
PROYECTO DE TITULO

FixWay

INTEGRANTES:

- Benjamín Isaac Garrido Aguilera
- Sebastian Esteban Quintana Guitierrez
- Samuel Gonzalo Gajardo Bazaes

SECCION: TIHI84/D-IEI-N8-P1-C2/D

ASIGNATURA: Proyecto de Titulo – TIHI84

PROFESOR: Boris Astorga – Manuel Rojas

FECHA: 11/12/2024

Contenido

1	Introducción	6
2	Identificación del Problema	7
2.1	Actualización y Justificación del problema.....	7
2.1.1	Descripción de la organización.....	8
	Figura 1	9
2.1.2	Descripción del problema	10
2.1.3	Relevancia del problema	11
2.1.4	Complejidad de la problemática	11
3	Levantamiento de Requerimientos	13
3.1	Determinación de los instrumentos a utilizar	13
3.2	Detalle de los requerimientos	14
4	Objetivos del Proyecto.....	16
4.1	Solución tecnológica.....	16
4.1.1	Formulación de la Solución	16
4.2	Impacto de la solución	17
4.2.1	Proceso de negocio afectado.....	17
4.2.2	Registro de Interesados.....	17
4.2.3	Indicadores de gestión	17
4.2.4	Niveles de servicio.....	18
4.3	Objetivos del proyecto.....	19
4.3.1	Objetivo General	19
4.3.2	Objetivo Especifico	19
5	Implementación de los KPI	22
5.1	Descripción de KPI	22
5.2	Descripción de SLA	27
6	Metodología de Trabajo.....	31
6.1	Metodología de Desarrollo de la solución.....	31
6.2	Duración y cronograma.....	33
	Figura 2:.....	33
	Figura 3:	33
6.3	Equipo de Trabajo.....	34
6.4	Plan de recursos	35

7	Detalle de las Tecnologías a Implementar	36
7.1	Análisis Cualitativo/Cuantitativo de las tecnologías que serán implementadas	36
	Figura 4:	38
	Figura 5:.....	40
	Figura 6:	42
	Figura 7:	43
	Figura 8	46
	Figura 9:	49
	Figura 10:	51
7.2	Herramientas, aplicaciones, lenguajes y componentes que serán Implementados	52
	Figura 11:	52
	Figura 12:.....	52
	Figura 13:	53
	Figura 14	53
	Figura 15:	54
	Figura 16:	54
	Figura 17:	55
	Figura 18:	55
	Figura 19:	56
	Figura 20:.....	56
	Figura 21:	57
	Figura 22:	57
	Figura 23:	58

8	Detalle de la Arquitectura a implementar	59
8.1	Diagrama BPMN	59
	Figura 24:	59
	Figura 25:	60
	Figura 26:	61
	Figura 27:	62
	Figura 28:	63
	Figura 29:	64
	Figura 30:	65
	Figura 31:	66
	Figura 32:	67
8.2	Diagramas de caso de uso	68
	Figura 33:	68
	Figura 34:	69
	Figura 35:	70
	Figura 36:	71
	Figura 37:	72
	Figura 38:	73
	Figura 39:	74
	Figura 40:	75
8.3	Diagrama de componentes	76
	Figura 41:	76
8.4	Modelo de datos	77
	Figura 42:	77
8.5	Topologías de comunicación	78
	Figura 43:	78
8.6	Diagrama de Infraestructura	79
	Figura 44:	79
8.7	Diagrama de Arquitectura	80
	Figura 45:	80

9	Plan de Pruebas y Aseguramiento de Calidad	81
9.1	Plan de Pruebas.....	81
	Figura 46:	84
	Figura 47:	84
	Figura 48:.....	85
	Figura 49:.....	86
	Figura 50	86
	Figura 51:	87
	Figura 52:	87
	Figura 53:	88
9.2	Normas y Estándares.....	89
	Estándares de Desarrollo de Software:.....	89
10	Conclusiones	92
11	Bibliografía	96
12	Anexos.....	101
	Figura 54:	101

1 Introducción

El proyecto FixWay tiene como objetivo fundamental mejorar la comunicación interna y externa en los talleres mecánicos mediante la implementación de un software especializado que optimice tanto la eficiencia operativa como la calidad del servicio ofrecido. En un entorno altamente competitivo, donde la satisfacción del cliente es crucial, la simplificación de los procesos de cotización, seguimiento de trabajos y la interacción con los clientes se presenta como una estrategia clave para transformar las operaciones del taller y posicionarlo como un líder en el sector automotriz.

Uno de los principales beneficios que ofrece FixWay es la capacidad de permitir a los mecánicos acceder de manera rápida y precisa a la información detallada de los vehículos y trabajos previamente registrados en el sistema. Este acceso ágil a la información facilitará diagnósticos más certeros y reparaciones más eficientes, lo que reducirá significativamente los tiempos de espera y mejorará la calidad del servicio. Como resultado, se espera un aumento considerable en la satisfacción de los clientes, quienes experimentarán un servicio más rápido y preciso, lo que, a su vez, contribuirá a la consolidación de la reputación del taller en el mercado.

Además, FixWay permitirá a los talleres proporcionar actualizaciones rápidas y transparentes sobre el estado de los vehículos, lo que generará mayor confianza en los servicios ofrecidos. La capacidad de informar en tiempo real sobre el progreso de las reparaciones disminuirá la incertidumbre de los clientes, mejorando su experiencia general y aumentando la fidelidad hacia el taller.

Finalmente, el fortalecimiento de la comunicación interna entre administradores, mecánicos y vendedores será otro pilar esencial de FixWay. Al mejorar la coordinación y reducir los errores de comunicación, se logrará una toma de decisiones más ágil y efectiva, optimizando los recursos y maximizando la productividad del taller. En conjunto, el software no solo busca mejorar la eficiencia operativa, sino también elevar la calidad del servicio y fomentar una relación más sólida y confiable con los clientes, posicionando al taller como un competidor destacado en el sector automotriz.

2 Identificación del Problema

La aplicación FixWay pretende enfrentar los problemas relacionados con la gestión eficiente de los registros de mantenimientos de vehículos. Actualmente, los usuarios de la aplicación, como administradores, mecánicos y vendedores, tienen dificultades para coordinar y organizar la información de los servicios realizados. Esto provoca errores en el registro de datos, complicaciones para llevar un seguimiento adecuado del historial de mantenimientos y la ejecución de diagnósticos precisos. Además, la falta de una comunicación fluida entre los diferentes roles en la aplicación limita la capacidad de ofrecer una atención eficiente, lo que impacta directamente en la experiencia del usuario y la confiabilidad del sistema.

2.1 Actualización y Justificación del problema

La actualización del problema en FixWay, como software de gestión de mantenimientos, radica en la falta de funcionalidades optimizadas para registrar y organizar de manera eficiente los datos relacionados con los servicios de mantenimiento. Actualmente, los usuarios experimentan problemas con la falta de integración en la visualización y actualización de información, lo que genera ineficiencia en la gestión de las órdenes de trabajo y dificulta la correcta planificación de tareas. Esto puede resultar en tiempos de respuesta prolongados y en la acumulación de errores humanos al procesar los registros manualmente o de forma descentralizada.

Justificación: La optimización de estos aspectos en el software es fundamental para garantizar un flujo de trabajo más ágil y reducir los errores en el manejo de datos. Al mejorar la funcionalidad de FixWay para gestionar mantenimientos, los usuarios podrán acceder fácilmente a la información actualizada, permitiendo una mejor toma de decisiones y asegurando que los procesos sean más eficaces y precisos. Esto no solo mejorará la experiencia del usuario, sino que también hará que el software sea más competitivo en el mercado.

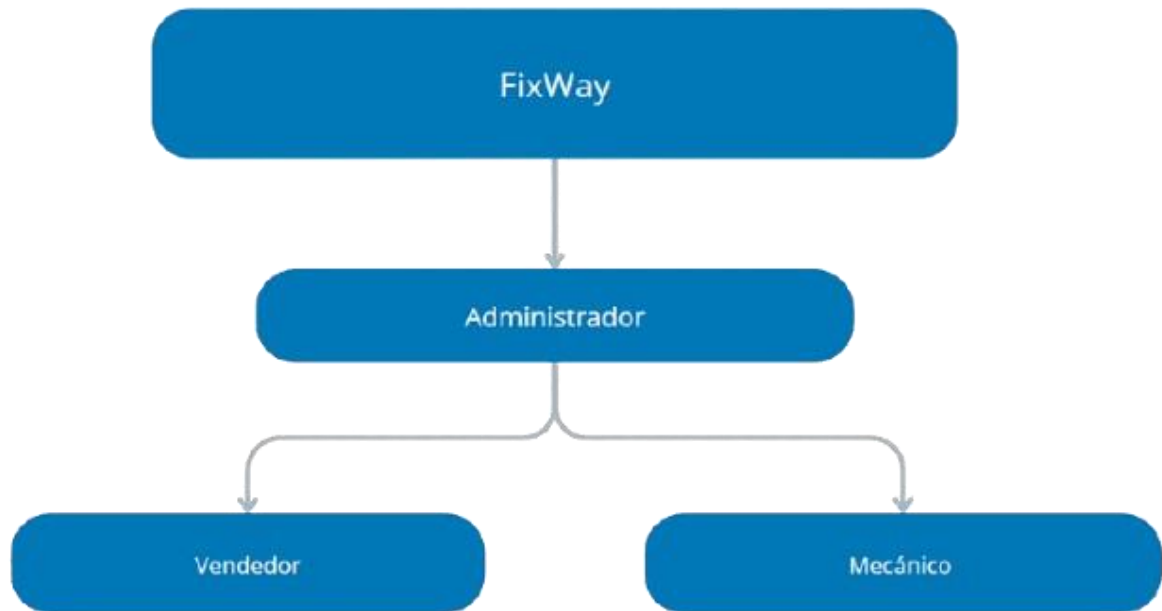
2.1.1 Descripción de la organización

FixWay es una aplicación de software diseñada específicamente para atender las necesidades de empresa en el sector de la mecánica automotriz. La organización detrás de FixWay se enfoca en proporcionar una solución tecnológica sencilla y eficiente que permita a pequeños talleres mejorar la gestión de sus procesos operativos, como el mantenimiento de vehículos, la comunicación con los clientes, y la optimización del flujo de trabajo. La estructura del software está orientada a usuarios con diferentes roles dentro del taller, facilitando la distribución de responsabilidades y tareas:

- **Administrador:** Encargado de gestionar los usuarios, supervisar las operaciones del taller, y controlar las órdenes de trabajo. El administrador también tiene acceso a informes y estadísticas que le permiten evaluar el rendimiento del taller.
- **Mecánico:** Responsable de registrar y actualizar la información relacionada con las reparaciones, contara con la toma de tareas y mantenimientos de los vehículos. Este rol facilita el seguimiento de cada trabajo en curso, mejorando la precisión en los diagnósticos y tiempos de ejecución.
- **Vendedor:** Focalizado en la atención al cliente, el vendedor se encarga de crear cotizaciones, gestionar pagos, y asegurar que los clientes reciban la información necesaria sobre sus vehículos. También ayuda a mantener la transparencia en el proceso de facturación.

