizaciones funcionan, se relacionan y proporcionan beneficios a sus clientes (Sacoto Cabrera & Perez Torres, 2023) . En este contexto, la combinación de tecnologías clave, tales como las redes de dispositivos conectados a Internet (IoT) y la adopción del protocolo de comunicación inalámbrica 5G, ha desempeñado un papel crucial en la redefinición de los paradigmas empresariales (Aranda, Sacoto-Cabrera, Haro-Mendoza, & Astudillo-Salinas, 2021). En este contexto, la integración de tecnologías fundamentales, como las redes de Internet de las Cosas (IoT) y la implementación del estándar de comunicación inalámbrica 5G, ha tenido un papel fundamental en la reconfiguración de los modelos de negocios (Sacoto-Cabrera, León-Paredes, & Verdugo-Romero, 2021). Las infraestructuras de IoT han posibilitado la interconexión inteligente de dispositivos, sensores y objetos cotidianos a través de la Internet, posibilitando una recopilación masiva de datos en tiempo real. Esta interconexión transforma no solo la forma en que interactuamos con el entorno, además de esto, también influye en la forma en que las empresas llevan a cabo sus operaciones y realizan sus elecciones estratégicas. La abundancia de datos generados por los dispositivos IoT ha facilitado la automatización, el uso de análisis avanzado y la toma de decisiones respaldada por datos, elementos fundamentales en el camino hacia la eficiencia y la agilidad empresarial (Sacoto-Cabrera, Castillo, Pauta, Trelles, & Guambaña, 2021).

Por otro lado, la implementación de redes 5G ha elevado la conectividad a nuevas alturas, proporcionando velocidades de transferencia de datos extremadamente rápidas y una latencia

mínima. Esta revolución en las comunicaciones inalámbricas no solo ha mejorado la experiencia del usuario, sino que también ha posibilitado aplicaciones y servicios hasta ahora inimaginables. La alta velocidad y capacidad de la red 5G son esenciales para impulsar la adopción masiva de tecnologías emergentes, como la realidad virtual, la inteligencia artificial y el procesamiento de datos en el borde, lo que contribuye significativamente a la Transformación Digital. En este contexto tecnológico, la Transformación Digital no solo se trata de adoptar nuevas tecnologías, sino también de repensar los modelos de negocio. Las empresas han comenzado a explorar modelos centrados en los datos, servicios basados en suscripciones, experiencias personalizadas y ecosistemas colaborativos. La capacidad para adaptarse y evolucionar en este paisaje digital en constante cambio se ha vuelto crucial para la permanencia y el logro sostenido de las organizaciones a largo plazo.

Las redes de Internet de las cosas (IoT) han posibilitado la conexión inteligente de dispositivos, sensores y objetos comunes a través de la web, permitiendo la recopilación masiva de información en tiempo real. Esta interconexión no solo altera la manera en que interactuamos con nuestro entorno, sino también cómo las empresas llevan a cabo sus operaciones y toman decisiones. El volumen considerable de información producida por los dispositivos de IoT ha simplificado la automatización, el análisis sofisticado y la toma de decisiones fundamentada en información recopilada, elementos esenciales para mejorar la eficacia y la capacidad de adaptación en el entorno institucional.

Por otro lado, la implementación de redes 5G ha llevado la conectividad a niveles sin precedentes, ofreciendo velocidades de transferencia de datos sumamente rápidas y una latencia mínima (Sacoto Cabrera, Gujjarro, & Maillé, 2022). Este avance en las comunicaciones inalámbricas ha potenciado la experiencia del usuario y ha abierto la puerta a aplicaciones y

servicios que antes parecían inimaginables. La rapidez y capacidad de la red 5G resultan fundamentales para fomentar la adopción generalizada de tecnologías emergentes, como la realidad virtual, la inteligencia artificial y el procesamiento de datos en el borde, contribuyendo de manera significativa a la Transformación Digital.

En este contexto tecnológico, la Transformación Digital implica más que simplemente incorporar nuevas tecnologías; implica también replantear los modelos de negocio. Las compañías están empezando a explorar enfoques centrados en los datos, ofrecer servicios mediante suscripciones, proporcionar experiencias personalizadas y desarrollar colaboraciones dentro de ecosistemas. En este entorno digital dinámico y en continua evolución, la habilidad para ajustarse y progresar se ha vuelto esencial para la viabilidad y el cumplimiento de los objetivos a largo plazo de las organizaciones (Rodríguez Alvarado & Sacoto-Cabrera, 2021).

En el contexto actual de continua evolución tecnológica, el desarrollo de un país está estrechamente vinculado al progreso y la innovación de sus entidades gubernamentales. La eficacia en la operación y la capacidad para adaptarse a las últimas tendencias tecnológicas son elementos cruciales para asegurar la competitividad y la relevancia en el mundo empresarial. Este principio se manifiesta de manera significativa en el ámbito público, donde la integración de tecnologías emergentes se convierte en un componente determinante para la optimización de métodos y servicios.

En este contexto, el estudio actual se enfoca en un asunto de gran importancia detectado en el ámbito del Gobierno Autónomo Descentralizado (GAD) Azogues, Ecuador. Se aborda la gestión y mantenimiento de su parque automotor, reconociendo la relevancia de la digitalización y la innovación en este proceso. La necesidad de mejorar la administración de vehículos, sus

servicios de mantenimiento y reservas se establece como objetivo de este proyecto, cuyo objetivo es desarrollar un sistema integral para gestionar eficazmente estos aspectos.

Exploraremos la conexión entre la digitalización, la administración de datos y la innovación en la gestión de flotas vehiculares, subrayando cómo estas acciones pueden resultar en mejoras sustanciales en la eficacia operativa, la seguridad y la transparencia en las instituciones públicas. Además, se analizará la implementación de metodologías ágiles, específicamente la metodología Scrum, para dirigir el desarrollo de este sistema, garantizando la capacidad de ajustarse a los cambios y proporcionar de manera constante valor agregado.

A través de este estudio, se pretende ofrecer una solución completa que no solo enfrente los problemas detectados en el GAD Azogues, sino que también establezca un precedente para la modernización de prácticas similares en otras instituciones. El desarrollo de este sistema no solo contribuirá a la eficacia operativa del GAD Azogues, sino que también sienta las bases para la modernización de prácticas similares en otras instituciones, mostrando cómo la combinación de tecnología, innovación y metodología ágil puede generar un cambio considerable.

2. Problema

2.1 Antecedentes

El GAD Azogues, como entidad pública encargada de brindar servicios a la comunidad, enfrenta desafíos significativos en la gestión de su parque automotor. El problema se origina en ausencia de una solución tecnológica efectiva que posibilite el manejo completo de datos y el mantenimiento de los vehículos asignados a la institución. Actualmente, los procesos de reserva de vehículos y control de mantenimientos se llevan a cabo de manera manual, lo que resulta en una gestión ineficiente y propensa a errores.

2.2 Importancia y Alcances

La jerarquía de abordar este problema está en su impacto en la eficiencia operativa, seguridad y transparencia del GAD Azogues. La falta de una solución tecnológica afecta la calidad del servicio, aumenta los costos operativos y dificulta la toma de decisiones fundamentadas. La dimensión del problema se evidencia en la continua necesidad de mejorar la administración de flotas vehiculares en instituciones similares a nivel nacional.

Datos estadísticos indican que el mantenimiento ineficiente de vehículos puede llevar a un aumento significativo en los costos operativos anuales. Además, informes internos del GAD Azogues han identificado la falta de seguimiento efectivo en la programación de mantenimientos, lo que resulta en vehículos no disponibles cuando se necesitan, afectando la operatividad de la institución.

La importancia de este proyecto de grado es fundamental. La implementación de un sistema destinado a la gestión y mantenimiento del parque automotor posibilitará una administración más

efectiva de los recursos, agilizará la realización de tareas, mejorará la calidad del servicio y reducirá los gastos operativos. El beneficio se extiende no solo al GAD Azogues, sino también en otras instituciones públicas que enfrentan desafíos similares en la gestión de sus flotas vehiculares.

Documentos oficiales y diagnósticos previos han resaltado la importancia de adoptar tecnologías modernas para incrementar la eficacia de las instituciones públicas. La falta de una solución tecnológica se manifiesta en la pérdida de tiempo en procesos manuales, en la falta de información actualizada y en la dificultad para tomar decisiones informadas sobre la gestión vehicular.

2.3 Delimitación

La intervención propuesta se circunscribe geográficamente al ámbito del GAD Azogues, limitando su implementación al contexto institucional de esta entidad. Durante un tiempo determinado de desarrollo y pruebas, el proyecto será ejecutado temporalmente, con fechas de inicio y término claramente definidas. El enfoque sectorial se concentra exclusivamente en la gestión de vehículos dentro de entidades gubernamentales y municipales, centrándose en las necesidades específicas de esta institución. La delimitación institucional asegura que la solución propuesta se ajuste de manera precisa a los procesos y requisitos particulares del GAD Azogues, sin perder de vista la posibilidad de replicabilidad en instituciones similares a nivel nacional.

3. Objetivos Generales y Específicos.

3.1 Objetivo General

Desarrollar un sistema para la gestión y el mantenimiento de los vehículos del GAD Azogues.

3.2 Objetivos Específicos

- **OE1:** Estudiar los fundamentos de programación de Java, Angular y SQL.
- **OE2:** Desarrollar un módulo para la gestión de datos de los vehículos orientado al administrador y a los empleados del GAD Azogues.
- **OE3:** Desarrollar un módulo para la reserva del mantenimiento de vehículos orientado al administrador y a los empleados del GAD Azogues.
- **OE4:** Desarrollar y ejecutar un plan de pruebas funcionales para validar el funcionamiento del sistema.

4. Revisión de la literatura o fundamentos teóricos.

A continuación, se exponen los principios teóricos más destacados del proyecto, con el propósito de satisfacer el Objetivo 1 de este trabajo de titulación, que consiste en "Analizar los fundamentos de programación en Java, Angular y SQL".

4.1 IntelliJ IDEA

Según (IntelliJ IDEA, 2023), se trata de un Entorno de Desarrollo Integrado (IDE) creado por JetBrains, que se destaca por su gran versatilidad y su amplia adopción en la industria. Ofrece un robusto editor de código con funciones como resaltado de sintaxis y autocompletado inteligente, así como una amplia gama de herramientas diseñadas para potenciar la productividad y la eficacia en el proceso de perfeccionamiento de software. Entre estas características se incluyen soporte para múltiples lenguajes de programación, capacidades de refactorización, depuración y perfilado, unificación con sistemas de control de versiones y herramientas de construcción, una extensa colección de complementos y extensiones, utilidades para el desarrollo web y en la nube, y una activa comunidad de usuarios y desarrolladores respaldando lo antes mencionado.

A continuación, se proporciona una descripción técnica de IntelliJ IDEA y sus principales aplicaciones:

- **Interfaz de Usuario:**
 - Editor de Código Inteligente: IntelliJ IDEA ofrece un editor de código inteligente con funciones como resaltado de sintaxis, autocompletado de código, refactorización y navegación rápida.

- Inspección de Código: Detecta y resalta posibles errores, problemas de estilo y ofrece sugerencias de mejora.
- Explorador de Proyectos: Permite una fácil navegación y gestión de archivos y proyectos.
- **Gestión de Proyectos:**
 - Sistema de Construcción Integrado: Soporta sistemas de construcción como Maven y Gradle, facilitando la configuración y gestión de dependencias.
 - Integración con VCS (Sistema de Control de Versiones): Compatible con sistemas como Git, SVN, y otros, permitiendo un seguimiento y control eficientes de las versiones del código fuente (Loja, Molina, Loja, & Cañarte, 2020).
- **Desarrollo de Aplicaciones Java:**
 - Compatibilidad Total con Java: Proporciona un entorno optimizado para el desarrollo en Java, con soporte completo para las últimas versiones del lenguaje.
 - Depuración Avanzada: Ofrece herramientas de depuración potentes, incluyendo puntos de interrupción, inspección de variables y seguimiento de ejecución.
- **Soporte para Otros Lenguajes:**
 - Kotlin: IntelliJ IDEA es conocido por su fuerte integración con Kotlin, el lenguaje de programación desarrollado por JetBrains.
 - Otros Lenguajes y Tecnologías: Ofrece soporte para una variedad de lenguajes y tecnologías, como JavaScript, TypeScript, HTML, CSS, SQL, entre otros.

- **Desarrollo Web:**

- Compatibilidad con Frameworks Web: Admite frameworks populares como Spring, Java EE, y otros, simplificando el proceso de creación de aplicaciones en web.
- Herramientas de Desarrollo Frontend: Incluye herramientas para el perfeccionamiento frontend, como soporte para Angular, React, y Vue.js.

- **Herramientas de Productividad:**

- Generación Automática de Código: Suministra la creación rápida de código con funciones de generación automática, como la creación de getters y setters.
- Refactorización de Código: Ofrece una diversidad de elecciones de refactorización para optimizar la estructura y legibilidad del código.

- **Pruebas y Análisis:**

- Soporte para Pruebas: Simplifica la elaboración y realización de pruebas tanto unitarias como de integración.
- Herramientas de Análisis de Código: Proporciona instrumentales de estudio estático y dinámico para mejorar el código.

- **Integración con Servidores de Aplicaciones y Bases de Datos:**

- Despliegue y Configuración de Servidores: Facilita el despliegue de aplicaciones en servidores como Tomcat y JBoss.

- Herramientas de Bases de Datos: Contiene herramientas para dirigir bases de datos y realizar consultas en SQL (Vladislav & Timofey, 2018).
- **Plug-ins y Extensibilidad:**
 - Amplia Gama de Plug-ins: Permite la instalación de numerosos plug-ins para ampliar las capacidades del IDE.
 - Desarrollo de Plug-ins Personalizados: Ofrece herramientas para el desarrollo de plug-ins personalizados según las necesidades del desarrollador.

4.2 PostgreSQL

Es un sistema de gestión de bases de datos relacionales de código abierto reconocido por su robustez, capacidad de adaptación y cumplimiento de estándares SQL. Sobresale por su soporte a transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), así como por conservar la integridad de los datos y ser fácilmente escalable, permitiendo a los usuarios definir funciones y tipos de datos personalizados. Es apropiado para una variedad de aplicaciones, desde pequeñas plataformas web hasta sistemas empresariales de gran envergadura. Su licencia de código abierto y la participación de una comunidad de desarrolladores y usuarios lo convierten en una opción sumamente popular en el campo de las bases de datos (PostgreSQL, 2023).

Ahora, se ofrece una descripción técnica detallada de PostgreSQL:

- **Modelo de Datos:**
 - Relacional: PostgreSQL sigue el tipo de datos relacional, basado en tablas que contienen filas y columnas (Vazquez, Mier, & Sotolongo, 2016).

- Integridad Referencial: Admite la integridad referencial, permitiendo definir y mantener relaciones entre tablas.

- **Lenguaje de Consulta:**
 - SQL Estándar: Soporta el estándar SQL ANSI, proporcionando una amplia gama de funciones y comandos para consultar y manipular datos.

- **Arquitectura:**
 - Arquitectura Cliente-Servidor: PostgreSQL utiliza un diseño cliente-servidor, donde múltiples clientes pueden conectarse al servidor simultáneamente (Ginestà & Pérez, 2009).
 - Procesos de Fondo: Emplea procesos de fondo para tareas como la limpieza, la replicación y la gestión de transacciones.

- **Transacciones y Concurrency:**
 - Control de Transacciones ACID: Garantiza la consistencia y fiabilidad de las transacciones, siguiendo los principios ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
 - Control de Concurrency: Aplica métodos de bloqueo y gestión de la concurrencia para controlar el acceso simultáneo a los datos y prevenir posibles conflictos.

- **Tipos de Datos:**

- Tipos de Datos Estándar: Brinda una diversa gama de tipos de datos convencionales, como números enteros, decimales, cadenas de caracteres, fechas y otros más (Domínguez, 2019).
- Tipos de Datos Personalizados: Facilita la generación de formatos de datos adaptados y añadidos conforme a las exigencias del usuario.
- **Índices y Optimización:**
 - Índices: Admite la creación de índices para perfeccionar el rendimiento de las consultas y búsquedas.
 - Estadísticas y Optimizador de Consultas: Emplea datos estadísticos exhaustivos y un optimizador de consultas para aumentar la eficacia en la realización de las consultas.
- **Seguridad:**
 - Autenticación y Autorización: Ofrece mecanismos de autenticación segura y controles de autorización para salvaguardar el ingreso a la base de datos.
 - SSL/TLS: Proporciona soporte para conexiones seguras a través de SSL/TLS.
- **Replicación y Alta Disponibilidad:**
 - Replicación Nativa: Proporciona soporte nativo para la replicación, permitiendo la creación de copias de bases de datos y la distribución de la carga de trabajo.
 - Failover Automático: Puede configurarse para implementar failover automático y garantizar la alta disponibilidad.

- **Extensibilidad y Ecosistema:**

- Extensiones y Módulos: Permite la creación de extensiones y módulos para ampliar las capacidades de PostgreSQL.
- Ecosistema Rico: Dispone de una comunidad participativa, una extensa gama de herramientas y recursos, y es compatible con múltiples lenguajes de programación.