Antecedentes de la Organización

Figura 1



Organigrama de la Aplicación (Creación Propia)

Diagnóstico de la situación actual

El software presenta problemas de dispersión de datos y falta de accesibilidad a la información relevante sobre los vehículos y las reparaciones previas, lo que dificulta a los usuarios (mecánicos y administradores) realizar diagnósticos y seguimientos efectivos. Además, la comunicación con los clientes a través de la aplicación no está suficientemente centralizada, lo que provoca retrasos en las actualizaciones sobre el estado de las reparaciones. Esto afecta la eficiencia de la aplicación y la experiencia del usuario, reduciendo la eficacia del servicio que ofrece FixWay

2.1.2 Descripción del problema

La aplicación FixWay enfrenta varios problemas que limitan su capacidad de proporcionar un servicio óptimo a empresas del sector de la mecánica automotriz. Uno de los problemas más significativos es la falta de integración entre las diferentes funciones del software, lo que resulta en una gestión ineficiente de las órdenes de trabajo y la comunicación interna.

Los talleres mecánicos suelen operar con múltiples procesos, desde la recepción del vehículo hasta la entrega final. Sin embargo, sin una plataforma unificada, los mecánicos, administradores y vendedores pueden verse obligados a utilizar diferentes herramientas o métodos para acceder a la información necesaria. Esta fragmentación provoca demoras en el flujo de trabajo y dificulta la coordinación entre los distintos roles.

Además, la falta de una interfaz intuitiva puede hacer que los usuarios se sientan abrumados o confusos, lo que podría llevar a errores en la entrada de datos y en la gestión de las órdenes. Esto no solo afecta la productividad, sino que también puede impactar negativamente en la satisfacción del cliente, ya que la comunicación sobre el estado de las reparaciones no es tan fluida como debería.

Por último, la ausencia de reportes analíticos y métricas en tiempo real dificulta la toma de decisiones estratégicas. Los administradores carecen de información crítica sobre el rendimiento del taller y la efectividad de los servicios ofrecidos, lo que limita su capacidad para implementar mejoras continuas y adaptarse a las necesidades cambiantes del mercado.

2.1.3 Relevancia del problema

La relevancia del problema que enfrenta la aplicación FixWay radica en su impacto directo en la operatividad y sostenibilidad de empresas del sector de la mecánica automotriz. Estos talleres, a menudo limitados en recursos, dependen de herramientas eficientes para gestionar sus operaciones diarias. La falta de un software integrado y fácil de usar puede significar la diferencia entre un servicio de calidad y una experiencia deficiente para los clientes.

En un entorno donde la competencia es feroz, los talleres que no pueden optimizar su gestión corren el riesgo de perder clientes frente a aquellos que ofrecen un servicio más ágil y transparente. La incapacidad de coordinar eficazmente las órdenes de trabajo y la comunicación entre mecánicos, administradores y vendedores puede resultar en retrasos en las reparaciones, errores en las cotizaciones y, en última instancia, insatisfacción del cliente. Esta situación puede llevar a una disminución de la lealtad del cliente y a la pérdida de reputación en el mercado.

Además, en un sector que cada vez más se está digitalizando, la falta de análisis de datos puede limitar la capacidad de los talleres para adaptarse a las tendencias del mercado y a las necesidades de sus clientes. Sin información sobre el rendimiento de sus servicios, los administradores no pueden identificar áreas de mejora, optimizar procesos ni realizar ajustes estratégicos que aseguren el crecimiento y la rentabilidad del negocio.

2.1.4 Complejidad de la problemática

La complejidad del problema en FixWay radica en varios factores interrelacionados que afectan tanto la funcionalidad de la aplicación como la experiencia del usuario. En primer lugar, la dispersión de los datos y la falta de integración entre diferentes módulos de la aplicación complican el acceso y manejo de información crítica, como historiales de reparaciones y diagnósticos previos. Esto no solo requiere una reestructuración técnica significativa, sino también un rediseño de la arquitectura de datos para asegurar la cohesión y accesibilidad.

Además, la implementación de un sistema de comunicación eficiente entre usuarios internos (mecánicos y administradores) y externos (clientes) implica desarrollar y optimizar funciones que permitan actualizaciones en tiempo real y una mayor transparencia. Esto puede requerir la adopción de nuevas tecnologías, la mejora de la interfaz de usuario y la sincronización de datos en diferentes dispositivos y plataformas, lo que añade otra capa de complejidad técnica.

Al abordar estos desafíos debe realizarse sin interrumpir el funcionamiento diario de la aplicación, lo que implica una planificación cuidadosa y una implementación gradual. También es necesario considerar la capacitación de los usuarios en las nuevas funcionalidades y asegurar una transición suave para mantener la confianza y satisfacción de los clientes. La interdependencia de estos factores hace que la resolución del problema sea un desafío complejo que requiere un enfoque multidisciplinario y coordinado.

3 Levantamiento de Requerimientos

3.1 Determinación de los instrumentos a utilizar

Tauri:

- **Propósito:** Tauri es un framework que permite crear aplicaciones de escritorio ligeras y seguras utilizando tecnologías web como HTML, CSS y JavaScript. Es especialmente útil para crear aplicaciones que requieran una interfaz gráfica, pero que deseen evitar el peso de los frameworks tradicionales.
- **Uso:** Es ideal para desarrollar aplicaciones que necesiten ejecutarse en múltiples plataformas (Windows, MacOS, Linux) con un rendimiento eficiente y bajo consumo de recursos.

Vitejs:

- **Propósito:** Vite es un build tool y un servidor de desarrollo rápido y ligero, enfocado en un inicio rápido y en la entrega de archivos eficientes en producción. Es compatible con la mayoría de los frameworks frontend modernos.
- **Uso:** Vite se utiliza para el desarrollo rápido de aplicaciones web, permitiendo hot-module replacement (HMR) y un entorno de desarrollo ágil, especialmente cuando se trabaja con frameworks como React, Vue o Svelte.

Firebase:

- **Propósito:** Firebase es una plataforma de desarrollo de aplicaciones móviles y web que ofrece una variedad de servicios, incluyendo bases de datos en tiempo real, autenticación, almacenamiento de archivos, hosting y funciones en la nube.
- **Uso:** Firebase se utiliza para gestionar el backend de aplicaciones sin necesidad de configurar un servidor completo. Es ideal para aplicaciones que requieren autenticación de usuarios, almacenamiento de datos en tiempo real y servicios backend escalables.

Expo:

- Propósito: Expo es un conjunto de herramientas y servicios para crear aplicaciones móviles con React Native. Simplifica el proceso de desarrollo móvil, permitiendo que los desarrolladores se centren en la lógica de la aplicación en lugar de la configuración del entorno.
- Uso: Se utiliza principalmente para el desarrollo de aplicaciones móviles en React Native, facilitando la creación, prueba y despliegue de aplicaciones sin necesidad de lidiar con la configuración nativa compleja.

3.2 Detalle de los requerimientos

- Mesa de Trabajo (Workshop)

Se realizó una mesa de trabajo con el objetivo de obtener la opinión de los trabajadores sobre el sistema que se quiere desarrollar. Esta reunión tuvo como finalidad capturar una visión clara de las necesidades y expectativas en el entorno de trabajo. La principal necesidad identificada es contar con un sistema que agilice la comunicación interna, especialmente en lo relacionado con la cotización y la visualización de los trabajos en curso.

- Análisis de Documentación

Durante la mesa de trabajo, se identificó que la principal demanda de los trabajadores es mejorar la comunicación dentro de la empresa. También se destacó la necesidad de que el software facilite el flujo de trabajo y mejore la eficiencia. Además, se implementará un módulo de análisis de datos en FixWay, que permitirá centralizar y gestionar toda la información relevante de la empresa dentro de la aplicación.

- Usabilidad

La aplicación FixWay, tanto en su versión móvil como en la web, será diseñada para ser amigable y fácil de usar, adaptándose al entorno de trabajo. Permitirá a los usuarios, incluyendo administradores y clientes, interactuar con el sistema de manera flexible y eficiente. El sistema incluirá un escáner de patentes de vehículos, que permitirá verificar rápidamente la información del vehículo. Si la patente se encuentra en la base de datos, se mostrará toda la información relacionada; en caso contrario, se generará un mensaje de error.

- Eficiencia

FixWay deberá soportar el escaneo de múltiples patentes de vehículos de manera simultánea sin interrupciones. El sistema no debería tardar más de 5 segundos en proporcionar la información completa del vehículo escaneado. Además, se garantizará la funcionalidad del sistema en cualquier momento del día, asegurando la seguridad y disponibilidad de los datos.

- Seguridad Lógica y de Datos

FixWay almacenará la información de los vehículos en una base de datos segura para brindar confianza a los usuarios y administradores. Se realizará un respaldo de la información cada 24 horas para asegurar la protección de los datos. Además, se implementará el algoritmo RSA, ampliamente reconocido por su eficacia en la seguridad de datos. Si el sistema detecta una vulneración, se bloqueará temporalmente para resolver el problema, tras lo cual el administrador podrá restaurar el funcionamiento normal de la aplicación.

4 Objetivos del Proyecto.

4.1 Solución tecnológica

La solución tecnológica propuesta para FixWay se centra en el desarrollo de un software de gestión integral diseñado específicamente para microempresas del sector de la mecánica automotriz. Este software tiene como objetivo centralizar y optimizar los procesos operativos del taller, mejorando así la eficiencia y la calidad del servicio brindado a los clientes.

4.1.1 Formulación de la Solución

FixWay ofrecerá diversas funcionalidades esenciales para el funcionamiento eficaz del taller:

- **Gestión de Órdenes de Trabajo:** Este módulo permitirá a los mecánicos y administradores recibir, asignar y supervisar las órdenes de trabajo de manera eficiente, asegurando que cada tarea se complete de forma oportuna y con los estándares de calidad requeridos.
- **Generación de Cotizaciones y Facturas:** El software facilitará la creación de cotizaciones precisas y la emisión de facturas, lo que permitirá a los vendedores ofrecer un servicio más rápido y eficiente, reduciendo el tiempo de espera para los clientes.
- **Seguimiento de Proyectos y Estado de Vehículos:** A través de esta funcionalidad, los clientes podrán consultar el estado de sus vehículos en tiempo real, lo que aumentará la transparencia y la confianza en los servicios ofrecidos por el taller.
- **Base de Datos de Clientes y Vehículos:** FixWay mantendrá un registro detallado de cada cliente y su vehículo, incluyendo el historial de reparaciones y mantenimiento. Esto permitirá a los mecánicos realizar diagnósticos más informados y personalizados.
- **Informes y Análisis:** Proporcionará herramientas de análisis de datos que permitirán a los administradores evaluar el rendimiento del taller, identificar áreas de mejora y tomar decisiones estratégicas basadas en información precisa.

4.2 Impacto de la solución

4.2.1 Proceso de negocio afectado

La implementación de FixWay mejorará significativamente el proceso de gestión de mantenencias en el taller mecánico, automatizando tareas que antes se realizaban de forma manual, como el seguimiento de reparaciones y la generación de cotizaciones. Esto permitirá una mayor transparencia y eficiencia, tanto para los empleados como para los clientes.

4.2.2 Registro de Interesados

- Mecánicos: Usuarios directos de la plataforma, quienes ingresarán los datos de las mantenencias y controlarán el progreso de las reparaciones.
- Administradores del taller: Supervisarán las operaciones y gestionarán los datos de clientes, vehículos y facturas.
- Clientes: Tendrán acceso a reportes sobre las mantenencias de sus vehículos mediante la web.
- Vendedor: Podrá realizar ventas mediante el sistema, pero también podrá realizar mediante la web.

4.2.3 Indicadores de gestión

Los indicadores de gestión son las métricas más importantes del proyecto, que se definen en la primera etapa del ciclo de vida de un proyecto. Su objetivo principal es facilitar la medición del progreso del proyecto, su eficiencia y finalmente el éxito.

- Tasa de finalización de mantenencias: Proporción de mantenencias completadas dentro del tiempo estimado, lo que refleja la eficiencia en la gestión de los trabajos asignados.
- Satisfacción del cliente: Medida mediante encuestas post-servicio, evaluando la percepción del cliente sobre la calidad del servicio recibido.

- Reducción de errores en la facturación: Comparativa entre los errores ocurridos antes y después de la implementación de FixWay, para evaluar la mejora en la precisión y la eficiencia del proceso de facturación.
- Tiempo de respuesta: Rapidez con la que se generan cotizaciones y facturas, lo que indica la eficiencia en la gestión administrativa y en la interacción con el cliente.

4.2.4 Niveles de servicio

Los acuerdos de nivel de servicio (SLA) son contratos entre el proveedor de subcontratación y tecnología que definen el nivel de servicio que dicho proveedor se compromete a brindar al cliente. Estos acuerdos describen métricas como tiempo de actividad, tiempo de entrega, tiempo de respuesta y tiempo de resolución. Un SLA también especifica el curso de acción en caso de que no se cumplan los requisitos.

- Disponibilidad: La plataforma debe estar disponible al menos el 99% del tiempo, considerando posibles mantenimientos.
- Tiempo de respuesta: Las acciones del sistema (generación de reportes, registro de datos) deben realizarse en un tiempo inferior a 3 segundos.
- Seguridad: Implementación de encriptación de datos y autenticación robusta con tokens JWT para asegurar el acceso.

4.3 Objetivos del proyecto

4.3.1 Objetivo General

El objetivo general del software FixWay es proporcionar una solución integral y accesible para la gestión de servicios de mecánica automotriz. Este software busca optimizar la comunicación y el seguimiento de trabajos, permitiendo a los usuarios, incluidos mecánicos, administradores y vendedores, acceder a información crítica de manera rápida y eficiente. A través de una interfaz intuitiva y funcionalidades específicas, FixWay tiene como meta mejorar la experiencia del cliente, aumentando la satisfacción y la confianza en los servicios mecánicos, al tiempo que permite a los talleres optimizar sus procesos internos, aumentar su competitividad en el mercado. FixWay aspira a convertirse en una herramienta clave que facilite la gestión de servicios automotrices, promoviendo la transparencia y la eficiencia en la interacción entre los talleres y sus clientes.

4.3.2 Objetivo Especifico

1. Crear una interfaz de usuario sencilla y accesible:

- Diseño intuitivo:

Desarrollar una interfaz que permita a mecánicos y administradores registrar fácilmente las mantenciones realizadas en cada vehículo. La interfaz debe ser clara y amigable para todos los usuarios, independientemente de su nivel técnico.

- Funcionalidades clave:

- ✓ Búsqueda rápida de registros para encontrar información relevante sin complicaciones.
- ✓ Categorización de tipos de mantención.
- ✓ Adición de notas y fotos a cada entrada, enriqueciendo la documentación de cada servicio.

2. Automatizar la generación de cotizaciones y facturas:

- Sistema automatizado: Desarrollar un sistema que genere automáticamente cotizaciones y facturas basadas en los datos ingresados sobre mantenciones, piezas y mano de obra. Esto reducirá el margen de error humano y mejorará la eficiencia en el proceso de facturación.
- Documentos claros y profesionales: Facilitar la creación de cotizaciones y facturas bien estructuradas, asegurando que los clientes reciban información precisa y oportuna sobre los costos de los servicios prestados.

3. Proveer un sistema de roles y permisos:

- Protección de datos: Establecer un sistema de roles y permisos que garantice que solo las personas autorizadas puedan acceder a información sensible y realizar acciones críticas dentro del sistema.
- Niveles de acceso: Definir diferentes niveles de acceso según el rol de cada usuario (administrador, mecánico, vendedor), asegurando que cada uno tenga acceso solo a la información y funcionalidades necesarias para su trabajo.

4. Fomentar la escalabilidad y adaptabilidad del software:

- Escalabilidad: Diseñar el software de manera que sea escalable y adaptable, permitiendo que se ajuste a las necesidades cambiantes de los talleres mecánicos y que pueda crecer junto con el negocio.
- Integración futura: Facilitar la integración de nuevas funcionalidades y servicios en el futuro, garantizando que FixWay pueda seguir siendo relevante y útil a medida que evolucionan las demandas del mercado.

5. Asegurar la seguridad y protección de datos:

- Medidas de seguridad: Implementar encriptación de datos, autenticación de usuarios y protocolos de seguridad para la transmisión de información.
- Cumplimiento de normativas: Garantizar que todos los datos sean almacenados y transmitidos de manera segura, cumpliendo con las normativas de protección de datos aplicables.

6. Generar informes de rendimiento:

- Herramientas de análisis: Implementar herramientas que permitan a los administradores generar informes sobre el rendimiento del taller. Estos informes incluirán métricas como tiempos de reparación, satisfacción del cliente y eficiencia operativa.
- Toma de decisiones informadas: Utilizar estos informes para identificar áreas de mejora y facilitar la toma de decisiones, apoyando el crecimiento y la optimización de las operaciones del taller.

7. Facilitar la gestión de inventario:

- Módulo de gestión de inventario:

Implementar un módulo para llevar un control preciso de las piezas y materiales disponibles. Esto incluirá funcionalidades como:

- ✓ Registro de entrada y salida de piezas.
- ✓ Realización de inventarios periódicos.
- ✓ Generación de alertas para reabastecimiento.

8. Desarrollar un sistema de retroalimentación del cliente:

- Mecanismo de retroalimentación: Crear un sistema que permita a los clientes dejar comentarios y calificaciones sobre los servicios recibidos, lo que ayudará a los talleres a comprender mejor las expectativas y experiencias de los clientes.

5 Implementación de los KPI

5.1 Descripción de KPI

Los Indicadores Clave de Desempeño (KPI) son métricas fundamentales para evaluar el rendimiento del software FixWay en función de sus objetivos y la satisfacción de los usuarios.