4.3 Angular

Según el sitio web (Angular, 2023), Angular se describe como un framework destinado al desarrollo tanto de aplicaciones web como móviles. Este framework emplea TypeScript, una extensión de JavaScript que incorpora funcionalidades significativas para las aplicaciones empresariales, como la tipificación de datos y los decoradores. Además, TypeScript ofrece una sólida verificación de tipos que contribuye al descubrimiento temprano de errores y simplifica la reorganización del código. Esta herramienta facilita la ejecución de tareas habituales en el desarrollo de aplicaciones y aplica una guía de estilo para asegurar la uniformidad en los componentes. Angular incluye diversas bibliotecas de componentes que permiten adaptar las aplicaciones según los requisitos de la marca.

A continuación, se ofrece una explicación técnica sobre Angular:

- **Lenguaje de Programación:**

- TypeScript: Angular se encuentra desarrollado en TypeScript, una extensión de JavaScript que incorpora tipado estático opcional y otras funcionalidades contemporáneas al lenguaje.

- **Arquitectura:**
 - Arquitectura de Componentes: Angular sigue un diseño basado en componentes, donde la interfaz de usuario se construye mediante la composición de componentes reutilizables.
 - Utilización de un sistema de inyección de dependencias para administrar las relaciones entre los distintos componentes y servicios.
- **MVVM (Model-View-ViewModel):**
 - Modelo de Datos: Angular utiliza un modelo de datos para representar la información y el estado actual de la aplicación.
 - Vista (View): La vista se representa mediante componentes y plantillas HTML.
 - ViewModel: Angular utiliza controladores o componentes para gestionar la lógica de la vista y la manipulación de datos.
- **Directivas:**
 - Directivas Incorporadas: Angular incluye directivas incorporadas para manipular el DOM, como ngIf, ngFor, ngSwitch, etc.
 - Creación de Directivas Personalizadas: Permite la creación de directivas personalizadas para extender el comportamiento del DOM.
- **Servicios:**
 - Servicios: Angular utiliza servicios para encapsular la lógica de negocio, compartir datos entre componentes y gestionar la comunicación con servidores.

- Inyección de Dependencias para Servicios: Los servicios se inyectan en componentes y otros servicios para su uso.

- **Routing:**
 - Enrutamiento: Angular ofrece un sistema de enrutamiento diseñado para administrar la transición entre diversas vistas dentro de una aplicación de una sola página (SPA).
 - Lazy Loading: Permite la carga perezosa de módulos para mejorar el rendimiento al cargar solo los componentes necesarios (Medrano & Ali, 2022).

- **HTTP Client:**
 - Módulo HTTP: Incluye un módulo HTTP para realizar solicitudes HTTP y manejar respuestas de forma asíncrona (Azaustre, 2015).
 - Interceptores HTTP: Permite la manipulación de solicitudes y respuestas HTTP mediante interceptores.

- **Testing:**
 - Herramientas de Pruebas: Angular proporciona herramientas integradas para realizar pruebas unitarias y de integración.
 - Jasmine y Karma: Se utilizan herramientas como Jasmine para escribir pruebas y Karma para ejecutarlas.

- **Integración con Bibliotecas Externas:**

- Compatibilidad con Librerías de Terceros: Angular puede integrarse fácilmente con bibliotecas y herramientas externas gracias a su estructura modular y sistema de construcción (Wilken, 2018).

5. Marco metodológico.

5.1 Metodología De Desarrollo Scrum

Scrum es un método ágil utilizado en el desarrollo de software que destaca por su enfoque en la adaptabilidad y el trabajo en equipo para resolver desafíos complejos de manera gradual y repetitiva. Este enfoque fue ideado por Jeff Sutherland y Ken Schwaber en los años 90 y ha ganado gran popularidad como uno de los métodos ágiles más utilizados en el desarrollo de software y otros tipos de proyectos.

5.1.1 Principales características de Scrum:

5.1.1.1 Iterativo e Incremental

Scrum descompone la labor en etapas de desarrollo conocidas como "sprints". Cada sprint representa un lapso establecido de tiempo, generalmente de dos a cuatro semanas, en el que se genera un avance en la funcionalidad del producto.

5.1.1.2 Roles Definidos

- **Product Owner:** Expresa los intereses del cliente y establece los requisitos del producto, además de tomar decisiones acerca de las prioridades en la lista de tareas pendientes del mismo.
- **Scrum Master:** Simplifica el procedimiento de Scrum, resuelve desafíos y contribuye a mantener un ambiente de desarrollo efectivo (Estrada Velasco, Nuñez Villacis, Saltos Chávez, & Cunuhay Cuchipe, 2021).
- **Equipo de Desarrollo:** Profesionales responsables de proporcionar el incremento del producto al término de cada iteración (Estayno & Meles, 2014).

5.1.1.3 Backlog del Producto: El Product Owner es encargado de administrar una lista ordenada de todas las características, mejoras y correcciones esenciales para el producto, ajustando las prioridades según los requerimientos del cliente.

4. Planificación del Sprint: Al comienzo de cada sprint, el equipo de desarrollo elige un conjunto de elementos del backlog para abordar durante ese período. Esto constituye el "sprint backlog".

5. Reuniones Diarias (Daily Scrum): Reuniones cortas diarias destinadas a que el equipo de desarrollo comparta su avance, detecte posibles problemas y coordine acciones para alcanzar los objetivos establecidos para el sprint.

6. Revisión del Sprint y Retrospectiva: Después de cada ciclo de desarrollo, se realiza una revisión en la que el equipo presenta el trabajo realizado al Propietario del Producto y otros involucrados. Luego, se lleva a cabo una reflexión para analizar el proceso y buscar oportunidades de mejora constante.

7. Entregables Incrementales: Al concluir cada iteración, el equipo genera un avance que puede ser entregado, lo que implica que el producto está operativo y listo para ser lanzado si así se requiere.

8. Flexibilidad y Adaptabilidad: Scrum se destaca por su enfoque flexible y su habilidad para adaptarse a cambios. En caso de que surjan nuevos requisitos o se reconozcan mejoras durante el proceso de desarrollo, estos pueden ser integrados en sprints posteriores.

Scrum ha demostrado su eficacia en diversos ámbitos además del desarrollo de software, abarcando la gestión de proyectos, marketing y otras esferas en las que la adaptabilidad y la capacidad de respuesta ágil ante cambios son esenciales. Implementar Scrum de manera efectiva implica una estrecha colaboración entre los integrantes del equipo y una mentalidad ágil que promueva la mejora continua.

De acuerdo con la investigación de (Ramírez, Soto , Moreno, Rojas, & Milán , 2019), la implementación de una metodología Scrum ofrece una gran cantidad de beneficios que son muy significativos. Estas capacidades abarcan la habilidad de adaptarse a los cambios durante la evolución del proyecto, lo cual promueve la flexibilidad. También se promueve la transparencia mediante la utilización de herramientas como el Scrumboard y el Sprint Burndown chart, facilitando así un ambiente de trabajo abierto que permite una retroalimentación constante.

La división del proyecto en entregables permite una evaluación y mejora progresiva. Los proyectos SCRUM también se caracterizan por una resolución más ágil de problemas, gracias a equipos multifuncionales que fomentan la colaboración y la proximidad física.

5.1.2 Actividades a Desarrollar

Roles Scrum:

Product Owner: Representante del GAD municipal de azogues

Scrum Master: Ing. Erwin Jairo Sacoto Cabrera (ES)

Scrum Team:

Developer N° 1: Pinargo Oña Daniel Fernando (DP)

Developer N° 2: Tapia Vidal José Luis (JT)

5.1.2.1 Sprint 1

Tabla 1 Sprint 1

OE1
ACT.1 Estudiar los fundamentos de programación de Java, Angular y SQL.
OE2
ACT.1 Iniciar el levantamiento de requisitos para el desarrollo del sistema de gestión y mantenimiento del parque automotor del GAD Azogues.

5.1.2.2 Sprint 2

Tabla 2 Sprint 2

OE2
ACT.2 Continuar y completar el levantamiento de requisitos.
OE3
ACT.1 Iniciar el diseño del sistema informático, considerando las tecnologías Java, Angular y SQL.

5.1.2.3 Sprint 3

Tabla 3 Sprint 3

OE3
ACT.2 Continuar y finalizar el diseño del sistema.

OE4
ACT.1 Comenzar el desarrollo del módulo para la gestión de datos de los vehículos orientado al administrador y a los empleados del GAD Azogues.

5.1.2.4 Sprint 4

Tabla 4 Sprint 4

OE4
ACT.2 Continuar y completar el desarrollo del módulo de gestión de datos de vehículos.
ACT.3 Iniciar el desarrollo del módulo de reserva del mantenimiento de vehículos.

5.1.2.5 Sprint 5

Tabla 5 Sprint 5

OE5
ACT.1 Continuar y finalizar el desarrollo del módulo de reserva de mantenimiento de vehículos.
OE6
ACT.1 Desarrollar y ejecutar un plan de pruebas funcionales para validar el funcionamiento del sistema.
OE7
ACT.1 Realizar el proceso de transferencia tecnológica y entrega de manuales.

6. Resultados

Es este apartado, se describen los resultados del desarrollo e implementación del sistema para gestión y administración del parque automotor del GAD Azogues, esto permitirá cumplir los objetivos 2, 3 y 4 planteados en esta tesis.

6.1 Levantamiento de requisitos.

Como parte esencial del desarrollo de este proyecto, se propuso llevar a cabo un relevamiento de requisitos para detectar, analizar y documentar minuciosamente tanto las necesidades como las especificaciones funcionales y no funcionales que el sistema debe cumplir para alcanzar los objetivos establecidos.

Este procedimiento es esencial para obtener una comprensión exhaustiva de las necesidades de los usuarios finales, los requisitos del negocio, y las restricciones tecnológicas del proyecto, es relevante señalar que este archivo no solo cumplirá la función de ser una guía de referencia para el equipo de desarrollo, sino que también se utilizará como un manual para validar y verificar los requisitos por parte de los stakeholders del proyecto.

Para realizar la etapa de recolección de requerimientos, se realizó una reunión inicial con representantes del GAD Azogues a través de la plataforma zoom. Durante esa sesión, se discutieron los procesos actuales utilizados para gestionar los mantenimientos de forma manual. Se discutieron los desafíos y obstáculos encontrados con estos métodos actuales, así como las áreas en las que se buscaba mejorar y optimizar el proceso.

También, se exploraron las necesidades específicas del GAD Azogues en cuanto a las funcionalidades y requisitos para el nuevo sistema de gestión del parque automotor. Esta reunión

inicial fue para comprender completamente el entorno operativo del cliente y sus expectativas, y de esta manera crear soluciones adaptadas a sus necesidades específicas.

Además, se proporcionó un documento en formato Excel el cual sirvió como ejemplo de cómo se llevaban a cabo el agendamiento de los mantenimientos manualmente, de este modo, se pudo comprender mejor sus prácticas y necesidades específicas, y así reconocer y registrar los requisitos operativos y no operativos, estableciendo una base sólida para la creación, elaboración e introducción del sistema, asegurando su alineación con las necesidades y deseos del GAD Azogues.

El documento completo se encuentra disponible en la sección de Anexos en la subsección Anexo 1.

6.2 Diseño del sistema informático

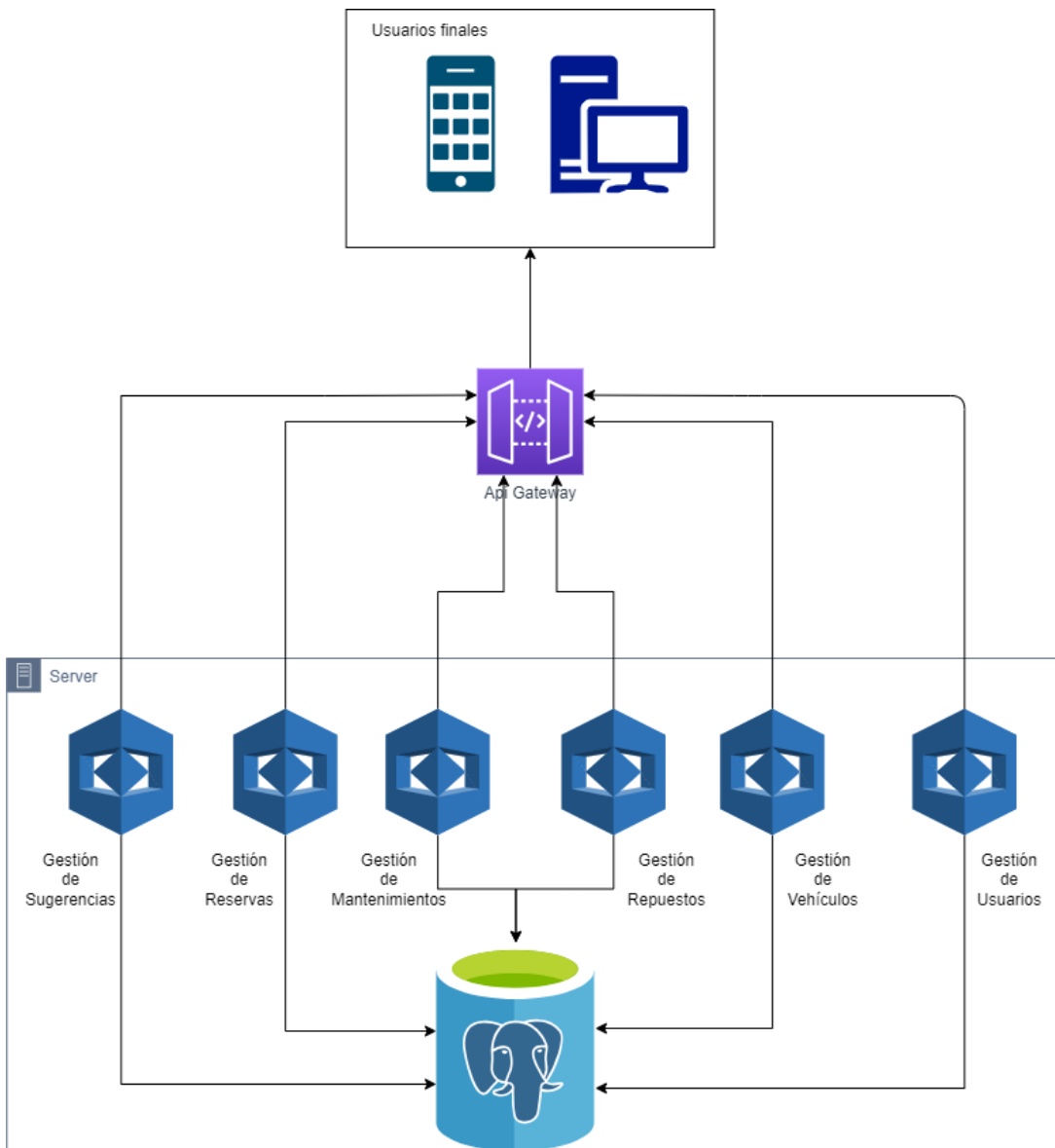
En esta sección se presenta la estructura empleada en la construcción del backend, la cual ha sido creada para asegurar la eficacia, capacidad de adaptación y seguridad del sistema, con el fin de cumplir con los requisitos funcionales y no funcionales.

6.2.1 *BackEnd*

Dentro del marco del proyecto, el backend cumple una función esencial en el desarrollo y control de las funciones principales del sistema, incluyendo la aplicación de la lógica empresarial, la administración de la base de datos, la seguridad y la provisión de Interfaces de Programación de Aplicaciones (APIs) para la interacción con el frontend y otros sistemas externos, asegurando su operatividad eficiente, segura y confiable.

A continuación, en la figura 1 se muestra la arquitectura empleada, la cual se fundamenta en un enfoque de microservicios para favorecer el modularidad, flexibilidad en el desarrollo y mantenimiento del sistema.

Figura 1 Arquitectura del Backend



El sistema se representa a través de un diagrama que ilustra sus componentes principales. Estos elementos permiten la operatividad y eficiencia del sistema en la administración de vehículos y actividades asociadas.

- **Usuarios Finales:** Los usuarios finales, incluyendo conductores, mecánicos y personal administrativo, son aquellos que interactúan directamente con el sistema, realizando tareas específicas relacionadas con la gestión del parque automotor.
- **API Gateway:** El API Gateway actúa como una puerta de enlace centralizada que permite a los usuarios finales acceder a los diversos servicios del sistema de manera uniforme y eficiente.
- **Servidor:** El servidor constituye el núcleo del sistema, almacenando y recuperando datos desde la base de datos.

Además, abarca la gestión de diversas áreas, como sugerencias, reservas, mantenimientos, repuestos, vehículos y usuarios, cada una con sus propias funcionalidades específicas.

- **Base de Dato:** La base de datos es fundamental para el almacenamiento de datos del sistema, alberga la información necesaria para el correcto funcionamiento y gestión del parque automotor.
- **Flujo de Información:** El flujo de información dentro del sistema sigue un proceso definido: los usuarios finales interactúan con la API Gateway, que a su vez envía las solicitudes al servidor. Este último redirige las peticiones a las bases de datos correspondientes, recopila los datos y los devuelve a través de la API Gateway hacia los usuarios finales.
- **Funcionalidades del Sistema:** El sistema ofrece varias funcionalidades que incluyen la gestión de usuarios, vehículos, reservas, mantenimientos, repuestos y

sugerencias. Estas capacidades permiten un control completo y eficiente del parque automotor.

Entre los beneficios que ofrece el sistema se encuentran la mejora en la optimización de la comunicación y la mejora en la toma de decisiones gracias a la disponibilidad de información.