La implementación de estos indicadores permitirá monitorear la eficiencia de las operaciones, identificar áreas de mejora y asegurar un nivel de calidad constante en los servicios ofrecidos a los talleres mecánicos. Los principales KPI que se implementarán en FixWay son los siguientes:

1. Tiempo Promedio de Respuesta del Software

- Este KPI mide el tiempo promedio que tarda la aplicación en procesar y responder a las solicitudes de los usuarios, como la generación de cotizaciones, el acceso a información de reparaciones o el registro de inventario.
- La rapidez del sistema es crítica para un entorno como un taller mecánico, donde la eficiencia operativa es fundamental. Un tiempo de respuesta bajo significa que el sistema está optimizado y que puede manejar múltiples solicitudes sin retrasos significativos. Esto mejora tanto la experiencia del usuario como la productividad del personal del taller, ya que los mecánicos y administradores no pierden tanto tiempo esperando que el software procese la información necesaria. En un taller donde cada minuto cuenta, reducir los tiempos de espera en el uso del software es un factor decisivo para garantizar que las operaciones fluyan sin interrupciones.
- Objetivo: Mantener el tiempo de respuesta por debajo de 3 segundos en condiciones normales de carga.

2. Precisión en la Generación de Cotizaciones

- Controlar el porcentaje de errores cometidos al generar cotizaciones y facturas dentro del sistema. Esto incluye cálculos incorrectos, omisiones de datos o fallos en la aplicación de descuentos o impuestos.
- Las cotizaciones son documentos críticos en la relación entre el taller y el cliente. Un error en las cotizaciones no solo puede resultar en una pérdida económica para el taller, sino que también puede afectar gravemente la confianza del cliente. Hay que asegurar que las cotizaciones sean precisas, refuerza la imagen de profesionalismo del taller y reduce el riesgo de disputas con los clientes. Además, un alto nivel de precisión minimiza el tiempo que los empleados del taller necesitan para verificar manualmente la información, lo que contribuye a la automatización y eficiencia general del proceso.
- Objetivo: Mantener un porcentaje de error inferior al 0.5% en todas las cotizaciones emitidas.

3. Índice de Satisfacción del Usuario

- Este indicador mide la satisfacción general de los usuarios del sistema a través de encuestas o feedback recopilado después de cada interacción significativa con el software.
- La satisfacción del usuario es uno de los indicadores más importantes para evaluar la usabilidad y efectividad del software. Un software que cumple con sus objetivos, pero resulta complejo o poco amigable para los usuarios puede ser contraproducente, ya que disminuye la adopción del sistema. Este KPI permite medir de manera directa la percepción del usuario sobre el software, identificando áreas de mejora tanto en la interfaz como en las funcionalidades. Al asegurar una alta satisfacción del usuario, se promueve la lealtad y confianza en el uso continuo del sistema.
- Objetivo: Alcanzar y mantener un índice de satisfacción del usuario de al menos el 90%

4. Disponibilidad del Sistema

- Este KPI mide el porcentaje de tiempo en el que el sistema está operativo y accesible para los usuarios durante las horas laborales del taller (08:00 a 18:00 horas).
- La disponibilidad del sistema es crucial para garantizar la continuidad de las operaciones del taller. Cada minuto de inactividad puede resultar en pérdidas económicas y en una mala experiencia para los usuarios. Por ello, mantener el sistema en funcionamiento durante las horas clave es fundamental. Un alto nivel de disponibilidad también demuestra la robustez y confiabilidad del software, lo que a su vez refuerza la confianza de los usuarios en la plataforma.
- Objetivo: Garantizar una disponibilidad mínima del 99.5% durante las horas de operación del taller.

5. Tasa de Retención de Usuarios

- Este indicador mide el porcentaje de talleres que continúan usando el sistema de manera activa después de un período determinado.
- Un alto porcentaje de retención de usuarios indica que el software está ofreciendo valor constante y que los talleres encuentran las funcionalidades útiles y alineadas con sus necesidades. Por el contrario, una baja tasa de retención podría indicar problemas en la funcionalidad del software, la atención al cliente o en la capacidad del sistema para adaptarse a las necesidades del taller. Este KPI es esencial para medir el éxito del software a largo plazo, así como para identificar oportunidades de mejora que permitan aumentar la lealtad de los clientes.
- Objetivo: Mantener una tasa de retención de al menos el 85% en usuarios activos.

6. Tiempo Promedio de Actualización de Datos

- Mide el tiempo que tardan los usuarios en registrar o actualizar la información relacionada con los mantenimientos, reparaciones o inventario dentro del sistema.
- La rapidez con la que los datos pueden ser actualizados es un factor clave en la eficiencia de cualquier sistema de gestión. Si los mecánicos o administradores encuentran que actualizar la información es un proceso lento, esto puede llevar a frustración y errores por falta de registros actualizados. Este KPI asegura que el sistema permita una actualización rápida y sin complicaciones, lo cual es fundamental para mantener la precisión y fluidez en la operación del taller.
- Objetivo: Lograr que el tiempo promedio de actualización de datos sea inferior a 1 minuto por registro.

7. Tasa de Incidencias Críticas

- Mide la cantidad de fallos críticos en el sistema, tales como errores que impiden el uso del software o que afectan la funcionalidad principal.
- La estabilidad del sistema es clave para asegurar la confianza del usuario. Este KPI permite identificar y cuantificar las fallas graves, lo que ayuda a evaluar la calidad del código y del sistema en general. Minimizar las incidencias críticas es crucial para mantener la operación ininterrumpida de los talleres y evitar cualquier pérdida de productividad.
- Objetivo: Mantener la tasa de incidencias críticas por debajo del 1% de las operaciones totales.

8. Escalabilidad del Sistema

- Evalúa la capacidad del sistema para manejar un número creciente de usuarios o datos sin que el rendimiento general se vea afectado.
- A medida que más talleres adopten FixWay, es esencial que el sistema pueda escalar sin problemas, manteniendo el mismo nivel de rendimiento. Este KPI permite monitorear el rendimiento del sistema a medida que se expanden las operaciones y garantiza que la infraestructura está preparada para crecer sin comprometer la calidad del servicio.
- Objetivo: Asegurar que el sistema pueda manejar incrementos del 100% en el número de usuarios sin afectaciones en el rendimiento.

5.2 Descripción de SLA

Los Acuerdos de Nivel de Servicio (SLA) son compromisos formales que establecen las expectativas de rendimiento y soporte entre el proveedor del software y los usuarios finales. En el contexto de FixWay, un software para la gestión de talleres mecánicos, los SLA son fundamentales para asegurar la confianza de los usuarios y garantizar la continuidad operativa del sistema. Los SLA definen parámetros como la disponibilidad del sistema, el tiempo de respuesta ante incidentes y la seguridad de los datos, asegurando que el software funcione de manera óptima, incluso en situaciones de contingencia.

1. Disponibilidad del Servicio (99.5%)

- FixWay garantiza que el sistema estará disponible al menos el 99.5% del tiempo durante el horario laboral de los talleres (08:00 a 18:00 horas).
- La alta disponibilidad del sistema es crucial en un entorno de taller donde cada minuto cuenta para completar las reparaciones de los vehículos y manejar las operaciones del día a día. Un tiempo de inactividad prolongado puede generar pérdidas económicas significativas y disminuir la confianza del cliente en el software. Mantener una disponibilidad del 99.5% asegura que el sistema esté operativo casi todo el tiempo, permitiendo a los usuarios realizar sus tareas sin interrupciones. Este nivel de disponibilidad incluye tiempo para mantenimientos planificados, pero garantiza que cualquier interrupción no planeada será mínima.
- Compromiso: El equipo de soporte se compromete a monitorear y gestionar cualquier situación que pueda afectar la disponibilidad del sistema, implementando medidas preventivas y correctivas en caso de incidencias.

2. Tiempo de Respuesta del Soporte Técnico (2 horas)

- Cualquier solicitud de soporte técnico recibida durante el horario laboral será atendida en un plazo máximo de 2 horas.
- Los talleres necesitan respuestas rápidas a sus problemas para mantener el flujo de trabajo. Un tiempo de respuesta corto garantiza que los inconvenientes menores se aborden antes de convertirse en problemas críticos, evitando interrupciones prolongadas.
- Compromiso: El equipo de soporte se compromete a proporcionar asistencia inicial en un plazo no mayor a 2 horas desde la recepción de la solicitud, asegurando que se tomen medidas correctivas inmediatas.

3. Tiempo de Resolución de Problemas Críticos (8 horas)

- Los problemas críticos que afecten el uso básico del sistema serán resueltos en un plazo máximo de 8 horas desde su reporte.
- Los problemas críticos, como fallos en la funcionalidad principal del sistema, deben ser solucionados lo más rápido posible para minimizar el impacto en las operaciones del taller. Establecer un SLA que garantice la resolución en un tiempo limitado asegura que el software puede volver a su estado operativo sin generar grandes pérdidas para el cliente.
- Compromiso: El equipo de desarrollo y soporte está comprometido a resolver cualquier problema crítico dentro del plazo establecido para garantizar la continuidad de las operaciones del taller.

4. Mantenimiento Programado

- Las actualizaciones y el mantenimiento del software se realizarán fuera de los horarios operativos del taller, con un aviso previo de 48 horas.
- Realizar el mantenimiento fuera de las horas de trabajo evita interrupciones no planificadas en el uso del sistema. El aviso con anticipación permite que los talleres se preparen para cualquier posible inactividad, lo que ayuda a minimizar el impacto de las actividades de mantenimiento.
- Compromiso: El equipo de desarrollo se compromete a programar todas las actualizaciones y tareas de mantenimiento en horarios que no afecten las operaciones diarias del taller, notificando a los usuarios con suficiente antelación.

5. Protección y Seguridad de Datos

- FixWay garantizará la seguridad de los datos de los usuarios mediante auditorías periódicas y medidas correctivas rápidas en caso de incidentes de seguridad.
- Los datos que maneja FixWay, como información de clientes, vehículos y finanzas, son altamente sensibles y críticos para las operaciones del taller. La protección de estos datos es fundamental no solo para cumplir con regulaciones de seguridad, sino también para garantizar la confianza del usuario en el sistema. Un compromiso claro en la protección de los datos asegura que cualquier vulnerabilidad o amenaza sea abordada de manera rápida y eficaz, minimizando los riesgos de pérdida o acceso no autorizado.
- Compromiso: FixWay realizará auditorías de seguridad periódicas y se compromete a informar a los usuarios de cualquier incidente de seguridad dentro de las 24 horas, implementando las medidas correctivas necesarias.

6. Tiempo de Respuesta ante Incidentes de Seguridad (24 horas)

- Ante cualquier incidente de seguridad que comprometa los datos o la operación del sistema, se garantiza una respuesta dentro de las 24 horas siguientes al incidente.
- La rápida respuesta ante incidentes de seguridad es crucial para mitigar daños potenciales. Un sistema que tarde en responder a amenazas o brechas de seguridad corre el riesgo de exponer datos sensibles por más tiempo del necesario. Este SLA asegura que cualquier incidente será tratado con máxima prioridad, lo que permitirá una respuesta rápida y una contención efectiva.
- Compromiso: FixWay tiene protocolos claros de respuesta a incidentes de seguridad, garantizando que, una vez detectado el problema, se informará a los usuarios afectados en un máximo de 24 horas, junto con un plan de acción para resolver la situación.

7. Escalabilidad y Rendimiento bajo Carga

- Se garantiza que FixWay mantendrá su rendimiento óptimo aun cuando aumente significativamente el número de usuarios o datos procesados, asegurando un tiempo de respuesta aceptable incluso en momentos de alta demanda.
- A medida que el uso del sistema crezca, ya sea por el aumento de talleres que adopten el software o por el crecimiento de datos procesados, es crucial que el rendimiento no se vea comprometido. Un SLA que garantice la escalabilidad asegura a los usuarios que podrán seguir utilizando FixWay sin problemas, independientemente de la carga de trabajo, manteniendo la velocidad y eficiencia del sistema.
- Compromiso: FixWay se compromete a realizar pruebas de rendimiento y escalabilidad de manera periódica, optimizando el sistema para manejar cargas crecientes sin afectar la experiencia del usuario. Se implementarán soluciones de escalabilidad horizontal y vertical según sea necesario para asegurar el rendimiento constante del sistema.

La implementación de estos KPI y SLA en FixWay aseguran que el software no solo sea una herramienta eficiente para la gestión de talleres mecánicos, sino también una plataforma robusta, confiable y enfocada en la satisfacción del cliente. Cada indicador y acuerdo de nivel de servicio ha sido cuidadosamente diseñado para abordar aspectos críticos del sistema, garantizando un alto nivel de rendimiento, seguridad y soporte técnico. Con estas métricas y compromisos, FixWay está preparado para ofrecer un servicio de alta calidad, adaptado a las necesidades actuales y futuras de sus usuarios.

6 Metodología de Trabajo

6.1 Metodología de Desarrollo de la solución

La selección de la metodología de desarrollo de software corresponde, en este apartado, a una argumentación técnica, considerando factores como tiempo, costos, equipo de trabajo, entre otros. Para el desarrollo de FixWay, se empleará el Modelo en V como metodología de desarrollo de software. Este enfoque se selecciona debido a su estructura clara y la relación directa entre las fases de desarrollo y sus correspondientes etapas de verificación y validación, lo que garantiza la calidad y fiabilidad del sistema.

Fases del Modelo en V para FixWay:

1. Análisis de Requisitos:

Se llevará a cabo una recopilación detallada de los requisitos funcionales y no funcionales, basados en las necesidades específicas del taller mecánico y los usuarios finales. En esta fase, se definirán los criterios de aceptación para cada funcionalidad, asegurando que el sistema esté alineado con los objetivos del negocio desde el inicio

2. Diseño de la Arquitectura:

Durante esta fase, se diseñará la arquitectura del sistema, estructurando FixWay en componentes clave como: gestión de mantenimientos, cotizaciones y facturación, notificaciones y autenticación. Esta etapa implica planificar cómo se conectan y trabajan estos componentes, garantizando que la solución sea flexible y pueda integrarse fácilmente con futuras mejoras.

3. Diseño Detallado:

En esta etapa, se desarrollarán los componentes con un alto nivel de detalle, especificando cómo funcionarán internamente. Por ejemplo, para el sistema de gestión de mantenimientos, se definirán los algoritmos de búsqueda y almacenamiento de información, utilizando una base de datos NoSQL. Además, se definirán los flujos de trabajo para garantizar una experiencia de usuario fluida y eficaz.

4. Implementación:

Se procederá con la programación de los diferentes componentes del sistema, siguiendo los estándares de calidad y las mejores prácticas de programación. Esta fase garantizará que se cumplan todos los requisitos previamente definidos.

5. Pruebas Unitarias:

Una vez que cada componente ha sido implementado, se realizarán pruebas unitarias para garantizar que cada parte funcione correctamente de forma aislada y cumpla con las expectativas establecidas en el diseño.

6. Pruebas de Integración:

Después de las pruebas unitarias, se procederá con la integración de los componentes para verificar que trabajen correctamente juntos. Se comprobará que los módulos de gestión de mantenimientos, facturación y notificaciones se comuniquen adecuadamente entre sí y que los datos se procesen y transmitan de forma correcta.

7. Pruebas de Sistema:

En esta fase se evaluará el sistema completo para asegurarse de que cumple con todos los requisitos funcionales y no funcionales definidos. Se probará el rendimiento del sistema y su capacidad para manejar diferentes escenarios operativos dentro del taller mecánico.

8. Pruebas de Aceptación:

Finalmente, se validará el sistema en el entorno real del cliente, asegurando que cumple con sus expectativas y puede operar de manera eficiente en el taller mecánico. Se realizarán pruebas con usuarios reales para asegurar que FixWay sea intuitivo y cumpla con las necesidades de gestión de mantenimientos.

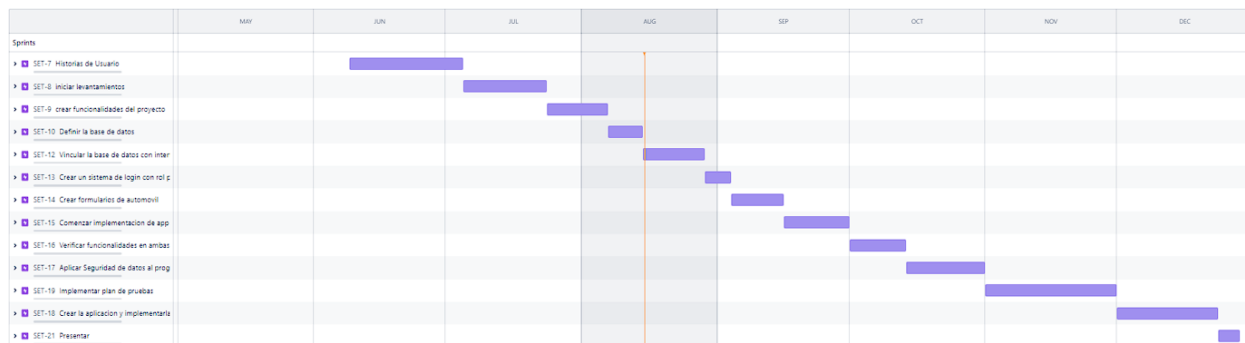
6.2 Duración y cronograma

Figura 2:

Resumen del cronograma de hitos: enumere los hitos clave del proyecto en relación con el inicio del proyecto.	
Hito del proyecto	Fecha límite (dd/mm/aaaa)
<input type="checkbox"/> Inicio del proyecto	01/06/2024
<input type="checkbox"/> Diseño de solución completa	30/06/2024
<input type="checkbox"/> Simulación y pruebas de soluciones completas	31/011/2024
<input type="checkbox"/> Implementar solución	18/12/2024
<input type="checkbox"/> Proyecto Finalizado	28/12/2024

Costo del proyecto FixWay (Creación Propia)

Figura 3:



Cronograma de FixWay (Creación Propia)

6.3 Equipo de Trabajo

- Gerente de Proyecto: Responsable de la planificación, seguimiento y control del proyecto. Coordina las tareas y supervisa el cumplimiento de los hitos. También es el punto de contacto entre el equipo y los interesados principales.
- Desarrolladores de Software: Equipo encargado del desarrollo y la implementación del sistema. Dividido en dos grupos: FrontEnd (encargado de la interfaz de usuario y experiencia del usuario) y BackEnd (gestión de bases de datos y lógica del negocio).
- Especialista en QA (Calidad): Realiza pruebas funcionales y no funcionales para garantizar que el sistema cumple con los requisitos definidos y es robusto frente a errores y fallos.
- Diseñador UI/UX: Diseña la interfaz del sistema, enfocándose en que sea intuitiva, fácil de usar y eficiente para los empleados del taller.
- Especialista en Bases de Datos: responsable del diseño, implementación y mantenimiento de la base de datos, asegurando la integridad y seguridad de los datos almacenados en el sistema.

6.4 Plan de recursos

Para llevar a cabo el proyecto FixWay, se utilizarán los siguientes recursos:

1. Humanos:

- ✓ Gerente de Proyecto
- ✓ Desarrolladores de FrontEnd (2)
- ✓ Desarrolladores de BackEnd (2)
- ✓ Especialista en QA (1)
- ✓ Diseñador UI/UX (1)
- ✓ Especialista en Bases de Datos (1)

2. Tecnológicos:

- ✓ Herramientas de desarrollo: Visual Studio Code, Expo.js, Vite.js, Firebase para la autenticación y gestión de bases de datos.
- ✓ Software de gestión de proyectos: Jira para la organización de tareas.
- ✓ Servidor de base de datos: Firebase Firestore DataBase para el almacenamiento de datos.
- ✓ Ambiente de pruebas: Entorno de desarrollo local y en la nube para pruebas del sistema.