6.2.1.1 Open Java Development Kit

El proyecto ha sido desarrollado utilizando el Open Java Development Kit (Open JDK) en su versión 21, ofrece una plataforma integral y actualizada para el desarrollo de aplicaciones, incluyendo herramientas, bibliotecas y recursos necesarios para pruebas del sistema en el contexto específico del parque automotor del GAD Azogues. Incluye el compilado Java, la máquina virtual Java (JVM) y una biblioteca estándar de clases Java.

A continuación, se describen algunas de las características principales de OpenJDK:

- **Compilación AOT:** permite traducir el código Java a código nativo antes de la ejecución lo que puede mejorar significativamente el rendimiento del software.
- **API de accesos a archivos de bajo nivel:** esta API permite un acceso más directo al sistema de archivos permitiendo la mejora del rendimiento de aplicaciones que trabajan con archivos grandes.

Open JDK es una alternativa gratuita y de código abierto al Java Development Kit (JDK) de Oracle, lo que ha beneficiado a nuestro proyecto puesto que es accesible y está disponible para su uso libre sin restricciones de licencia, lo que ha permitido trabajar de manera más flexible y colaborativa sin preocupaciones de costos adicionales o restricciones legales, además de ser una opción popular en la comunidad de desarrollo de software dando un amplio soporte y una gran cantidad de recursos disponibles.

6.2.1.2 Maven

Además, se implementó la versión 4.0.0 de Maven, esta herramienta es esencial en el proceso de construcción y gestión de dependencias. Esta versión de Maven presenta mejoras significativas garantizando una mayor eficiencia en la compilación y administración de bibliotecas, el uso de la estructura de proyecto declarativa de Maven facilita la configuración y organización de código contribuyendo a un desarrollo más ordenado y mantenible.

Ahora, se describen características fundamentales de Maven:

- **Gestión de proyectos:** Maven proporciona una estructura uniforme y convenciones predefinidas, las cuales facilitan la organización y gestión de proyectos de software, incluyendo la definición de estructura de directorios, la configuración de archivos de propiedades y la gestión de dependencias.
- **Gestión de dependencias:** Maven permite simplificar la gestión de dependencias declarándolas en un archivo de configuración centralizado (pom.xml), además se encarga de descargar automáticamente las dependencias necesarias desde repositorios remotos permitiendo manejar las versiones de estas garantizando la consistencia y compatibilidad del proyecto.
- **Plug-ins:** Reutilización de plug-ins en Maven permite extender su funcionalidad y realizar diversas tareas durante el ciclo de vida del proyecto, como compilar el código, ejecutar pruebas, generar documentación, empaquetar aplicaciones, entre otras funcionalidades.
- **Estándares y buenas prácticas:** Maven fomenta el uso de estándares y buenas prácticas en el desarrollo de software, es decir, promueve la adopción de estándares de

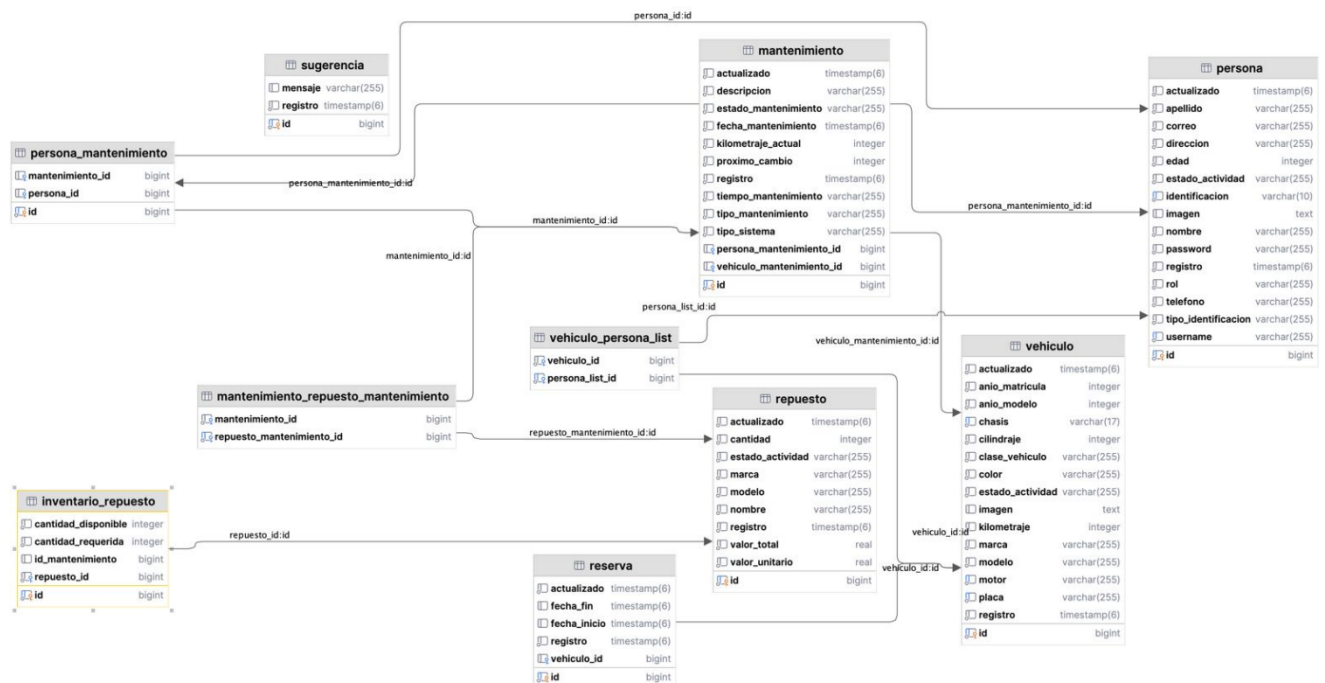
codificación, la generación automática de documentación, la ejecución de pruebas unitarias y la gestión adecuada de dependencias.

En este sentido la utilización de Maven en este proyecto simplifica la construcción, despliegue y gestión de este, gracias a su enfoque basado en convenciones y estándares mejorando la productividad y calidad del software desarrollado

6.2.2 PostgreSQL

Dentro del proyecto, se ha optado por utilizar la versión 16 de PostgreSQL como el principal sistema de gestión de base de datos (SGBD). Este SGBD, de código abierto, es muy común en aplicaciones empresariales y proyectos de desarrollo de software. A continuación, se presenta en la figura 1 la estructura detallada de las tablas de datos creadas y administradas por el BackEnd.

Figura 2 Estructura de tablas de datos



El sistema de gestión del parque automotor está diseñado para administrar de manera eficiente todos los aspectos relacionados con los vehículos y su mantenimiento. Se compone de varias entidades clave que representan diferentes elementos del sistema:

- **Persona:** Representa a los usuarios del sistema, que pueden ser conductores, mecánicos o personal administrativo. Cada persona está identificada por atributos como nombre, apellido, identificación, dirección, teléfono, correo electrónico y rol en el sistema.
- **Vehículo:** Representa los vehículos que se gestionan en el sistema, con atributos como placa, marca, modelo, año, kilometraje, estado, tipo de combustible y clase.
- **Mantenimiento:** Registra los mantenimientos realizados a los vehículos, incluyendo detalles como fecha, tipo de mantenimiento, descripción, estado, costo y la persona que lo realizó.
- **Repuesto:** Representa los repuestos utilizados en los mantenimientos, con atributos como nombre, descripción, cantidad disponible, cantidad mínima, precio unitario, marca y modelo.
- **Reserva:** Registra las reservas de vehículos realizadas por las personas, con detalles como fecha de inicio, fecha fin, estado, la persona que la realizó y el vehículo reservado.
- **Sugerencia:** Esta entidad permite a los usuarios enviar sugerencias sobre el sistema.

Relaciones y Funcionalidades

Las relaciones entre estas entidades permiten un flujo coherente de información y acciones dentro del sistema. Por ejemplo, las personas pueden realizar mantenimientos a los vehículos, reservar vehículos y enviar sugerencias sobre el sistema. Los mantenimientos se realizan en vehículos y pueden utilizar varios repuestos. Además, las reservas se realizan por una persona para un vehículo específico.

Las funcionalidades del sistema incluyen la gestión completa de usuarios, vehículos, mantenimientos, repuestos, reservas y sugerencias, permitiendo una administración integral del parque automotor, desde la creación y modificación de usuarios hasta el registro y seguimiento de mantenimientos y reservas.

6.2.3 FrontEnd

El frontend posee un rol fundamental en la construcción de una experiencia de usuario eficiente y satisfactoria, debido que constituye la interfaz del software que interactúa directamente con el usuario. Algunos elementos esenciales del frontend y su implementación abarcan:

- **Interfaz de usuario intuitiva:** El frontend tiene la responsabilidad de crear y mostrar una interfaz de usuario que sea intuitiva y accesible, facilitando a los usuarios una interacción eficaz con el sistema mediante la elaboración de pantallas, la disposición de elementos, la navegación y la retroalimentación visual que guíen al usuario a través de las funcionalidades del sistema de manera clara y coherente.
- **Experiencia de usuario personalizada:** El desarrollo del frontend posibilita la adaptación de la experiencia del usuario conforme a las necesidades y preferencias de los distintos usuarios del sistema. Esto incluye la capacidad de personalizar las vistas y

ajustar la interfaz para diversos dispositivos y dimensiones de pantalla, asegurando una experiencia óptima en cualquier situación de uso.

- **Integración con el backend:** La comunicación con el backend del sistema se realiza a través de APIs para acceder y manipular datos, esta integración garantiza que la información presentada este actualizada y sea coherente con los datos almacenados en el backend.
- **Tecnologías de desarrollo web:** El empleo de tecnologías web como HTML (Lenguaje de Marcado de Hipertexto), CSS (Hojas de Estilo en Cascada) y TypeScript, en conjunto con el marco de trabajo y las bibliotecas de Angular, posibilitaron el desarrollo de interfaces dinámicas, adaptables e interactivas que satisfacen tanto los requisitos del proyecto como las preferencias de los usuarios.

6.2.3.1 Angular

El proyecto ha sido desarrollado con angular 15 el cual ha permitido aprovechar las últimas características y así mejorando la plataforma de desarrollo de aplicaciones web para brindar un entorno de desarrollo sólido y adaptable, simplificando la creación de interfaces de usuario dinámicas e interactivas a través de su enfoque centrado en componentes., además de posibilitar una organización eficiente de código promoviendo la reutilización y la escalabilidad del sistema.

6.2.3.2 Node.js

El proyecto se ha elaborado utilizando Node.js en su edición 18, lo que ha posibilitado la utilización de las funcionalidades avanzadas de esta plataforma de ejecución, generando un

entorno altamente eficiente. Esto ha simplificado la creación de un frontend escalable y de alto rendimiento, aprovechando la diversidad de bibliotecas y módulos disponibles.

A continuación, se detallan características principales de node.js:

- **Eficiencia en el rendimiento:** ofrece un ambiente de ejecución de alto rendimiento que permite manejar un gran número de conexiones de forma eficiente gracias a su modelo de E/S no bloqueante.
- **Escalabilidad:** posee una arquitectura escalable que permite escalar horizontalmente con facilidad mediante la adición de instancias de servidores, haciendo adecuado su uso en sistemas con grandes volúmenes de tráfico.
- **Facilidad de uso:** es conocido por su facilidad de uso y su bajo tiempo de aprendizaje, su sintaxis en TypeScript permite a desarrolladores crear aplicaciones web de forma rápida y eficaz.
- **Amplio ecosistema de paquetes:** cuenta con un extenso repositorio de paquetes Node Package Manager (NPM) que ofrece una amplia gama de módulos y bibliotecas para una variedad de funcionalidades, facilitando el desarrollo de aplicaciones complejas.

6.3 Implementación del BackEnd

La ejecución del backend implica la conversión del diseño y la estructura empresarial de una aplicación en código funcional, el cual gestiona todas las operaciones del sistema, desde el manejo de las solicitudes del usuario hasta el acceso y manipulación de la base de datos, así como la implementación de medidas de seguridad, entre otras funciones.

Se ha considerado como punto de partida el Modelo-Vista-Controlador (MVC), un patrón de diseño arquitectónico ampliamente empleado en aplicaciones web y otras plataformas con interfaces de usuario.

A continuación, se detalla la implementación realizada:

- **Entidades:** en este paquete se encuentran las clases que representan las entidades del sistema, por ejemplo, la clase usuario, vehículo, etc. Cada una de las clases contiene los atributos que representan las propiedades de la entidad, como nombre, apellido, teléfono, dirección, etc. Estas clases tienen anotaciones específicas del framework utilizado en este caso Java con SpringBoot, como `@Entity`, `@Table`, `@Data`, etc, como se observa en la figura 3.

Figura 3 Entidad

```
@Entity
@Data
public class Persona implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @JsonProperty
    private Long id;

    @JsonProperty
    @Column(nullable = false)
    private String nombre;

    @JsonProperty
    @Column(nullable = false)
    private String apellido;
```

- **Repositorios:** en este paquete se encuentran las interfaces que definen las operaciones de acceso a los datos de cada entidad. En la figura 4 se muestra la configuración presente en cada una de las interfaces.

Figura 4 Repositorio

```
@Repository
@Transactional
public interface InventarioRepuestoRepository extends JpaRepository<InventarioRepuesto, Long> {

    2 usages new *
    @Query("SELECT ir FROM InventarioRepuesto ir WHERE ir.repuesto.id = :repuestoId AND ir.idMantenimiento = :idMantenimiento")
    Optional<InventarioRepuesto> buscarRepuestoIdYMantenimientoId(@Param("repuestoId") Long repuestoId, @Param("idMantenimiento") Long idMantenimiento);
}
```

Al igual que en las entidades las interfaces utilizan, anotaciones como `@Repository`, `@Transactional`, `@Query`, etc., para manipular las operaciones de consultas específicas del lenguaje de consulta de la base de datos empleando métodos proporcionados por el framework Object-Relational Mapping (ORM).

- **Controladores:** en este paquete se encuentran las clases que manejan las solicitudes HTTP entrantes y definen la lógica de negocio para el sistema. Por ejemplo, en la figura 4 se muestra un controlador de sugerencias el cual integra las anotaciones `@CrossOrigin`, `@RestController`, `@RequestMapping` para la comunicación con el frontend y a su vez se definen métodos para el registro y la obtención de todas las sugerencias guardadas en la base de datos.

Estos métodos están anotados con anotaciones específicas del framework que indican el tipo de solicitud HTTP, por ejemplo, `@GetMapping`, `@PostMapping`, `@PutMapping`, `@DeleteMapping`, etc.

Figura 5 Controlador

```
@CrossOrigin("*")
@RestController
@RequestMapping("/api/sugerencia")
public class SugerenciaControlador {
    2 usages
    private static final Logger log = LoggerFactory.getLogger(PersonaControlador.class);

    3 usages
    private final SugerenciaRepositorio sugerenciaRepositorio;

    nando8082
    public SugerenciaControlador(@Autowired SugerenciaRepositorio sugerenciaRepositorio) {...}

    nando8082
    @GetMapping("/listarSugerencias")
    public ResponseEntity<?> obtenerSugerencias(){...}

    nando8082
    @PostMapping("/crearSugerencia")
    public ResponseEntity<?> crearSugerencia(@RequestBody CrearSugerencia crearSugerencia){...}
}
```

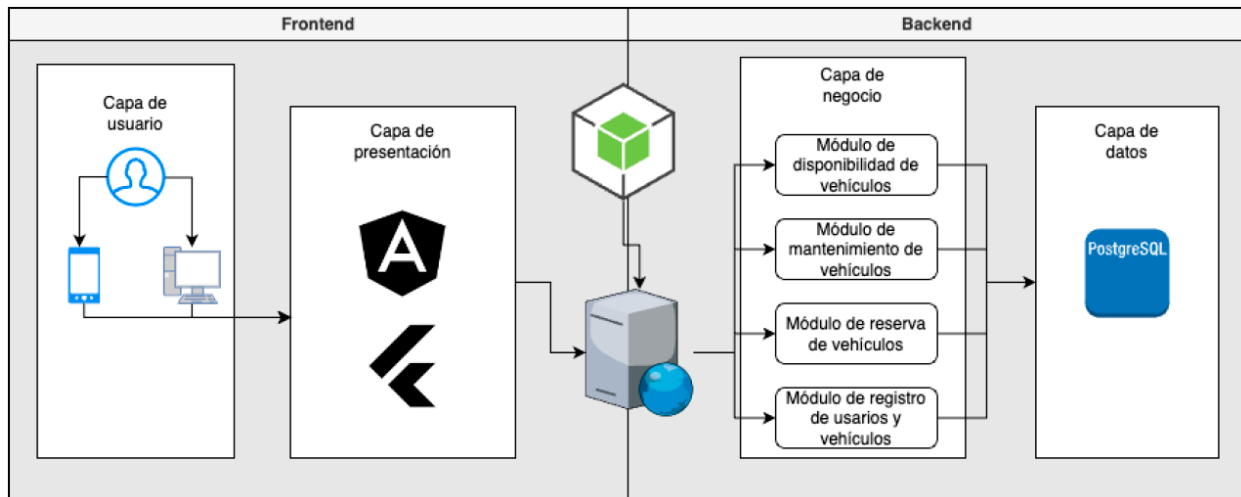
En conjunto estos paquetes forman la estructura básica del backend donde las entidades representan los objetos principales del sistema, los repositorios manejan el acceso a la información de la base de datos y los controladores manejan las solicitudes HTTP y la lógica de negocio asociada a esas solicitudes facilitando la organización y mantenimiento del código y permitiendo una clara separación de responsabilidades en el proyecto.

6.4 Diagrama lógico

En esta sección, se inicia con la presentación del diagrama lógico en el cual se representan los diferentes componentes del sistema desarrollado y sus interacciones lógicas. Este diagrama muestra una visión de la arquitectura del sistema, mostrando cómo cada módulo o elemento del

sistema se conecta entre sí para lograr funciones específicas y así conocer mejor la organización interna del sistema para facilitar la identificación de fallos y áreas de mejora.

Figura 6 Diagrama Lógico



A continuación, se explora en detalle las capacidades del sistema desde la gestión de vehículos y empleados hasta el desarrollo de un sitio web robusto y accesible.

6.5 Funciones del Sistema

Aquí, se detallan las capacidades del sistema desarrollado para administrar y conservar el parque automotor del GAD Azogues. Se expondrá su funcionamiento y las habilidades que posee para cumplir con estas responsabilidades.

6.5.1 Gestión de Vehículos y Empleados

Se ha integrado el módulo Jasypt en el backend con el fin de asegurar la confidencialidad de las contraseñas de los usuarios. Este recurso codifica las contraseñas antes de ser almacenadas, ofreciendo una capa extra de seguridad contra potenciales amenazas. Mediante el uso de Jasypt,

las contraseñas de los usuarios son cifradas a través de un algoritmo robusto, lo que implica que, incluso en caso de una intrusión en la base de datos, las contraseñas se mantendrán protegidas y fuera del alcance de personas no autorizadas.

En la ilustración 1, se muestra cómo se puede utilizar la clase `StandardPBEStrEncryptor` de la biblioteca `Jasypt` para cifrar contraseñas en Java. El cifrado es una forma importante de proteger la información confidencial, como contraseñas, claves secretas y datos financieros.

Figura 7 Clave Secreta (Jasypt)

```
@Value("${jasypt.encryptor.password}")  
private String claveSecreta;
```

La anotación `@Value` permite acceder a valores definidos en archivos de configuración externos, por ejemplo, archivos `.properties` o `.yml`, y utilizarlos en el código de la aplicación. En este caso, la propiedad `jasypt.encryptor.password` contiene la clave secreta necesaria para cifrar y descifrar las contraseñas de los usuarios.

Al emplear este método, se asegura que la clave secreta se configure de manera externa y no se incorpore directamente en el código fuente, lo que aumenta la seguridad y la adaptabilidad del sistema. Así, cualquier cambio en la contraseña se puede realizar fácilmente ajustando el archivo de configuración correspondiente, sin necesidad de alterar el código de la aplicación.

Además, se incorporó la dependencia `Lombok` al proyecto, lo cual simplificó considerablemente la generación de métodos `getters` y `setters` en las clases de Java. Esta integración contribuyó a la disminución de la extensión del código y a su mejor legibilidad y mantenibilidad. `Lombok` automatiza la creación de estos métodos, eliminando la necesidad de escribirlos

manualmente en cada clase. Esto no solo reduce la cantidad de código a desarrollar, sino que también facilita la reutilización de las clases en diferentes entornos.

A continuación, en la ilustración 2, se muestra cómo se puede utilizar la anotación `@Data` de Lombok para simplificar una clase. La anotación `@Data` genera automáticamente getters, setters, equals, hashCode y toString, reduciendo el código necesario para las clases.

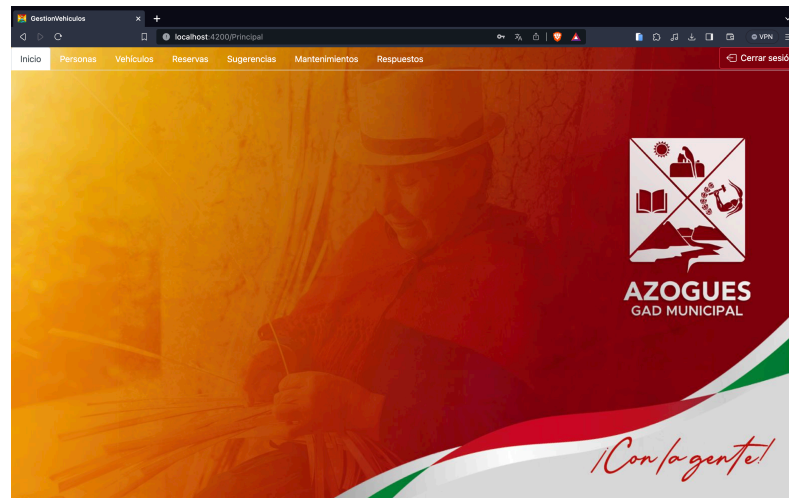
Figura 8 Simplificado con Lombok

```
@Data  
public class Persona implements Serializable {
```

6.5.2 Desarrollo del sitio Web

La creación de aplicaciones modernas y accesibles a través de Internet depende en gran medida del desarrollo web, que es un elemento fundamental. En esta sección, examinaremos los diferentes aspectos del desarrollo web que afectaron la creación de la aplicación Angular para el Gobierno Autónomo Descentralizado (GAD) Municipal. Cada aspecto del desarrollo web, desde la implementación de funciones hasta otros detalles, juega un papel fundamental en garantizar una experiencia de usuario positiva y fluida.

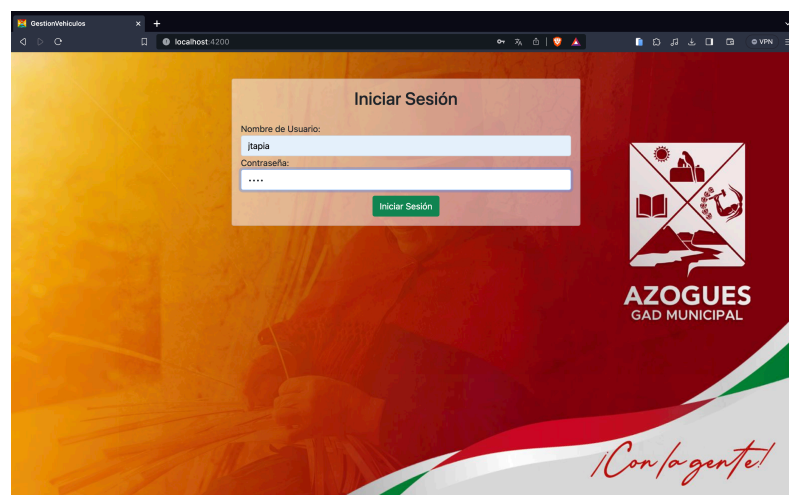
Figura 9 Pantalla de Inicio



6.5.2.1 Inicio de sesión

Este diseño ofrece una interfaz que permite a los usuarios ingresar a una aplicación web al introducir sus datos en los campos designados. Una vez ingresada esta información, deben clic en un botón para iniciar sesión y usar la aplicación. La apariencia visual del diseño es contemporánea y bien organizada, lo que mejora la experiencia de inicio de sesión, haciéndola más agradable y comprensible para los usuarios.

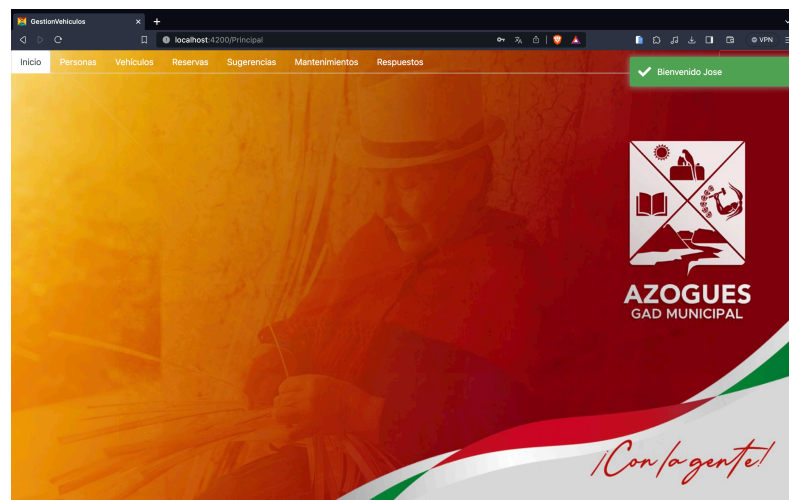
Figura 10 Pantalla de Inicio de Sesión



6.5.2.2 Pantalla de bienvenida

Este presenta la imagen completa que utiliza el GAD Azogues como la parte de navegación de la aplicación web, donde los usuarios pueden moverse entre secciones o páginas. Dependiendo del rol del usuario (como administrador, mecánico o chofer), se muestran diferentes opciones de navegación en la barra superior. Un administrador puede acceder a todas las secciones, mientras que un mecánico o chófer tiene acceso a un conjunto limitado de opciones. Además, hay un botón para cerrar sesión que permite al usuario salir de su sesión actual.

Figura 11 Pantalla de bienvenida

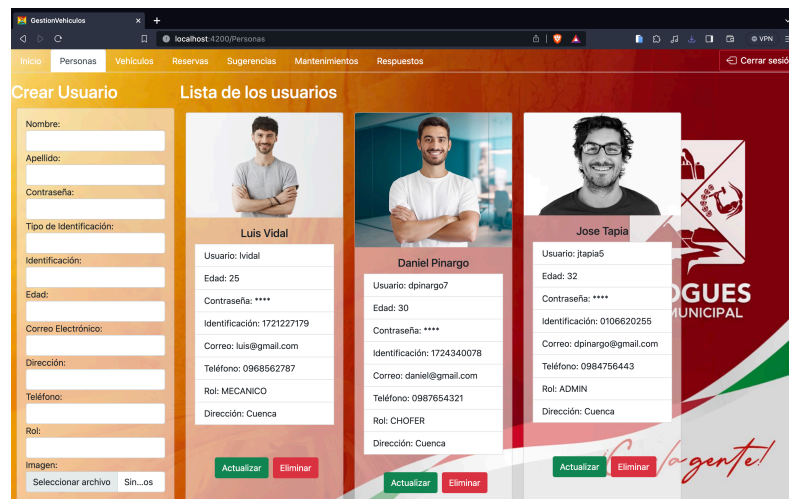


6.5.2.3 Personas

Este fragmento describe una interfaz de administración de usuarios en una aplicación web. Esta página está diseñada para que un administrador pueda añadir, visualizar, modificar y eliminar usuarios del sistema. La sección principal de la página presenta dos paneles: uno para la creación de nuevos usuarios y otro para listar los usuarios ya existentes. El panel de creación de usuarios contiene formularios para ingresar la información del nuevo usuario, como nombre, apellido,

contraseña, tipo de identificación, entre otros detalles. Además, ofrece la posibilidad de cargar una imagen de perfil. Por otro lado, el panel de listado de usuarios muestra una lista de los usuarios actuales, con opciones para editar, actualizar y eliminar cada uno. La información de cada usuario se despliega en tarjetas individuales, y se brinda la capacidad de editar estos datos directamente en línea.

Figura 12 Pantalla de las Personas

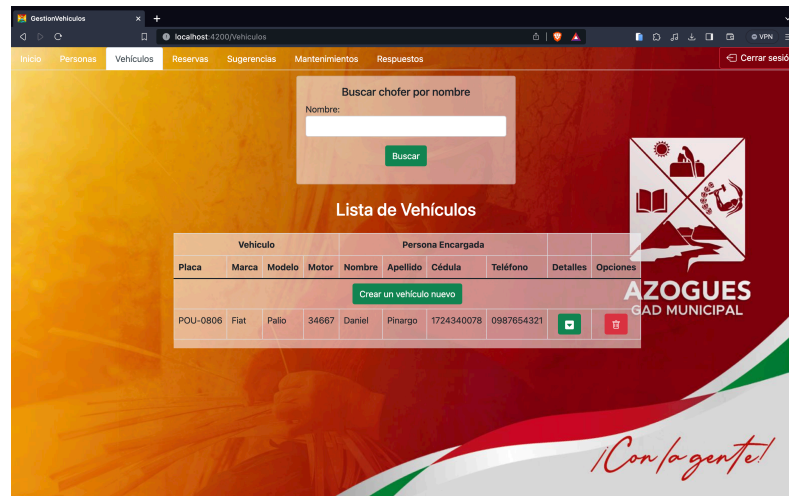


6.5.2.4 Vehículos

Se gestiona vehículos y personas asociadas a ellos, diseñada para ser utilizada por administradores, conductores y mecánicos. Justo debajo de la barra de navegación, hay un formulario que permite buscar conductores por nombre. Al ingresar el nombre y hacer clic en "Buscar", se mostrarán los conductores que coincidan con el nombre proporcionado. Después del formulario de búsqueda, se presenta una lista de vehículos. Cada vehículo tiene sus respectivos detalles. Los administradores pueden crear nuevos vehículos haciendo clic en "Crear un vehículo nuevo" y completando el formulario correspondiente. Para cada vehículo, se pueden ver los

detalles y las personas asociadas a él haciendo clic en el botón de detalles. Desde aquí, los usuarios pueden eliminar personas asociadas a ellos si es necesario.

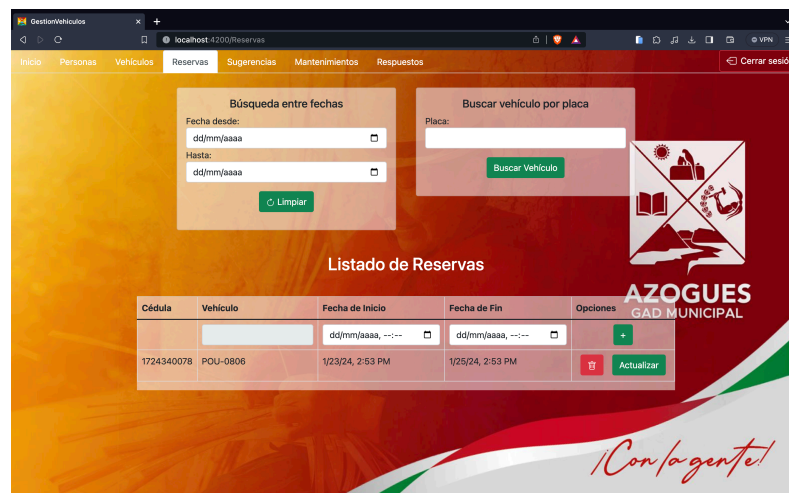
Figura 13 Pantalla de Vehículos



6.5.2.5 Reservas

Se representa una interfaz de usuario para gestionar reservas de los vehículos. Permite buscar vehículos por placa, filtrar reservas por fechas y realizar operaciones como crear, actualizar y eliminar reservas. La interfaz es intuitiva, con secciones específicas para cada funcionalidad y elementos de entrada que facilitan la interacción del usuario.

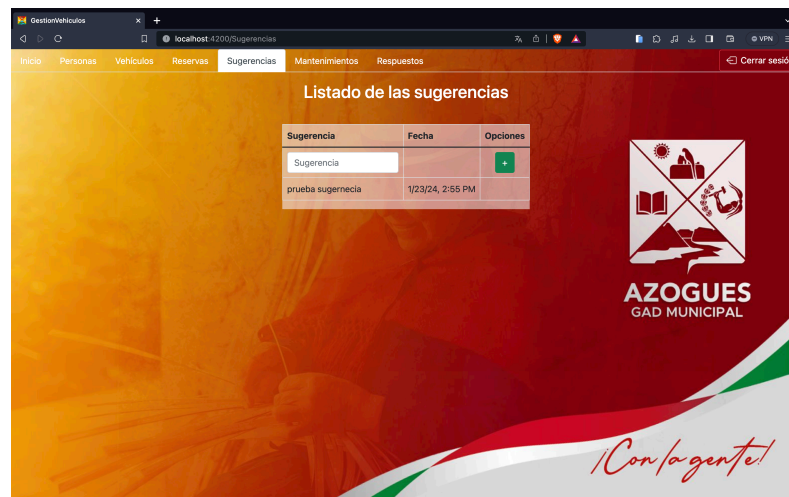
Figura 14 Pantalla de Reservas



6.5.2.6 Sugerencias

Se representa la interfaz para que los usuarios con roles de administrador pueden enviar y ver sugerencias. La página incluye un formulario para ingresar nuevas sugerencias y una lista para mostrar las sugerencias anteriores. La interfaz muestra un formulario donde los usuarios pueden ingresar sus sugerencias y un botón para enviarlas, junto con una lista de todas las sugerencias anteriores. Además, se comprueba si el usuario cuenta con privilegios de administrador antes de conceder acceso a esta página.