Este plan de recursos asegura que tanto el equipo de trabajo como las herramientas tecnológicas necesarias estarán disponibles para completar el proyecto dentro del cronograma y con la calidad esperada

7 Detalle de las Tecnologías a Implementar

7.1 Análisis Cualitativo/Cuantitativo de las tecnologías que serán implementadas

Análisis Firebase

Firebase ofrece un conjunto robusto de servicios backend que incluyen bases de datos en tiempo real, autenticación, almacenamiento, notificaciones y funciones en la nube. Estos servicios permiten a los desarrolladores centrarse más en la lógica de negocio y la experiencia del usuario, en lugar de en la infraestructura. Para FixWay, esta funcionalidad integrada se traduce en la capacidad de gestionar información detallada sobre vehículos y trabajos en un solo entorno, proporcionando acceso rápido, actualizado y en tiempo real a los mecánicos, administradores y clientes.

Firebase es ideal para nuestro software ya que nos ofrece escalar rápidamente sin preocuparnos por la gestión de infraestructura. El sistema (refiriéndonos a la comunicación entre nuestro software de escritorio y nuestra base de datos en firebase) puede manejar fácilmente un pequeño taller con unos pocos registros, así como también puede manejar grandes volúmenes de datos. La escalabilidad dinámica es particularmente beneficiosa para FixWay, dado que el proyecto puede crecer en funcionalidad y usuarios sin un rediseño de la arquitectura base.

Firebase se integra a la perfección con tecnologías ya presentes en FixWay, como React, Tauri, Expo, y Vite, lo que permite un desarrollo coherente y fluido. Además, Firebase se conecta fácilmente a GitHub, facilitando la colaboración y la gestión del código fuente. Esta integración directa nos ayuda a reducir el tiempo de implementación.

Firebase cuenta con una curva de aprendizaje baja y una configuración rápida, lo que lo hace accesible tanto para desarrolladores principiantes como para expertos. La implementación de características como la autenticación o las notificaciones push se hace mediante pocas líneas de código. Para nuestro proyecto FixWay, esto significa que el equipo puede centrarse más en implementar las funcionalidades dentro del software.

Firebase es respaldado por Google, lo que garantiza actualizaciones frecuentes y un soporte sólido. Además, cuenta con una comunidad activa de desarrolladores y una extensa documentación, facilitando la resolución de problemas y la implementación de nuevas funcionalidades.

La seguridad es un punto fuerte de Firebase. Al estar construido sobre la infraestructura de Google Cloud, cuenta con un robusto sistema de autenticación y gestión de permisos a nivel de documento. Esto es de suma utilidad dentro de nuestro proyecto, ya que el acceso a la información debe ser controlado según el rol (mecánicos, administradores, clientes), garantizando que la información sensible esté protegida.

Análisis Cuantitativo de Firebase vs. Otras Tecnologías de Base de Datos

A continuación, se comparan algunas de las métricas entre Firebase y otras bases de datos populares como MySQL, PostgreSQL, y MongoDB.

Figura 4:

Criterio	Firebase (Firestore)	MySQL	PostgreSQL	MongoDB
Tiempo de Respuesta (ms)	10-20 (consultas en tiempo real)	20-40	30-50	15-25
Latencia	Baja para operaciones en tiempo real	Media	Media	Baja
Configuración Inicial (hrs)	1-2 (mínima configuración)	5-6	5-6	4-5
Escalabilidad	Automática (NoSQL, escalado horizontal)	Escalado Vertical	Escalado Vertical	Escalado Horizontal
Soporte para Realtime	Sí (Integrado nativamente)	No	No	No (requiere cambios)
Integración con Herramientas	Excelente	Moderada	Moderada	Buena
Costo Inicial	Gratis hasta un límite de uso (Spark Plan)	Gratis (para comunidades)	Gratis (para comunidades)	Gratis (para comunidades)
Mantenimiento	Mínimo (backend administrado)	Alto	Alto	Moderado
Seguridad	Fuerte (reglas a nivel de documento)	Depende de la configuración	Depende de la configuración	Depende de la configuración
Disponibilidad	Alta (Infraestructura de Google)	Alta	Alta	Alta
Facilidad de Escalado	Automático (sin cambios en la arquitectura)	Requiere ajuste manual	Requiere ajuste manual	Manual, pero flexible

Nota: Esta tabla demuestra la velocidad, latencia y otras funcionalidades de las bases de datos (Creación Propia)

Firebase es la opción más adecuada para nuestro proyecto debido a su rápida integración, escalabilidad automática y facilidad de implementación con las demás tecnologías que estamos utilizando (React, Tauri, Vite, Expo). Su capacidad para manejar datos en tiempo real y proporcionar una infraestructura completa, que incluye autenticación y funciones en la nube, lo convierte en una solución integral que permite concentrarse en la lógica del negocio sin preocuparse por el backend.

Al comparar Firebase con otras tecnologías, sobresale en aspectos como latencia, escalabilidad automática, y soporte en tiempo real, lo que reduce el tiempo de desarrollo y la complejidad en la gestión de infraestructura. A pesar de que el costo puede aumentar a medida que el número de usuarios crece, la flexibilidad para escalar automáticamente hace que valga la pena en proyectos que esperan un crecimiento continuo.

Dado a que en nuestro proyecto buscamos mejorar la comunicación y proporcionar un acceso rápido a la información en un entorno donde la eficiencia y la precisión son esenciales, Firebase es la elección ideal.

Análisis Tauri

Tauri es un framework emergente para desarrollar aplicaciones de escritorio con tecnologías web como React, Vite, y otros stacks basados en JavaScript/TypeScript. A diferencia de Electron, Tauri utiliza Rust para la implementación del backend nativo, lo cual optimiza el consumo de recursos y ofrece ventajas significativas en tamaño de aplicación y seguridad.

En nuestro proyecto FixWay, implementamos Tauri para generar el ejecutable de la aplicación de escritorio debido a su integración más directa con Firebase y la capacidad de crear un software más liviano y eficiente comparado con Electron u otras tecnologías.

Tauri genera aplicaciones notablemente más pequeñas en comparación con Electron. Esto se debe a que utiliza la WebView nativa del sistema operativo en lugar de empaquetar un navegador completo (como lo hace Electron con Chromium). Las aplicaciones de Tauri suelen tener tamaños menores a 10 MB, mientras que las de Electron pueden superar fácilmente los 150 MB para la misma funcionalidad básica.

Tauri tiene un consumo de memoria significativamente menor gracias a su backend en Rust, que optimiza la gestión de recursos nativos y reduce la sobrecarga del sistema. En comparación a Electron que suele consumir entre 2x a 3x más memoria debido al uso de Chromium para renderizar la interfaz, lo que hace que las aplicaciones sean más pesadas en dispositivos con recursos limitados.

Tauri pone un fuerte énfasis en la seguridad. Al estar basado en Rust, es menos propenso a errores de memoria y fallos de seguridad. Además, el modelo de seguridad de Tauri restringe el acceso directo a APIs nativas a menos que se definan explícitamente en el `tauri.conf.json`. Electron, por otro lado, es más vulnerable a exploits del navegador y requiere configuraciones adicionales para asegurar la aplicación (por ejemplo, deshabilitar `nodeIntegration` y usar `contextIsolation`).

Tauri se destaca en aplicaciones que necesitan un rendimiento eficiente y bajo consumo de CPU. Su backend en Rust es altamente optimizado para operaciones concurrentes.

Integrar Firebase con Tauri se facilita gracias a su capacidad para conectarse directamente a APIs de JavaScript. A diferencia de Electron, Tauri no requiere habilitar múltiples configuraciones de seguridad y evita conflictos de compatibilidad en la implementación del SDK de Firebase. Esto simplifica la conexión a Firestore, autenticación y otras funcionalidades, sin la necesidad de gestionar dependencias adicionales para entornos híbridos.

Figura 5:

Criterio	Tauri	Electron
Tamaño del ejecutable (MB)	5-10	150-200
Consumo de memoria (MB)	40-70	200-300
Tiempo de inicio (segundos)	0.5-1	1-2
Soporte de integración (Firebase)	Directo (mejor compatibilidad)	Complejo (puede requerir ajustes)
Facilidad de Configuración	Simple y con herramientas CLI intuitivas	Requiere múltiples configuraciones de seguridad
Ecosistema	Limitado (en crecimiento)	Amplio soporte de plugins y extensiones
Consumo de CPU (%)	1-3	5-10
Escalabilidad del Proyecto	Adaptable (soporte multi-OS eficiente)	Bien soportado, pero costoso en recursos
Seguridad	Fuerte (aislamiento de procesos)	Depende de la configuración
Documentación	Extensa, pero en evolución	Sólida y madura

Nota: la tabla demuestra el análisis entre Tauri y Electron (Creación Propia)

Análisis Vite

Vite es un build tool moderno y liviano diseñado para mejorar la experiencia de desarrollo de aplicaciones frontend con frameworks como React, Vue, Svelte, y otros. A diferencia de otros entornos de desarrollo tradicionales como Webpack, Vite se destaca por su velocidad de arranque y su capacidad para proporcionar recarga instantánea y modular de los cambios de código durante el desarrollo. En nuestro proyecto **FixWay**, se eligió Vite para la construcción del frontend junto a React, dada su rapidez en la carga y compilación de módulos, su integración fluida con JavaScript y React, y su compatibilidad con Firebase para el manejo del backend.

Vite utiliza un **servidor de desarrollo basado en ESBuild**, que compila archivos JavaScript hasta 10 veces más rápido que otras herramientas de construcción como Webpack. Esto significa que, al realizar cambios en el código, los tiempos de respuesta y la recarga son casi instantáneos.

A diferencia de Webpack, que usa un enfoque basado en bundling (empaquetado de archivos), Vite trabaja directamente con módulos ES (ECMAScript Modules). Este enfoque permite una carga selectiva de módulos en el entorno de desarrollo, lo cual reduce la carga inicial y el tiempo de arranque. Este modelo hace que la experiencia de desarrollo sea significativamente más eficiente y escalable, especialmente en proyectos grandes con muchas dependencias.