Figura 15 Pantalla de Sugerencias

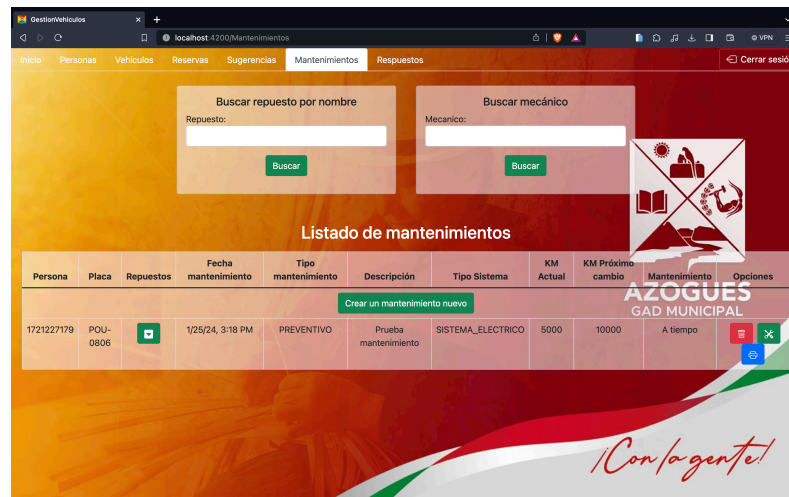


6.5.2.7 Mantenimientos

Se gestionan mantenimientos de vehículos. Permite a los usuarios buscar repuestos por nombre y mecánicos por nombre, además de crear y listar mantenimientos. La interfaz muestra dos secciones principales: una para buscar repuestos y otra para buscar mecánicos. Una vez encontrados, se pueden agregar al formulario de creación de mantenimientos. También hay una tabla que lista todos los mantenimientos existentes, con opciones para finalizarlos, agregar repuestos y mostrar detalles. El formulario de creación de mantenimientos permite ingresar datos

como la placa del vehículo, la fecha del mantenimiento, el tipo de mantenimiento, la descripción, el tipo de sistema y el kilometraje actual. Una vez creados, los mantenimientos se pueden listar y gestionar desde la tabla principal.

Figura 16 Pantalla de Mantenimientos



6.5.2.8 Repuestos

Se permite la gestión de una lista de repuestos. Diseñada para usuarios con roles específicos como administradores o mecánicos, la página proporciona las capacidades de enumerar, generar, modificar y eliminar piezas de repuesto. La pantalla de usuario presenta un catálogo de piezas de repuesto, incluyendo información detallada como el nombre, la cantidad, el precio unitario, la marca y el modelo, y habilita a los usuarios para buscar piezas por su nombre. Además de gestionar la autenticación de usuarios, el código ofrece mensajes de notificación para informar sobre el resultado de las acciones realizadas, como la creación o eliminación de repuestos, brindando una experiencia intuitiva y funcional para la administración de repuestos.

Figura 17 Pantalla de Repuestos



Para acceder a información más exhaustiva sobre el funcionamiento de la aplicación y consultar el manual del usuario, consultar el Anexo 16.2. Además, para acceder y examinar a detalle el código completo de la aplicación y explorar su implementación, visitar el Anexo 16.3, donde se encuentra el enlace del repositorio de GitHub.

7. Cronograma

Total de horas: 600

Horas completadas Pinargo Oña Daniel Fernando (DP): 300

Horas completadas Tapia Vidal José Luis (JT): 300

Erwin Jairo Sacoto Cabrera (ES)

Fecha inicio: lunes 30/10/2023

Fecha fin: viernes 07/01/2024

Tabla 6 Cronograma

Objetivo	Actividad	Responsable	Horas	Fecha Inicio	Fecha Fin
Sprint 1					
OE1	ACT.1	DP - JT	50	30/10/2023	05/11/2023
OE2	ACT.1	DP - JT	55	06/11/2023	12/11/2023
Sprint 2					
OE2	ACT.2	DP - JT	55	13/11/2023	18/11/2023
OE3	ACT.1	DP - JT	55	19/11/2023	26/11/2023
Sprint 3					
OE3	ACT.2	DP - JT	55	27/11/2023	03/12/2023
OE4	ACT.1	DP - JT	55	04/12/2023	10/12/2023
Sprint 4					
OE4	ACT.2	DP - JT	55	11/12/2023	17/12/2023
OE4	ACT.3	DP - JT	55	18/12/2023	24/12/2023
Sprint 5					
OE5	ACT.1	DP - JT	55	25/12/2023	31/12/2023

OE6	ACT.1	DP - JT	55	25/12/2023	31/12/2023
OE7	ACT.1	DP - JT	55	01/01/2024	07/01/2023

8. Presupuesto

Tabla 7 Presupuesto

DENOMINACIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
	UNIDADES	DÓLARES	DÓLARES
1. Tecnológico			
MacBook Pro	2	2000.00	4000.00
Celular	2	200.00	400.00
Licencia IntelliJ Idea	2	779.00	1558.00
2. Servicios			
Servicios de Internet	2	25.00	50.00
3. Otros			
Imprevistos	2	100.00	200.00
TOTAL:			6208.00

9. Conclusiones

El dominio de Java, Angular y SQL ha sido importante para el desarrollo del proyecto. Java permitió desarrollar un backend robusto y eficiente, Angular posibilitó un frontend dinámico e interactivo, y SQL facilitó la gestión eficaz de la base de datos. Gracias a estas tecnologías, se superaron los desafíos técnicos del proyecto y se alcanzaron los objetivos planteados.

Además, la implementación de tecnologías modernas garantiza la confiabilidad y seguridad del sistema, así como su capacidad para manejar grandes volúmenes de datos de manera eficiente y escalable. La adopción de prácticas de desarrollo estándar como el patrón de diseño MVC no solo asegura la modularidad y mantenibilidad del código, sino que también permite una clara separación de responsabilidades entre los diferentes componentes del sistema facilitando su comprensión y futuras actualizaciones.

Gracias al dominio de estas tecnologías, se logró diseñar un sistema adaptado a las necesidades específicas de la gestión de datos de vehículos y las reservas de vehículos, lo que contribuye directamente al cumplimiento de los objetivos 2 y 3 planteados para este proyecto de tesis.

Es importante destacar, la integración de prácticas de seguridad como Jasyp y la mejora de la eficiencia del desarrollo con Lombok fortalece la solidez y confiabilidad del sistema, garantizando la protección de datos sensibles y reduciendo la complejidad del código base.

Las pruebas unitarias realizadas han demostrado la funcionalidad y confiabilidad del sistema en cada una de sus etapas. Los resultados obtenidos cumplen con las expectativas en términos de calidad además de ser intuitivo y amigable con el usuario final.

Sin embargo, se debe reconocer que siempre existirán áreas de mejora y oportunidades de crecimiento del sistema, tomando en consideración la incorporación de nuevos módulos para futuras actualizaciones.

10. Recomendaciones

A continuación, se ofrecen algunas sugerencias fundamentales para el informe definitivo sobre el sistema de administración y mantenimiento del parque vehicular del GAD Azogues:

- **Capacitación en Tecnologías Relevantes:** Asegurarse de que el equipo esté debidamente capacitado en las tecnologías clave para el desarrollo del sistema, como Java, Angular y SQL. Esto garantizará un desarrollo eficiente y efectivo.
- **Priorización de la Experiencia del Usuario (UX):** Proporcionar una atención específica al diseño de la interfaz de usuario del sistema de gestión y mantenimiento de vehículos con el objetivo de garantizar una experiencia amigable e intuitiva para los usuarios. Este enfoque ayudará a mejorar la aceptación del sistema entre los usuarios finales.
- **Seguridad de Datos:** Implementar prácticas de seguridad sólidas, especialmente al tratar con datos confidenciales, como la información de los vehículos y los empleados del GAD.
- **Planificación y Seguimiento del Proyecto:** Mantener un estricto control sobre el cronograma del proyecto de desarrollo del sistema y realiza un seguimiento regular del progreso. Cualquier desviación significativa debe ser abordada de inmediato para evitar retrasos innecesarios.
- **Pruebas Rigurosas:** Reservar un periodo adecuado para la planificación y realización exhaustiva de pruebas funcionales, asegurando así la calidad del sistema de gestión y mantenimiento de vehículos.
- **Documentación Clara y Completa:** Asegurarse de mantener una documentación detallada y exhaustiva para el sistema, abarcando tanto el código fuente como el manual de usuario del sistema de gestión y mantenimiento vehicular. Esto simplificará las actualizaciones futuras y la transmisión de conocimientos.

- **Considerar Integración Continua:** Evaluar la posibilidad de implementar prácticas de integración continua para la localización temprana de errores y garantizar una entrega continua de nuevas funcionalidades.
- **Gestión de Imprevistos:** Reservar un espacio en el presupuesto y en la planificación del proyecto de desarrollo del sistema para posibles contratiempos. Esta medida facilitará la gestión de cualquier obstáculo imprevisto.

11. Referencias bibliográficas

Angular. (2023). *Angular*. Obtenido de Angular: <https://angular.io>

Aranda, J., Sacoto-Cabrera, E., Haro-Mendoza, D., & Astudillo-Salinas, F. (2021). Redes 5G: una revisión desde las perspectivas de arquitectura, modelos de negocio, ciberseguridad y desarrollos de investigación. *Revista Digital Novasinergia*, 6-41.

Azaustre, C. (Enero de 2015). *Desarrollo Web ágil con AngularJS: Aprendes buenas prácticas y desarrollo ágil con el framework de JavaScript AngularJS*. Obtenido de Google Libros: https://books.google.es/books?hl=es&lr=&id=OXpDBAAQBAJ&oi=fnd&pg=PT2&dq=angular+aplicacion+web&ots=LzWwR-_Ubd&sig=F6tB0kmWhv9iNE5zapzMaJqEd-Q#v=onepage&q=angular%20aplicacion%20web&f=false

Domínguez, J. (2019). *FUNDAMENTOS DE POSTGRESQL*. Obtenido de Creative Commons: https://d1wqtxts1xzle7.cloudfront.net/61304277/libro_postgresql20191122-55847-1dti2nd-libre.pdf?1574462508=&response-content-disposition=inline%3B+filename%3DFUNDAMENTOS_DE_POSTGRESQL.pdf&Expires=1706591135&Signature=TDH7phz7~lnI7VWO7Lhtswfj0WfzxB2hKxJjeF

Estayno, M., & Meles, J. (noviembre de 2014). El Rol del Product Owner en la definición y validación de las user stories. *El Rol del Product Owner en la definición y validación de las user stories*. Buenos Aires, Buenos Aires, Argentina. Obtenido de <http://dspace.palermo.edu/dspace/handle/10226/1317>

Estrada Velasco, M. V., Nuñez Villacis, J. A., Saltos Chávez, P. R., & Cunuhay Cuchiye, W. C. (9 de diciembre de 2021). Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software. *Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software*, 7(4), 54. Riobamba, Cotopaxi, Ecuador. doi:10.23857/dc.v7i4.2429

- Ginestà, M., & Pérez, O. (2009). *Bases de datos en PostgreSQL*. Obtenido de <https://artpatcusco.com/sis/pdf/20140527172742en.pdf>
- IntelliJ IDEA. (2023). *Jet Brains*. Obtenido de Jet Brains: <https://www.jetbrains.com/idea/>
- Loja, M., Molina, J., Loja, F., & Cañarte, E. (Agosto de 2020). *UTMACH*. Obtenido de <http://repositorio.utmachala.edu.ec/bitstream/48000/17919/1/436-Texto%20del%20art%C3%ADculo-787-1-10-20200824.pdf>
- Medrano, A., & Ali, L. (2022). *Análisis comparativo de tecnologías Front End Angular Js Vs React Js, en el modelo de procesos para el desarrollo de aplicaciones web*. Obtenido de DSpace de la Universidad Técnica de Babahoyo: <http://dspace.utb.edu.ec/handle/49000/11386>
- PostgreSQL. (2023). *PostgreSQL*. Obtenido de PostgreSQL: <https://www.postgresql.org/>
- Ramírez, M., Soto, M., Moreno, H. B., Rojas, E., & Milán, N. (Enero de 2019). Metodología SCRUM y desarrollo de Repositorio Digital. *Metodología SCRUM y desarrollo de Repositorio Digital*, 1062-1072. Baja California, Tijuana, México: Associação Ibérica de Sistemas e Tecnologias de Informacao. Obtenido de <https://www.proquest.com/openview/7635ce5360bdb82d0c42c815e17f8323/1?pq-origsite=gscholar&cbl=1006393>
- Rodriguez Alvarado, D., & Sacoto-Cabrera, E. (2021). Implementation and Analysis of the Results of the Application of the Methodology for Hybrid Multi-cloud Replication Systems. *In The International Conference on Advances in Emerging Trends and Technologies* (págs. 273-286). Springer International Publishing.

- Sacoto Cabrera, E., & Perez Torres, A. (2023). Digital Transformation: a Review of Enabling Technologies, Maturity Models, and Open Research Issues. *2023 IEEE Seventh Ecuador Technical Chapters Meeting (ECTM)*, (págs. 1-6). Ambato.
- Sacoto Cabrera, E., Guijarro, L., & Maillé, P. (2022). Game theoretical analysis of a multi-MNO MVNO business model in 5G networks. *Electronics*, 933.
- Sacoto-Cabrera, E., Castillo, I., Pauta, W., Trelles, P., & Guambaña, L. (2021). Smart-Water: Digital Transformation of Urban Water Measurement. *2022 IEEE ANDESCON* (págs. 1-6). Barranquilla: IEEE.
- Sacoto-Cabrera, E., León-Paredes, G., & Verdugo-Romero, W. (2021). Lorawan: Application of nonlinear optimization to base stations location. *. n Communication, Smart Technologies and Innovation for Society: Proceedings of CITIS 2021* (págs. 15-524). Guayaquil: Springer Singapore.
- Vazquez, Y., Mier, L., & Sotolongo, A. (5 de Mayo de 2016). *SciELO*. Obtenido de http://scielo.sld.cu/scielo.php?pid=S2227-18992016000600006&script=sci_arttext&tlng=en
- Vladislav, T., & Timofey, B. (2018). *CEUR Workshop Proceedings*. Obtenido de https://ceur-ws.org/Vol-2135/SEIM_2018_paper_32.pdf
- Wilken, J. (7 de Marzo de 2018). *Angular in action*. Obtenido de Google Libros: <https://books.google.es/books?hl=es&lr=&id=XTgzEAAAQBAJ&oi=fnd&pg=PT11&dq=angular+application+web&ots=g7By1gk84h&sig=dxPqQuiI9U-jgwPbH98UAu3WqIE#v=onepage&q=angular%20application%20web&f=false>

12. Anexos

Anexo 1

Especificación de requisitos de software

Proyecto: DESARROLLO DE UN SISTEMA PARA LA
GESTIÓN Y EL MANTENIMIENTO DEL PARQUE
AUTOMOTOR DEL GAD MUNICIPAL DE AZOGUES

Revisión: 1

Ficha del documento

Fecha	Revisión	Autor
23/11/2023	24/11/2023	Daniel Pinargo
23/11/2023	24/11/2023	José Tapia

Documento validado por las partes en fecha: 18/12/2023

Por el cliente	Por el cliente
Ing. Cristian Cardenas	Ing. Andres Coronel

Introducción

La Especificación de Requisitos de Software (SRS) Para el desarrollo del Sistema de Gestión y Mantenimiento del Parque Automotor del GAD Azogues tiene como objetivo principal proporcionar una visión detallada de los requisitos y funcionalidades que se deben implementar en el sistema. Este sistema tiene dos componentes principales: la gestión de agendamiento de vehículos diarios y el plan de mantenimiento preventivo y correctivo para el área de mecánica del GAD.

Propósito

El documento pretende describir los requisitos funcionales y no funcionales del Sistema de Gestión y Mantenimiento del Parque Automotor del GAD Azogues, guía y de referencia para desarrolladores y demás partes involucradas en el diseño, implementación y evaluación del sistema.

Alcance

El producto se denominará "Sistema de Gestión y Mantenimiento del Parque Automotor del GAD Azogues". Este sistema comprenderá dos módulos principales:

- Módulo de Agendamiento de Vehículos Diarios
- Módulo de Plan de Mantenimiento Preventivo y Correctivo

Personal involucrado

Nombre	Erwin Jairo Sacoto Cabrera
Rol	Scrum Master
Responsabilidades	Facilitar y coordinar las actividades del equipo Scrum.

	<p>Asegurarse de que se sigan los procesos y prácticas ágiles.</p> <p>Eliminar impedimentos que afecten al equipo.</p> <p>Colaborar con el Product Owner para mantener actualizado el backlog.</p>
Información de contacto	desacoto@ups.edu.ec
Aprobación	

Nombre	Pinargo Oña Daniel Fernando
Rol	Developer
Responsabilidades	<p>Desarrollar y mantener el software según los requisitos</p> <p>Colaborar en la planificación y estimación de tareas.</p> <p>Participar en reuniones de revisión y retrospectiva.</p> <p>Colaborar con otros miembros del equipo para lograr los objetivos.</p>
Información de contacto	dpinargo@est.ups.edu.ec
Aprobación	

Nombre	Tapia Vidal José Luis
Rol	Developer

Responsabilidades	<p>Contribuir al desarrollo y mantenimiento del software.</p> <p>Participar en reuniones de planificación y revisión.</p> <p>Colaborar en la identificación y resolución de problemas.</p> <p>Mantenerse actualizado sobre las tecnologías relevantes.</p>
Información de contacto	jtapiav2@est.ups.edu.ec
Aprobación	

Nombre	Ing. Cristian Cardenas Rodriguez
Rol	Product Owner
Responsabilidades	<p>Contribuir con los requerimientos del sistema.</p> <p>Participar en reuniones de planificación y revisión.</p>
Información de contacto	cristian.cardenas@azogues.gob.ec
Aprobación	

Nombre	Ing. Andres Coronel Ortega
Rol	Product Owner
Responsabilidades	<p>Contribuir con los requerimientos del sistema.</p> <p>Participar en reuniones de planificación y revisión.</p>

Información de contacto	andres.coronel@azogues.gob.ec
Aprobación	

Definiciones, acrónimos y abreviaturas

- **GAD Azogues:** Gobierno Autónomo Descentralizado Municipal de Azogues
- **Vehículo:** Todo medio de transporte motorizado propiedad del GAD Azogues
- **Chofer:** Persona autorizada para conducir un vehículo del GAD Azogues
- **Administración:** Persona encargada de gestionar el sistema
- **Calendario:** Herramienta que permite visualizar la disponibilidad de los vehículos y los choferes
- **Filtro de búsqueda:** Herramienta que permite buscar información específica en el calendario
- **Sugerencia o reclamo:** Comentario o queja que se puede enviar al sistema
- **Impresión:** Proceso de generar una copia física de un documento electrónico
- **Modificación:** Proceso de cambiar la información de un documento
- **Almacenamiento:** Proceso de guardar información en un dispositivo
- **Múltiples oficinas:** Más de una oficina del GAD Azogues
- **Mantenimiento preventivo:** Mantenimiento realizado de forma periódica para prevenir fallas
- **Mantenimiento correctivo:** Mantenimiento realizado para corregir fallas que ya se han producido
- **ABC:** Análisis, diagnóstico y corrección

- **Subsistema:** Conjunto de componentes que forman parte de un sistema

Descripción general

Perspectiva del producto

El sistema es un producto independiente del sistema de gestión del parque automotor del GAD Azogues, diseñado para usarlo por los empleados, incluyendo conductores, administradores y otros usuarios autorizados.