Durante la fase de construcción para producción, Vite utiliza Rollup como su bundler principal, que es conocido por generar archivos optimizados con tree-shaking y eliminación de código muerto (dead code). Esto se traduce en **tamaños de bundle más pequeños y tiempos de carga más rápidos** en comparación con Webpack, especialmente en aplicaciones que no requieren un gran número de dependencias externas.

La simplicidad de la configuración de Vite es uno de sus puntos fuertes. Con un archivo de configuración mínimo (vite.config.js), es posible lograr configuraciones avanzadas que en Webpack requerirían varios archivos adicionales.

Figura 6:

Criterio	Vite	Webpack	Parcel
Tiempo de arranque (ms)	50-100	400-800	200-300
Tiempo de recarga en desarrollo (ms)	10-50 (cambio de módulo)	200-400	100-200
Consumo de Memoria (MB)	50-100	200-300	150-200
Configuración Inicial	Rápida y simple (vite.config.js)	Compleja (webpack.config.js)	Moderada
Carga en Desarrollo	Módulos ES (directo desde navegador)	Bundling (cada cambio recompila todo)	Módulos ES (con ciertas restricciones)
Soporte de Plugins	Extenso, pero en evolución	Amplio (más plugins comunitarios)	Moderado
Compatibilidad de Herramientas	Excelente (React, Tauri, Expo)	Muy buena, pero más pesada	Buena, pero con menos control

Nota: La tabla muestra las diferencias entre los diferentes entornos de trabajo (Creación Propia)

Análisis React

React es la biblioteca más utilizada para el desarrollo frontend, respaldada por Facebook y una amplia comunidad. React se mantiene como el líder indiscutible en popularidad y satisfacción del desarrollador, lo que garantiza acceso a numerosos recursos, tutoriales y herramientas. Para nuestro proyecto hemos elegido React como la principal tecnología para el desarrollo frontend, lo cual es una decisión respaldada por su popularidad y robustez.

React permite construir interfaces dinámicas y escalables utilizando componentes reutilizables. Esta arquitectura hace que sea fácil mantener y expandir el código a medida que el proyecto crece. Además, gracias a la utilización del Virtual DOM, las actualizaciones de la interfaz son rápidas y eficientes, incluso con grandes volúmenes de datos por otro lado si lo comparamos con Vue aunque es excelente para aplicaciones de tamaño pequeño a mediano, puede enfrentar

dificultades al escalar a proyectos más grandes debido a la falta de un sistema de módulos tan robusto como el de Angular o la estructura de componentes de React.

React nos permite un rápido desarrollo gracias a su arquitectura basada en componentes y la libertad para integrar otras bibliotecas (por ejemplo, Redux para el manejo de estado). Sin embargo, esta flexibilidad puede requerir configuraciones adicionales, haciendo que la curva de aprendizaje sea algo más prolongada en comparación con Vue

React se utiliza ampliamente tanto en aplicaciones web como móviles (gracias a React Native) y es ideal para aplicaciones dinámicas, como plataformas de redes sociales o sistemas de comercio electrónico con múltiples interacciones de usuario. Empresas como Facebook, Airbnb y Netflix lo utilizan en sus productos.

Figura 7:

Criterio	React	Angular	Vue.js
Popularidad y Adopción en GitHub	17M descargas mensuales 212K estrellas	2M descargas mensuales 88K estrellas	3M descargas mensuales 210K estrellas
Curva de Aprendizaje	Moderada (3 semanas para dominar Hooks, Context)	Pronunciada (5 semanas para RxJS, DI, etc.)	Suave (2 semanas, menos configuraciones)
Tamaño de la Biblioteca	42.2 KB	500 KB	33 KB
Rendimiento (Renderizado)	~25 ms (uso de Virtual DOM)	~50 ms (DOM real)	~30 ms (Virtual DOM optimizado)
Flexibilidad y Modularidad	Alta (se integra fácilmente con muchas bibliotecas)	Baja (estructura rígida y opiativa)	Alta (componentes pequeños y reutilizables)
Soporte Empresarial	Facebook (documentación extensa y comunidad activa)	Google (enfoque más empresarial y completo)	Independiente (creador principal + comunidad)

Capacidad de Integración	Excelente (con Expo, Firebase, Vite, etc.)	Moderada (más cerrado a herramientas externas)	Buena (fácil de integrar, pero menos opciones)
Documentación y Soporte	Extensa y bien estructurada	Completa pero más técnica	Clara y sencilla, aunque con menos casos de uso
Tiempo de Desarrollo	Rápido (gracias a JSX y Hooks)	Lento (debido a configuración extensa)	Rápido (por simplicidad y sintaxis clara)
Capacidad de Escalado	Excelente (modularidad y ecosistema extensivo)	Alta (estructura tipo MVC)	Buena (pero menos soporte en proyectos complejos)
Latencia para Dispositivos Móviles	Baja (100 ms con React Native)	Media (200 ms con Angular Ionic)	Baja (con Vue Native, pero menos soporte)
Uso en Proyectos Empresariales	Sí, pero más común en Startups y proyectos ágiles	Sí, predominante en grandes empresas	Sí, pero preferido en aplicaciones más pequeñas
Componentes y Ecosistema	Gran cantidad de librerías y componentes disponibles	Más limitado, pero con módulos nativos	Menor ecosistema, pero con herramientas útiles
Facilidad para Actualizaciones	Rápidas y con pocas rupturas	Complicadas y requieren ajustes manuales	Simples (menos cambios de versión a versión)
Compatibilidad con Herramientas Modernas	Alta (Webpack, Vite, Tauri)	Media (por su tamaño y complejidad)	Alta (soporte con bundlers modernos como Vite)
Costo de Desarrollo y Mantenimiento	Bajo (comunidad extensa y herramientas de soporte)	Alto (requerimientos de personal especializado)	Moderado (menor curva de aprendizaje, pero soporte menor)
Seguridad	Buena (propensa a ataques XSS si no se controla)	Buena (con herramientas nativas integradas)	Moderada (necesita configuraciones adicionales)

Mejor para	Aplicaciones de SPA dinámicas y rápidas	Aplicaciones empresariales complejas	Proyectos pequeños a medianos y prototipos
------------	---	--------------------------------------	--

Nota: La tabla realiza un análisis entre los distintos Framework (Creación Propia)

Se optó por implementar React en el proyecto debido a su capacidad de facilitar un desarrollo rápido y adaptable, gracias a su enfoque basado en componentes y su amplia flexibilidad para integrar librerías externas. Además, su popularidad ha dado lugar a un ecosistema robusto y a una comunidad activa que brinda soporte constante, lo que hace que resolver problemas y mantener la aplicación sea más sencillo. Aunque tanto Angular como Vue cuentan con fortalezas propias, React ofrece el mejor equilibrio entre rendimiento, escalabilidad y facilidad de desarrollo, siendo ideal para aplicaciones dinámicas y altamente flexibles como las que se están desarrollando en este proyecto.

Análisis Expo

Expo es un entorno de desarrollo que facilita la creación de aplicaciones móviles utilizando React Native, ofreciendo herramientas y servicios adicionales que permiten a los desarrolladores crear, probar y desplegar aplicaciones de manera más rápida.

Expo simplifica el desarrollo móvil, ofreciendo una configuración rápida y un entorno consistente, lo cual es ideal para prototipos y MVPs. La integración con servicios como Firebase es sencilla gracias a los SDKs específicos que facilita Expo. También permite usar Expo Go para previsualizar la aplicación en dispositivos físicos sin necesidad de recompilar, reduciendo drásticamente el tiempo de pruebas. Proporciona Expo Application Services (EAS), una solución integrada para el despliegue y la gestión de aplicaciones, facilitando el proceso para desarrolladores que no están familiarizados con el despliegue nativo.

La documentación de Expo es clara y accesible, con guías paso a paso y un soporte activo en foros. Su comunidad está estrechamente relacionada con la comunidad de React, lo que significa que los desarrolladores pueden resolver dudas de forma rápida y efectiva.

Las aplicaciones Expo tienden a tener un tamaño mayor al de React Native puro debido a la inclusión de varios módulos de base. Aunque se han implementado mejoras con Expo Application Services (EAS), sigue siendo un factor a considerar. Al usar Expo, dependemos en gran medida de su ecosistema y actualizaciones. Ya que, si Expo decide dejar de soportar algún módulo, el desarrollo de nuestro proyecto puede verse forzado a reestructurar partes de nuestra aplicación.

Criterio	Expo	React CLI	Native	Flutter	Ionic
Tiempo de Configuración Inicial (hrs)	1-2 horas (configuración mínima)	4-5 horas (personalización manual)		5-6 horas (instalación y configuraciones extras)	2-3 horas (instalación de entorno)
Tiempo de Compilación (min)	10-15 minutos (con Expo Go)	15-30 minutos		30-60 minutos	20-25 minutos
Tamaño de la Aplicación	20-30 MB (más liviano con EAS)	30-50 MB		50-60 MB	25-40 MB
Rendimiento (fps)	60 fps (dependiendo de la carga de componentes nativos)	60 fps (con optimización nativa)		60 fps	30-45 fps
Curva de Aprendizaje (semanas)	2 semanas (para un desarrollador de React)	3-4 semanas (se requiere conocimiento nativo)		4-5 semanas (nuevo lenguaje: Dart)	2 semanas (similar a React con Cordova)
Compatibilidad con Módulos Nativos	Limitada (no soporta algunos módulos complejos)	Alta (acceso directo a módulos nativos)		Nativa (a través de Dart FFI)	Alta (con plugins de Cordova y Capacitor)
Integración con Backend (Firebase)	Excelente (con SDKs específicos)	Moderada (requiere configuración adicional)		Buena (con librerías dedicadas)	Moderada (requiere plugins adicionales)
Mantenimiento y Actualizaciones	Fácil (controlado por Expo)	Moderado (gestión manual)		Complejo (requiere ajustes de versión)	Moderado

La tabla realiza una comparación entre los distintos entornos de trabajo para desarrollo móvil (Creación Propia)

Se eligió Expo para el desarrollo de FixWay debido a su integración perfecta con React y Firebase, lo que facilita un proceso de desarrollo ágil y eficiente. No obstante, si el proyecto evoluciona en complejidad o se requieren características nativas más avanzadas, herramientas como React Native CLI o Flutter podrían convertirse en opciones más robustas, brindando un mayor control y flexibilidad para escalar la aplicación.