Funcionalidad del producto

El sistema proporciona las siguientes funcionalidades principales:

- Programación de vehículos diarios: El sistema permite a los usuarios programar vehículos para el uso diario.
- Plan de mantenimiento preventivo y correctivo: El sistema permite a los usuarios gestionar el plan de mantenimiento preventivo y correctivo de los vehículos

Restricciones

El sistema debe cumplir con las siguientes restricciones:

- El sistema debe ser compatible con el hardware y el software existente del GAD Azogues.
- El sistema debe ser accesible desde múltiples oficinas del GAD Azogues.
- El sistema debe ser seguro y proteger la información de los usuarios.

Suposiciones y dependencias

El sistema se basa en las siguientes suposiciones y dependencias:

- Los usuarios tienen acceso a Internet.

Evolución previsible del sistema

En el futuro, el sistema se puede mejorar para incluir las siguientes características:

La posibilidad de generar informes más detallados sobre el uso de los vehículos.

La posibilidad de integrar el sistema con otros sistemas del GAD Azogues.

Requisitos específicos

Número de requisito	RF1
Nombre de requisito	Registro de Vehículos y Choferes
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir el registro de información detallada de los vehículos, incluyendo datos del vehículo y del chofer, junto con sus respectivas fotografías.

Número de requisito	RF2
Nombre de requisito	Autenticación de Usuarios
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe contar con un mecanismo de autenticación mediante usuario y contraseña para el acceso exclusivo del administrador.

Número de requisito	RF3
Nombre de requisito	Selección de Vehículo
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Los usuarios deben poder elegir un vehículo disponible a través de una interfaz clara y fácil de usar.

Número de requisito	RF4
Nombre de requisito	Reserva con Fecha y Hora
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir la reserva de vehículos para fechas y horas específicas, evitando conflictos de disponibilidad.

Número de requisito	RF5
Nombre de requisito	Calendario de Disponibilidad

Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Se debe implementar un calendario visual que muestre la disponibilidad de cada chofer para facilitar la programación.

Número de requisito	RF6
Nombre de requisito	Filtros de Búsqueda
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir la búsqueda de reservas por rango de fecha, facilitando la visualización de actividades programadas.

Número de requisito	RF7
Nombre de requisito	Validación de reservas
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

No debe ser posible seleccionar el mismo vehículo en la misma fecha y hora para evitar conflictos de programación.

Número de requisito	RF8
Nombre de requisito	Casillero de Sugerencias o Reclamos
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Debe existir un espacio donde los usuarios puedan dejar sugerencias o reclamos relacionados con el sistema.

Número de requisito	RF9
Nombre de requisito	Impresión de Calendario (Administrador)
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El administrador debe tener la opción de imprimir el calendario con las reservas programadas.

Número de requisito	RF10
Nombre de requisito	Modificación de Reservas (Administrador)
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El administrador debe poder realizar modificaciones en las reservas y datos registrados en el sistema.

Número de requisito	RF11
Nombre de requisito	Almacenamiento de Información
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe almacenar la información de todos los vehículos durante un período de al menos 3 meses.

Número de requisito	RF12
Nombre de requisito	Plan de Mantenimiento Preventivo
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe proporcionar un cronograma para el mantenimiento preventivo de los vehículos, informando cuando se acerque el kilometraje establecido para el mantenimiento.

Número de requisito	RF13
Nombre de requisito	Generación de Orden de Mantenimiento
Tipo	X Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	X Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir la generación de órdenes de mantenimiento. Para ello, al hacer clic en un sistema (motor, frenos, sistema eléctrico), se deben desplegar los subsistemas y permitir la generación de la orden.

Número de requisito	RF14
Nombre de requisito	Registro de Mecánicos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación del usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El sistema debe permitir la carga de información sobre los mecánicos responsables de los mantenimientos de un vehículo.

Requisitos no funcionales

Requisitos de rendimiento

Rendimiento General del Sistema

- El sistema deberá soportar usuarios concurrentes.
- El tiempo de respuesta promedio para transacciones críticas no deberá exceder los 2 segundos.
- El sistema deberá ser capaz de manejar varias transacciones por minuto.

Seguridad

Protección contra Accesos No Autorizados

- Se implementará un sistema de autenticación basado en usuario y contraseña.

- Se utilizarán técnicas criptográficas para garantizar la seguridad de la información sensible.
- El sistema mantendrá registros de actividad (logs) para detectar y registrar cualquier acceso no autorizado.

Fiabilidad

- El sistema deberá tener una tasa de fallos bajo durante un período de un año.
- El tiempo promedio entre incidentes deberá ser de al menos 500 horas de tiempo de actividad continuo.

Disponibilidad

- El sistema deberá estar disponible para los usuarios finales al menos el 99.9% del tiempo.
- Se permitirá un tiempo de inactividad para mantenimiento y actualizaciones.

Mantenibilidad

Responsabilidades y Programación de Mantenimiento

- Las tareas de mantenimiento rutinario serán realizadas por el equipo de administración del sistema.
- Se generará un informe estadístico semanal y mensual para evaluar el rendimiento del sistema y planificar el mantenimiento.

Frecuencia de Mantenimiento

- Las actualizaciones de seguridad serán aplicadas mensualmente.
- El mantenimiento preventivo, como la revisión de logs y la optimización del rendimiento, se realizará trimestralmente.

Portabilidad

- El sistema será independiente.
- El código estará escrito en un lenguaje de programación estándar para facilitar la portabilidad.
- El software será compatible con los sistemas operativos Windows, Linux y macOS.

Otros requisitos

Requisitos Culturales y Políticos:

- El sistema respetará las normativas culturales y políticas locales del GAD Azogues.

Requisitos Legales:

- El sistema cumplirá con todas las leyes y regulaciones locales relacionadas con la gestión del parque automotor y la protección de datos.

Anexo 2

Introducción

Bienvenido al manual de usuario de la aplicación Angular. Este manual está diseñado para proporcionar una comprensión detallada de las funcionalidades disponibles en la aplicación y cómo utilizarlas de manera efectiva. La aplicación se desarrolló para proporcionar una plataforma intuitiva y eficiente para la gestión de vehículos, mantenimientos, personal y otras actividades relacionadas en el contexto del GAD Azogues. Con una interfaz amigable y funcionalidades bien estructuradas, nuestra aplicación busca facilitar las tareas diarias de los usuarios y mejorar la eficiencia en la gestión de recursos

En este manual se detalla cómo navegar por la aplicación, acceder a sus diferentes secciones y utilizar cada funcionalidad de manera óptima. Desde el inicio de sesión hasta la gestión de vehículos y mantenimientos, nos hemos esforzado por proporcionar una guía clara y completa que te permita aprovechar al máximo todas las capacidades de la aplicación.

El equipo de desarrollo ha trabajado arduamente para garantizar que la aplicación cumpla con los más altos estándares de calidad y usabilidad. Hemos integrado las últimas tecnologías y mejores prácticas de desarrollo de software para ofrecerte una experiencia fluida y satisfactoria en cada interacción con la aplicación. Además, existe el compromiso con la mejora continua y el equipo estará atento a tus comentarios y sugerencias para seguir mejorando y adaptando la aplicación a tus necesidades.

Navegación

La navegación en la aplicación se refiere a cómo los usuarios se desplazan entre las diferentes páginas y secciones de esta. Esto se logra mediante la configuración de rutas de navegación, que definen la correspondencia entre las URL del navegador y los componentes de Angular que se deben mostrar.

Configuración de rutas

En Angular, la configuración de las rutas de navegación se realiza principalmente en el archivo 'app-routing.module.ts'. Aquí es donde se definen las rutas y se especifica qué componente se debe cargar cuando se accede a una determinada URL.

```
const routes: Routes = [  
  { path: '', component: LoginComponent},  
  { path: 'Vehiculos', component: VehiculosComponent},  
  { path: 'Personas', component: PersonasComponent},  
  { path: 'Sugerencias', component: SugerenciasComponent},  
  { path: 'Reservas', component: ReservasComponent},  
  { path: 'Principal', component: HomeComponent},  
  { path: 'Mantenimientos', component: MantenimientoComponent},  
  { path: 'Repuestos', component: RepuestosComponent}  
];
```

Imagen 1: Configuración de rutas

Control de Acceso

En la aplicación, el acceso a ciertas páginas puede estar restringido según el rol del usuario. Esto significa que solo los usuarios con ciertos roles específicos pueden ver ciertas partes de la aplicación. Por ejemplo, un usuario con el rol de "ADMIN" puede acceder a todas las páginas, mientras que un usuario con el rol de "CHOFER" y "MECANICO" solo puede acceder a algunas páginas específicas.

Implementación en la Interfaz de Usuario

En la interfaz de usuario de la aplicación, la navegación se presenta típicamente mediante menús, barras de navegación u otros elementos visuales. Estos elementos de navegación pueden cambiar dinámicamente según el rol del usuario. Por ejemplo, un usuario con el rol de "ADMIN" puede ver un conjunto completo de opciones de menú, mientras que un usuario con el rol de "CHOFER" y "MECANICO" puede ver un conjunto limitado de opciones.

Consideraciones de Diseño

Al diseñar la navegación de la aplicación, se tuvo en cuenta la usabilidad y la accesibilidad. La navegación sea intuitiva y fácil de usar para los usuarios.

Funcionalidades

La aplicación Angular que hemos desarrollado ofrece una amplia gama de funcionalidades diseñadas para simplificar y optimizar la gestión de vehículos, mantenimientos, personal y más en el ámbito del GAD. Desde el proceso de inicio de sesión hasta la gestión detallada de cada aspecto del sistema, nuestra aplicación proporciona herramientas poderosas y fáciles de usar que se adaptan a las necesidades específicas de los usuarios. A continuación, se presenta una descripción detallada de cada funcionalidad disponible, acompañada de instrucciones claras sobre cómo utilizarlas de manera efectiva para mejorar la eficiencia y la productividad en tus tareas diarias.

Inicio de sesión

Este componente permite a los usuarios iniciar sesión en la aplicación proporcionando su nombre de usuario y contraseña.

Como utilizar

- Abra la página de inicio de sesión de la aplicación.
- Ingrese su nombre de usuario y contraseña en los campos correspondientes.
- Haga clic en el botón "Iniciar Sesión" para acceder a la aplicación.

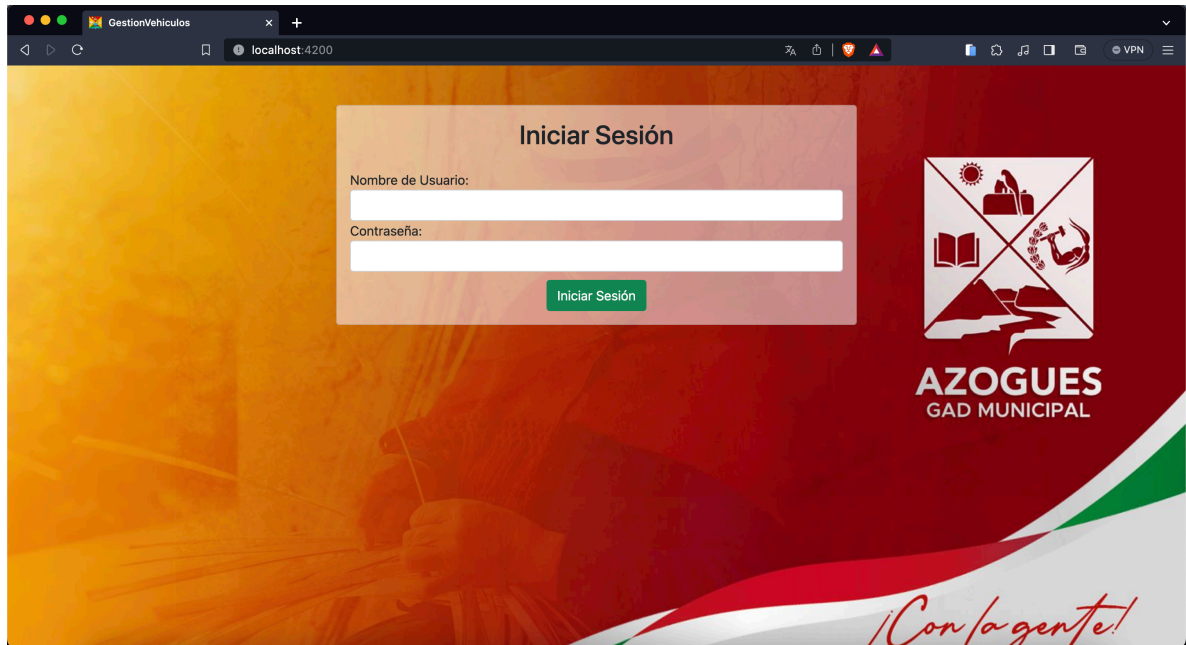


Imagen 2: Página de inicio de sesión

Home

Este componente es la página principal de la aplicación que se muestra después de iniciar sesión. Proporciona acceso a diferentes secciones de la aplicación según el rol del usuario.

Como utilizar

- Después de iniciar sesión, será redirigido automáticamente a la página de inicio (Home) de la aplicación.

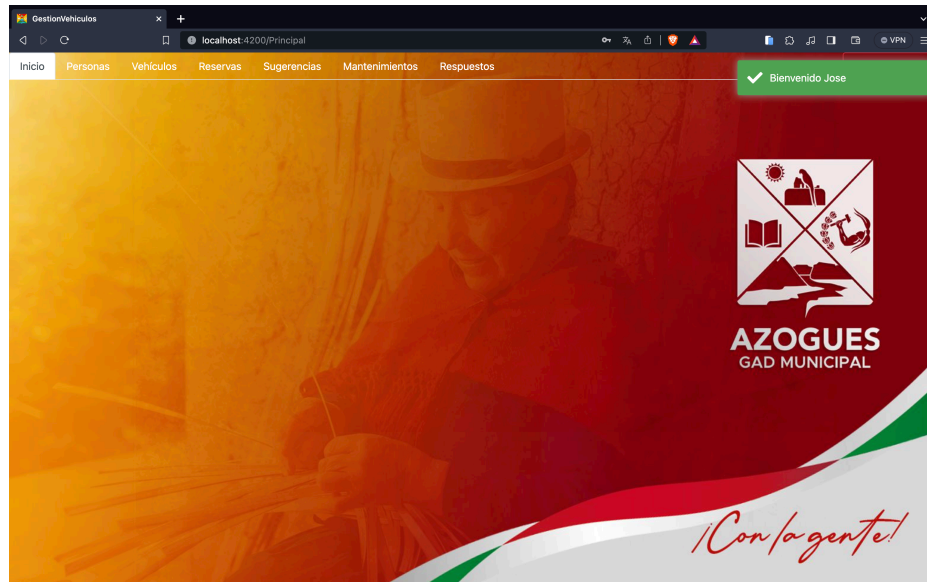


Imagen 3: Página de bienvenida (Home)

- Desde esta página, puede navegar a diferentes secciones de la aplicación utilizando los enlaces proporcionados en la barra de navegación.

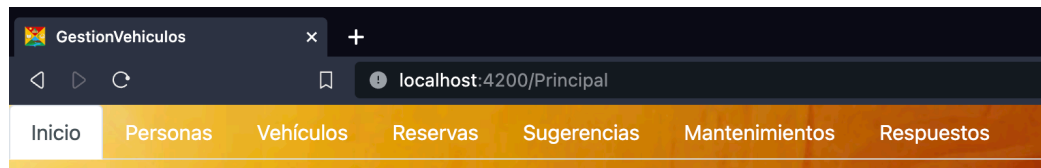


Imagen 4: Barra de navegación

- Para cerrar sesión, haga clic en el botón "Cerrar Sesión" ubicado en la esquina superior derecha de la página.

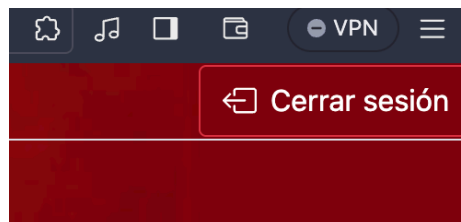


Imagen 5: Cerrar sesión

Mantenimiento

El componente de Mantenimiento está diseñado para permitir a los usuarios administradores y mecánicos gestionar los mantenimientos de los vehículos del GAD Municipal

Como utilizar

- Inicio de Sesión: Inicia sesión en la aplicación con tus credenciales de usuario.
- Exploración de la Lista de Mantenimientos: Una vez dentro, puedes explorar la lista de mantenimientos existentes.

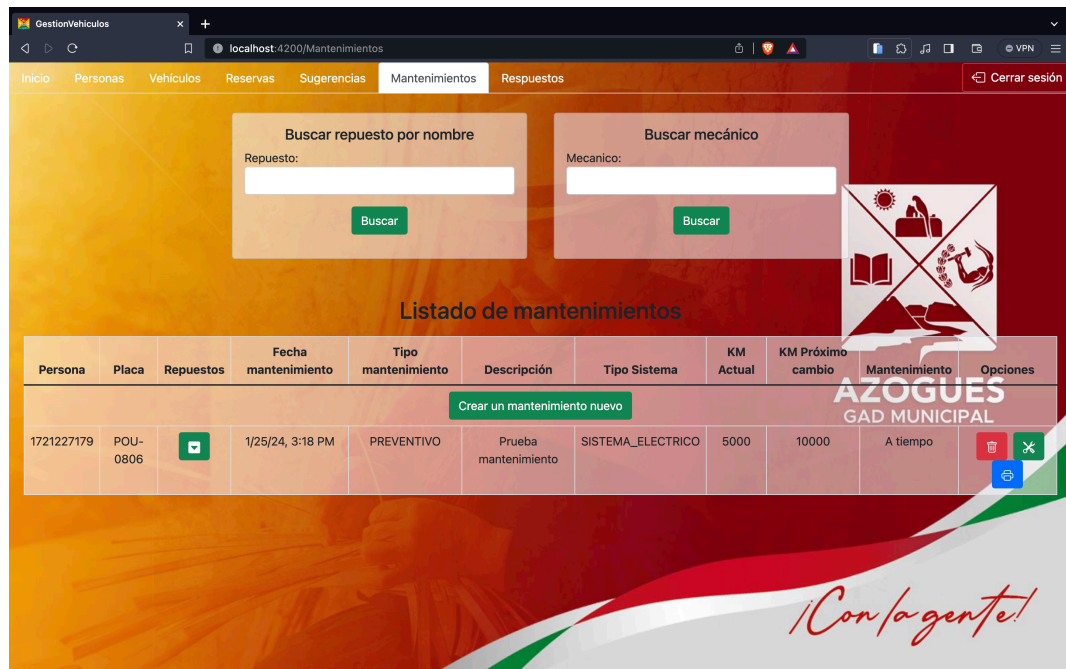


Imagen 6: Página de Mantenimientos

- Creación de un Nuevo Mantenimiento: Para programar un nuevo mantenimiento, haz clic en el botón "Crear un mantenimiento nuevo" y completa el formulario con la información requerida.


Persona	Placa	Repuestos	Fecha mantenimiento	Tipo mantenimiento	Descripción	Tipo Sistema	KM Actual	KM Próximo cambio	Mantenimiento	Opciones
---------	-------	-----------	---------------------	--------------------	-------------	--------------	-----------	-------------------	---------------	----------

Crear un mantenimiento nuevo

Nuevo Mantenimiento

Placa del vehículo:	Tipo de mantenimiento:
<input type="text"/>	<input type="text"/>
Identificación persona:	Descripción:
<input type="text"/>	<input type="text"/>
Fecha del mantenimiento:	Tipo de sistema:
dd/mm/aaaa, --:-- <input type="calendar"/>	<input type="text"/>
	Kilometraje actual:
	<input type="text"/>

Cancelar Crear Mantenimiento



AZOGUES
GAD MUNICIPAL

Imagen 7: Nuevo mantenimiento

- Finalización de Mantenimientos: Cuando un mantenimiento se complete, puedes finalizarlo haciendo clic en el botón del basurero.



Imagen 8: Finalizar mantenimiento

- Búsqueda de Repuestos: Si necesitas agregar repuestos a un mantenimiento, puedes buscarlos por nombre utilizando el campo de búsqueda y luego seleccionar la cantidad deseada.

Buscar repuesto por nombre

Repuesto:

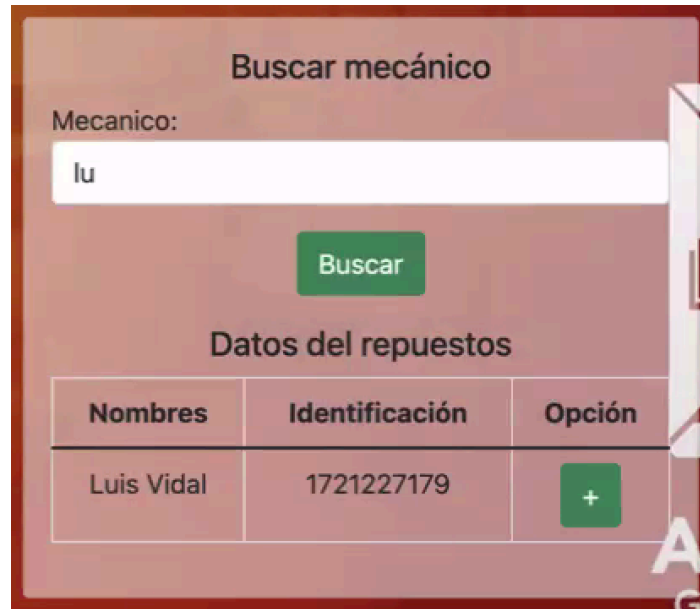
Buscar

Datos del repuestos

Repuesto	Cantidad Total	Opción
Llantas	56	+

Imagen 9: Buscar repuestos

- Búsqueda de Mecánicos: Del mismo modo, puedes buscar mecánicos por nombre para asignarlos a los mantenimientos.



Nombres	Identificación	Opción
Luis Vidal	1721227179	+

Imagen 10: Buscar mecánico

- Gestión de Repuestos Asociados: Una vez agregados los repuestos, puedes ver los detalles de los repuestos asociados y gestionar su cantidad si es necesario.
- Impresión de Detalles: Si deseas imprimir los detalles de un mantenimiento, puedes hacerlo utilizando el botón de impresión.



Imagen 11: Imprimir detalles

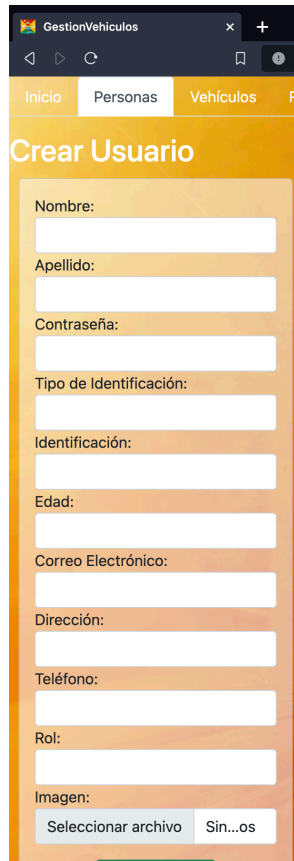
Personas

Este componente permite a los usuarios con el rol de administrador crear, listar, actualizar y eliminar personas en la aplicación.

Como utilizar

- Navegue hasta la página de personas desde el menú de la aplicación.
- En la sección "Crear Usuario", ingrese los datos necesarios de la persona que a crear.

- Haga clic en el botón "Crear Usuario" para crear la persona.



The image shows a mobile application interface for creating a new user. The browser address bar at the top displays 'GestionVehiculos'. Below the browser, there is a navigation bar with tabs for 'Inicio', 'Personas', 'Vehículos', and 'Reservaciones'. The main content area is titled 'Crear Usuario' and contains a form with the following fields: 'Nombre:', 'Apellido:', 'Contraseña:', 'Tipo de Identificación:', 'Identificación:', 'Edad:', 'Correo Electrónico:', 'Dirección:', 'Teléfono:', 'Rol:', and 'Imagen:'. The 'Imagen:' field includes two buttons: 'Seleccionar archivo' and 'Sin...os'.

Imagen 12: Crear nuevo usuario

- En la sección "Lista de los usuarios", puede ver una lista de todas las personas existentes en la aplicación.
- Puede actualizar la información de una persona haciendo clic en el botón "Actualizar" junto a sus detalles.

<p>Imagen: <input type="button" value="Seleccionar archivo"/> <input type="text" value="Sin...os"/></p> <p>ID: <input type="text" value="3"/></p> <p>Nombre: <input type="text" value="Luis"/></p> <p>Apellido: <input type="text" value="Vidal"/></p> <p>Username: <input type="text" value="lvidal"/></p> <p>Password: <input type="text" value="...."/></p> <p>Tipo de Identificación: <input type="text" value="CEDULA"/></p> <p>Identificación: <input type="text" value="1721227179"/></p> <p>Edad: <input type="text" value="25"/></p>	<p>Password: <input type="text" value="...."/></p> <p>Tipo de Identificación: <input type="text" value="CEDULA"/></p> <p>Identificación: <input type="text" value="1721227179"/></p> <p>Edad: <input type="text" value="25"/></p> <p>Correo: <input type="text" value="luis@gmail.com"/></p> <p>Dirección: <input type="text" value="Cuenca"/></p> <p>Teléfono: <input type="text" value="0968562787"/></p> <p>Rol: <input type="text" value="MECANICO"/></p> <p> <input type="button" value="Guardar"/> <input type="button" value="Cancelar"/> <input type="button" value="Eliminar"/> </p>
--	---

Imagen 13: Actualizar usuario

- Para eliminar una persona, haga clic en el botón "Eliminar" correspondiente a la persona que desea eliminar.



<p>Luis Vidal</p> <p>Usuario: lvidal</p> <p>Edad: 25</p> <p>Contraseña: ****</p> <p>Identificación: 1721227179</p> <p>Correo: luis@gmail.com</p> <p>Teléfono: 0968562787</p> <p>Rol: MECANICO</p> <p>Dirección: Cuenca</p> <p> <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> </p>

Imagen 14: Eliminar usuario

- Para cerrar la sesión, haga clic en el botón "Cerrar sesión" en la parte superior derecha de la página.

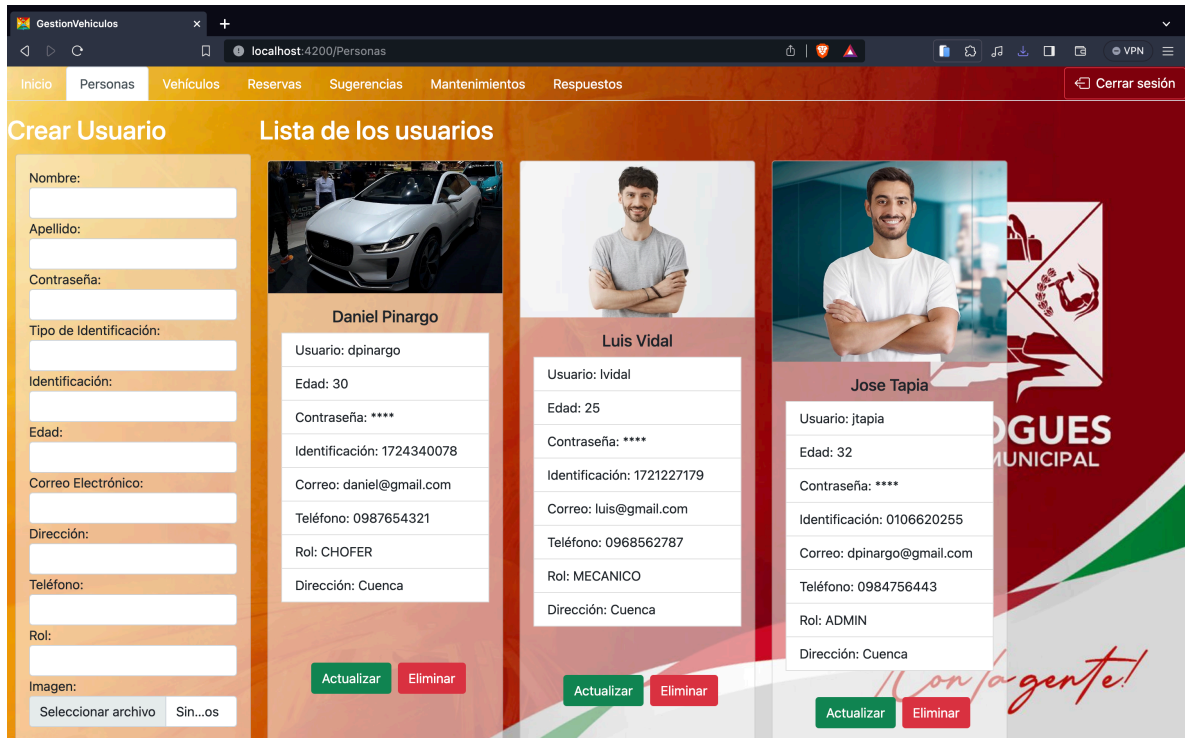


Imagen 15: Página de Personas

Repuestos

Este componente permite a los usuarios administradores y mecánicos gestionar los repuestos disponibles en la aplicación.

Como utilizar

- Accede al menú de Repuestos desde la barra de navegación.
- Utiliza la función de búsqueda por nombre para encontrar repuestos específicos.
- Puedes crear nuevos repuestos llenando el formulario correspondiente.

Nombre	Cantidad	Valor Unitario	Valor Total	Marca de vehículo	Modelo de vehículo	Estado	Registro	Actualizado	Opciones
<input type="text" value="Nombre"/>	<input type="text" value="Cantidad"/>	<input type="text" value="Valor"/>		<input type="text" value="Marca"/>	<input type="text" value="Modelo"/>				<input type="button" value="+"/>

Imagen 16: Crear nuevo repuesto

- Para actualizar un repuesto existente, haz clic en el botón "Actualizar" y modifica los datos en el formulario que aparece.



Imagen 17: Actualizar repuesto

- Para eliminar un repuesto, haz clic en el botón "Eliminar".



Imagen 18: Eliminar repuesto

- Si deseas cerrar sesión, haz clic en el botón "Cerrar sesión".

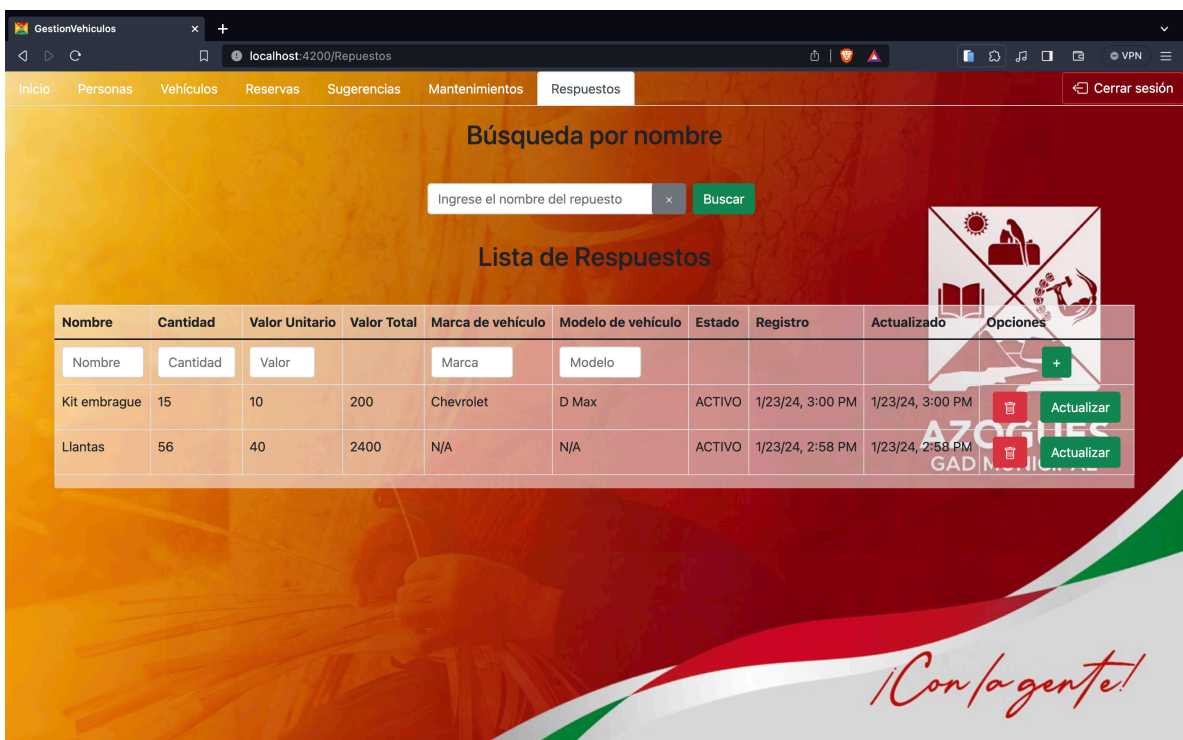


Imagen 19: Página de Repuestos

Reservas

Este componente permite gestionar las reservas de vehículos en la aplicación. Los usuarios pueden buscar vehículos por placa, crear nuevas reservas, ver un listado de reservas existentes, actualizar y eliminar reservas.

Como utilizar

- Navegue a la sección de reservas desde el menú de navegación.
- Utilice los formularios proporcionados para buscar vehículos por placa y entre fechas.



Buscar vehículo por placa

Placa:

POU-0806

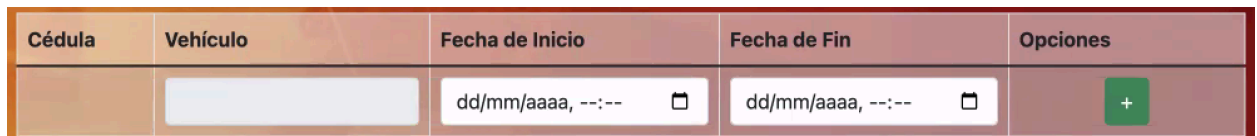
Buscar Vehículo

Datos del vehículo

Placa	Marca	Modelo	Opción
POU-0806	Fiat	Palio	+

Imagen 20: Buscar vehículo por placa

- Puede crear nuevas reservas ingresando los datos requeridos y haciendo clic en el botón "+".



Cédula	Vehículo	Fecha de Inicio	Fecha de Fin	Opciones
		dd/mm/aaaa, --:--	dd/mm/aaaa, --:--	+

Imagen 21: Crear nueva reserva

- Puede ver el listado de reservas existentes, actualizar o eliminar una reserva según sea necesario.

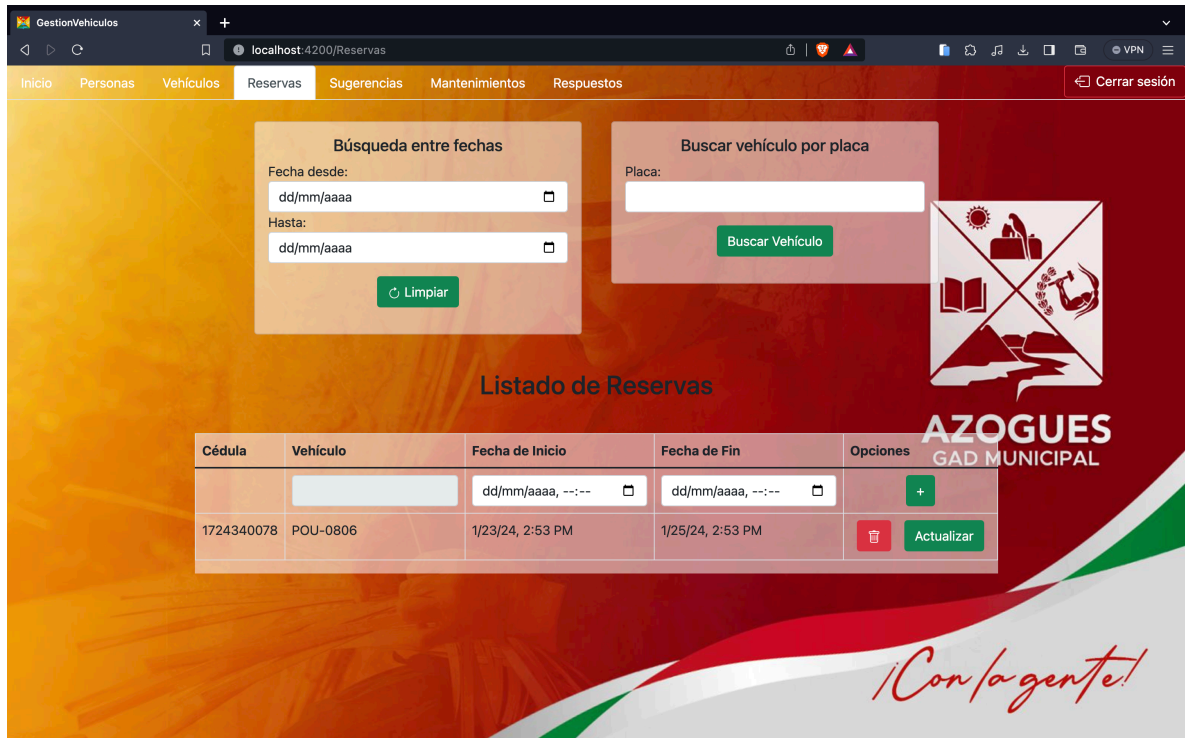


Imagen 22: Página de Reservas

Sugerencias

Este componente permite a los usuarios enviar sugerencias a la aplicación.

Como utilizar

- Navegue hasta la sección de "Sugerencias" en el menú de la aplicación.
- Escriba su sugerencia en el campo proporcionado.
- Haga clic en el botón "+" para enviar su sugerencia.

Sugerencia	Fecha	Opciones
<input type="text" value="Sugerencia"/>		<input type="button" value="+"/>

Imagen 23: Crear nueva sugerencia

- Las sugerencias enviadas se mostrarán en la tabla junto con la fecha de registro.

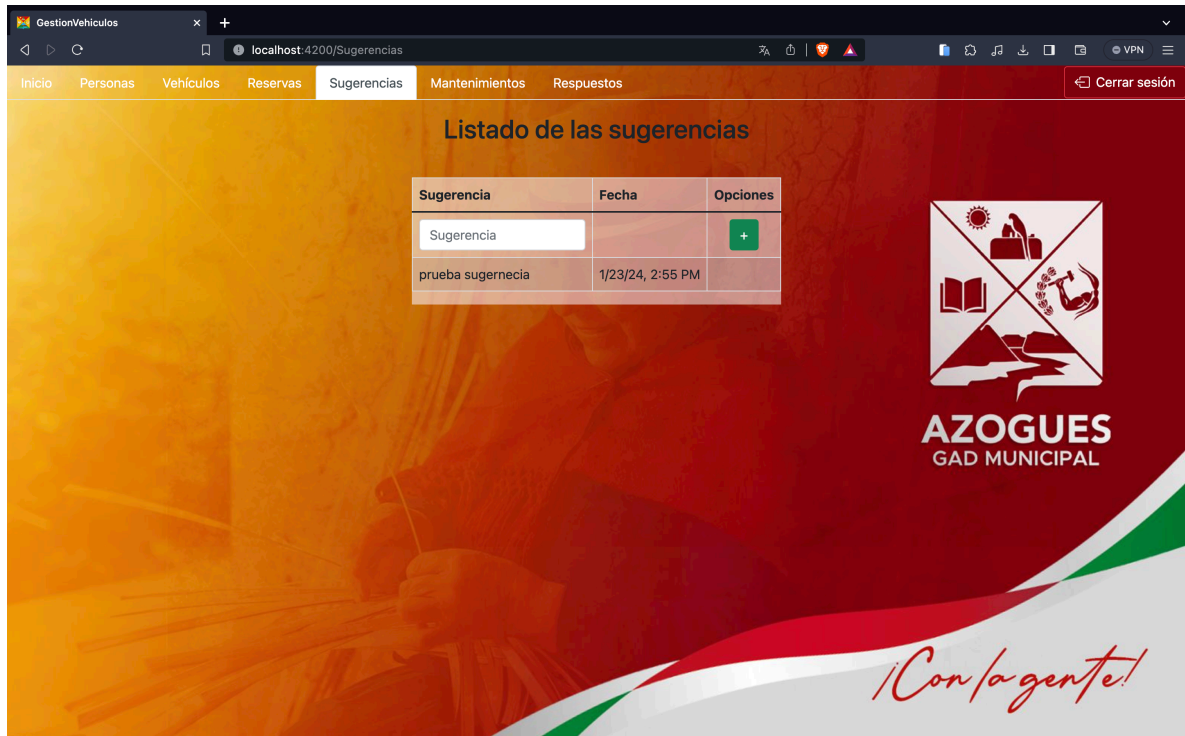


Imagen 24: Página de Sugerencias

Vehículos

El componente de vehículos está diseñado para permitir a los usuarios administrar información sobre vehículos. Proporciona funciones para buscar vehículos por placa, listar vehículos, crear nuevos vehículos, actualizar y eliminar vehículos existentes, así como también agregar y eliminar personas asociadas a cada vehículo.

Como utilizar

- Listar vehículos: Se mostrará una lista de todos los vehículos registrados en la base de datos. La lista incluye detalles como la placa, marca, modelo, chasis, motor, y la persona encargada asociada a cada vehículo. Puedes ver más detalles de cada vehículo haciendo clic en el botón de la flecha hacia abajo.

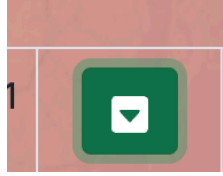


Imagen 25: Mostrar detalles del vehículo

- Crear un nuevo vehículo: Haz clic en el botón "Crear un vehículo nuevo". Completa todos los campos obligatorios en el formulario de creación. Haz clic en el botón "Crear Vehículo" para guardar el nuevo vehículo.

El formulario tiene un encabezado con el título "Nuevo Vehículo" y un botón "Crear un vehículo nuevo" en la parte superior. Los campos de entrada están organizados en una cuadrícula de dos columnas. El campo "Persona encargada" está deshabilitado. En la parte inferior, hay un campo de imagen con un botón "Seleccionar archivo" y el texto "Sin archivos seleccionados", y dos botones de acción: "Cancelar" (verde) y "Crear Vehículo" (azul).

Imagen 26: Crear nuevo vehículo

- Actualizar un vehículo existente: Haz clic en el botón de edición (lápiz) en la fila del vehículo que deseas actualizar. Realiza los cambios necesarios en los campos del formulario de edición. Haz clic en el botón "Guardar" para aplicar los cambios.
- Eliminar un vehículo: Haz clic en el botón de eliminación (papelera) en la fila del vehículo que deseas eliminar. Confirma la eliminación en el mensaje de confirmación.



Imagen 27: Eliminar persona del vehículo

- Agregar y eliminar personas asociadas a un vehículo: Para agregar una persona al vehículo, busca la persona por nombre en el campo de búsqueda, luego haz clic en el botón "+" en la fila correspondiente. Para eliminar una persona asociada al vehículo, haz clic en el botón "Eliminar persona" en la sección de detalles de la persona.

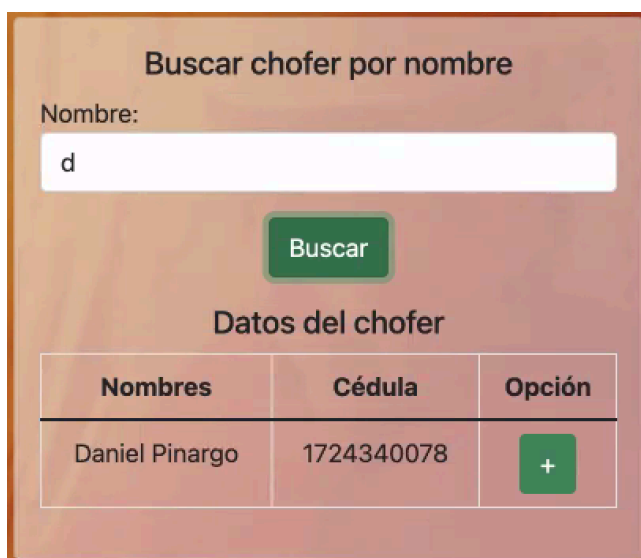


Imagen 28: Buscar chofer

- Cerrar sesión: clic en el botón Cerrar sesión en la esquina superior derecha para cerrar la sesión actual.

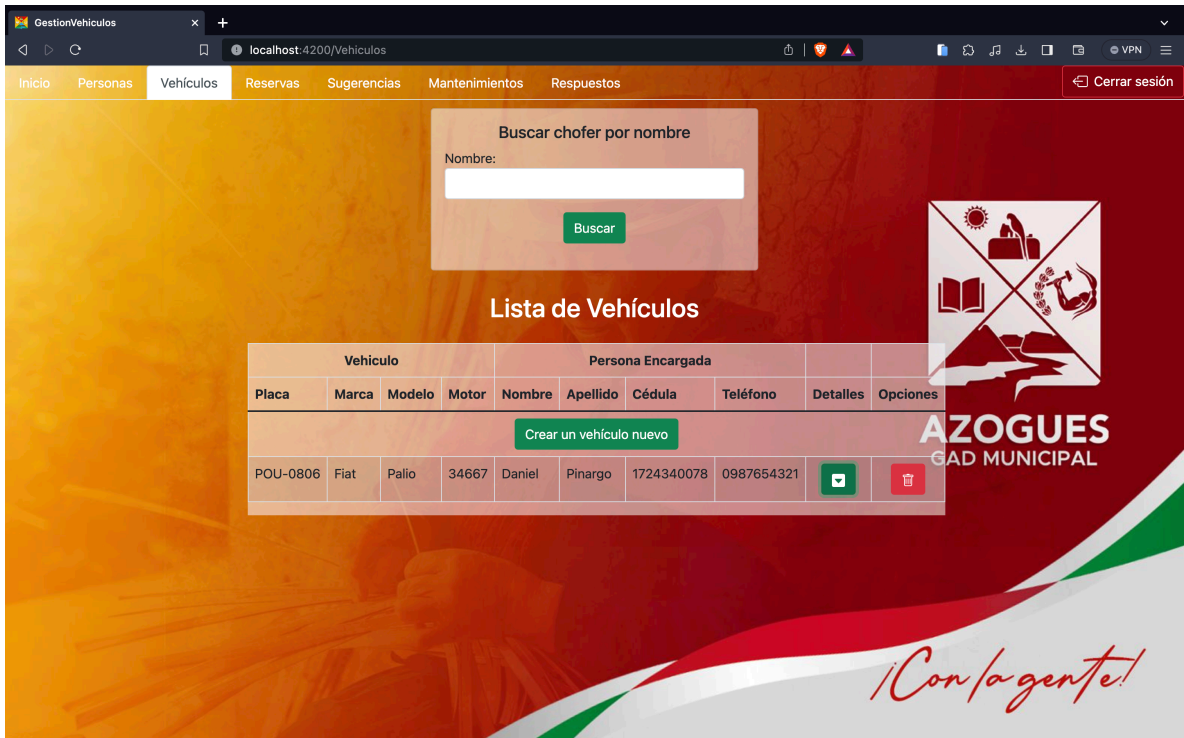


Imagen 29: Página de Vehículos

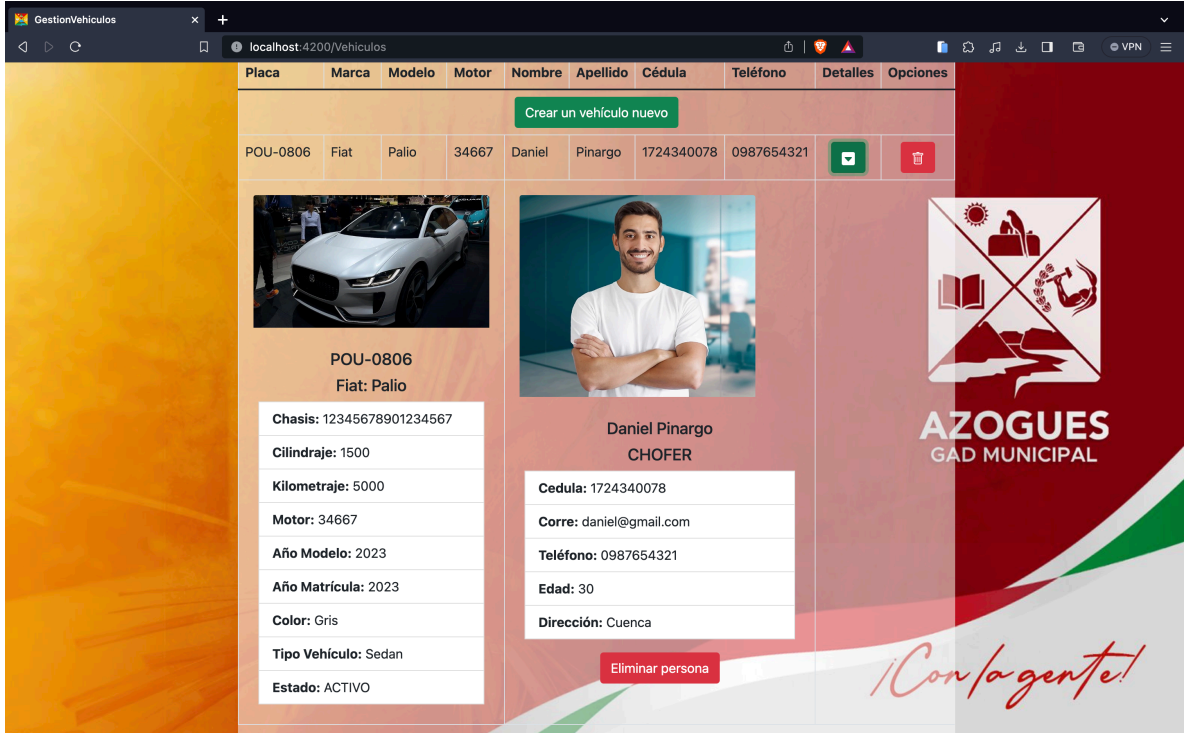


Imagen 30: Visualización de todos los detalles del vehículo

Anexo 3

El siguiente enlace almacena al código fuente desarrollado como parte de este proyecto. Este código representa la implementación práctica de los conceptos y funcionalidades descritos en el presente documento. A través del enlace proporcionado, los interesados pueden acceder al código completo, explorar su estructura y funcionamiento, y realizar cualquier evaluación o análisis adicional según sea necesario. El código está organizado de manera que refleje la arquitectura y diseño discutidos previamente, lo que facilita su comprensión y utilización por parte de desarrolladores y otros profesionales interesados en este proyecto.

Enlace del código: <https://github.com/nando8082/gestionVehiculos>