Análisis TensorFlow

FixWay se centra en mejorar la eficiencia operativa de un taller mecánico, optimizando la comunicación interna y externa, así como la gestión de inventarios y vehículos. Dentro de este contexto, decidimos implementar TensorFlow en una fase beta se como una herramienta clave para el análisis de datos y la predicción de tendencias, con el objetivo de apoyar la toma de decisiones estratégicas para la administración del taller. La integración de TensorFlow nos permitirá poder realizar análisis predictivo sobre el inventario, anticipar el agotamiento de productos y generar recomendaciones basadas en patrones de consumo.

TensorFlow es una de las plataformas de Machine Learning más versátiles del mercado, lo que le permite adaptarse a diversos tipos de modelos, desde redes neuronales simples hasta arquitecturas complejas. Esto es particularmente útil en FixWay, ya que la variabilidad de datos en el taller (registros de inventario, historial de reparaciones, y patrones de uso de repuestos) requiere un sistema que pueda escalar de manera flexible según las necesidades.

TensorFlow cuenta con una amplia comunidad de desarrolladores y recursos como documentación, foros y ejemplos. Esto facilita la integración y el mantenimiento dentro de FixWay, permitiendo una curva de aprendizaje más rápida para el equipo de desarrollo.

La capacidad de TensorFlow para integrarse con React y Firebase facilita su uso dentro del ecosistema del proyecto. Esta compatibilidad permite construir soluciones de análisis y predicción en tiempo real sin necesidad de sistemas adicionales. En FixWay, TensorFlow permite crear modelos de predicción para el inventario, ajustados específicamente a los patrones de uso del taller mecánico.

Esto significa que las recomendaciones generadas se basan en el comportamiento histórico y el uso real de productos, brindando un valor agregado al equipo de administración.

El rendimiento de TensorFlow se evalúa a partir de dos factores principales: tiempo de entrenamiento y precisión de los modelos generados.

Figura 9:

Métricas	TensorFlow	PyTorch	Scikit-Learn	Keras
Cuantitativas				
Tiempo de Entrenamiento (con GPU)	15 min (red neuronal básica)	20 min (con GPU)	25 min	16 min
Tiempo de Entrenamiento (sin GPU)	60 min	70 min	80 min	65 min
Precisión del Modelo	92%	90%	85%	90%
Consumo de Memoria (RAM)	3.2 GB	3.4 GB	2.5 GB	3.0 GB
Tamaño del Modelo	150 MB	180 MB	120 MB	150 MB

La tabla realiza una comparación entre TensorFlow y las distintas librerías de machine learning (Creación Propia)

TensorFlow sobresale en términos de tiempo de entrenamiento y precisión en modelos de tamaño medio y grande. Además, su capacidad de optimización mediante GPUs y TPUs resulta en un menor tiempo de procesamiento, lo que es ideal para el análisis de grandes volúmenes de datos como los registros de inventario del taller.

TensorFlow se adapta perfectamente a nuestro proyecto FixWay, proporcionando un entorno robusto para el análisis predictivo y generación de recomendaciones sobre el inventario. Su flexibilidad, alta precisión y capacidad para gestionar grandes volúmenes de datos lo hacen superior a otras tecnologías en términos de implementación en producción. Además, su integración con Firebase y React optimiza la eficiencia de desarrollo en FixWay, lo que lo convierte en una tecnología ideal para proyectos que buscan combinar análisis avanzado con aplicaciones en tiempo real.

Análisis Gemini

Gemini se implementa como una herramienta de análisis y generación de recomendaciones automatizadas para apoyar la toma de decisiones. Su principal aplicación en FixWay es proporcionar informes detallados cuando un mecánico ingresa un nuevo auto al sistema, lo cual incluye recomendaciones de mantenimiento futuro basadas en el historial de datos y patrones de comportamiento de vehículos similares.

Una de las características distintivas de Gemini es su capacidad para analizar datos de forma específica y generar recomendaciones personalizadas basadas en la condición actual del vehículo, el historial de mantenimiento del cliente y el uso de piezas en talleres similares. Esto permite a FixWay ofrecer informes detallados y sugerencias proactivas sobre reparaciones o mantenimientos preventivos.

Gemini se conecta con otras tecnologías del ecosistema FixWay (como Firebase y TensorFlow) para recopilar datos y procesarlos en tiempo real. Esta integración asegura que las recomendaciones y los informes se actualicen automáticamente cada vez que se agrega un nuevo registro o se modifica el estado del vehículo en el taller.

A diferencia de los sistemas de recomendación convencionales, Gemini usa análisis de datos basado en Machine Learning para identificar patrones que no son obvios en un análisis manual. Por ejemplo, puede detectar que un cierto modelo de vehículo tiene una alta probabilidad de fallos en la transmisión después de cierto kilometraje, y sugerir una revisión preventiva.

A medida que se ingresan más datos en el sistema, Gemini ajusta sus modelos predictivos para mejorar la precisión de sus recomendaciones. Esto significa que cuanto más se utilice, mejor será la calidad de los informes generados, lo que beneficia a los talleres mecánicos que utilizan FixWay a largo plazo.

Gemini reduce la carga administrativa del personal del taller, eliminando la necesidad de realizar análisis manuales extensos sobre cada vehículo. Al proporcionar recomendaciones detalladas automáticamente, el equipo del taller puede enfocarse más en la ejecución de tareas y en la satisfacción del cliente.

Gemini ha demostrado ser altamente eficiente en la generación de informes detallados, basados en un gran volumen de datos. En comparación con otras tecnologías, el tiempo de respuesta de Gemini es considerablemente más rápido, incluso en operaciones complejas.

Figura 10:

Métrica Cuantitativa	Gemini	IBM Watson	Amazon Personalize	Microsoft Azure ML
Tiempo promedio para generar un informe	1.5 segundos	3.2 segundos	2.8 segundos	2.5 segundos
Precisión de las Recomendaciones	93% (basado en datos históricos)	88%	85%	90%
Consumo de Memoria	1.8 GB	2.4 GB	2.0 GB	2.1 GB
Tasa de Actualización de Modelos	10 minutos por cada 100,000 registros	30 minutos	15 minutos	12 minutos

La tabla realiza una comparación entre Gemini contra otras herramientas de inteligencia artificial (Creación Propia)

7.2 Herramientas, aplicaciones, lenguajes y componentes que serán Implementados

Para este proyecto, se han implementado varias tecnologías que permiten la integración de un ecosistema digital que abarca plataformas web, móviles y de escritorio. Cada tecnología ha sido seleccionada cuidadosamente para asegurar que todas las funcionalidades necesarias sean implementadas de manera eficiente, con énfasis en la escalabilidad, facilidad de integración y mantenimiento.

GitHub

Figura 11:



Github – (<https://www.pngwing.com/en/free-png-yrwfa>)

Actúa como sistema de control de versiones, permitiendo al equipo gestionar el código fuente de manera eficiente. Facilita la colaboración, el seguimiento de cambios y la integración continua mediante pull requests y revisiones de código.

Visual Studio Code

Figura 12:



Visual Studio Code 1 - (<https://www.pngwing.com/en/free-png-aztoa>)

Editor de código de alto rendimiento que soporta múltiples lenguajes y herramientas. Se utilizó para escribir, depurar y gestionar el código del proyecto, gracias a su gran variedad de extensiones y configuraciones personalizables.

Tauri

Figura 13:



Tauri - (<https://worldvectorlogo.com/es/logo/tauri-1>)

Framework utilizado para empaquetar la aplicación de escritorio, permitiendo el desarrollo de aplicaciones ligeras que utilizan tecnologías web. Proporciona una interfaz nativa y un rendimiento óptimo al crear aplicaciones que combinan frontend y backend de manera efectiva.

Expo

Figura 14:



Expo.js - (<https://negativeepsilon.com/es/blog/expo-setup-project/>)

Herramienta para el desarrollo de aplicaciones móviles con React Native. Facilita el proceso de creación, prueba y despliegue de la aplicación móvil, ofreciendo acceso a componentes nativos y una API simplificada para acceder a funcionalidades del dispositivo.

Firebase

Figura 15:



Firestore 1 - (<https://www.pngwing.com/en/free-png-azwzr>)

Plataforma backend que proporciona servicios como autenticación, base de datos en tiempo real y almacenamiento. Se integra perfectamente con otras tecnologías del proyecto, permitiendo una gestión eficiente de usuarios y datos.

TensorFlow

Figura 16:



TensorFlow - (<https://www.pngwing.com/en/free-png-azfhe>)

Biblioteca de machine learning utilizada para el análisis de datos e implementación de modelos predictivos. TensorFlow permite al proyecto realizar análisis de inventario y ofrecer recomendaciones basadas en patrones de uso, mejorando la eficiencia del taller.

Gemini

Figura 17:



Gemini - (<https://uxwing.com/google-gemini-icon/>)

Se está implementando a modo de beta dentro de nuestro software como una herramienta que genera recomendaciones personalizadas para los mecánicos, como informes detallados sobre nuevos autos ingresados al taller. Facilita la toma de decisiones informadas basadas en datos históricos, contribuyendo a un servicio más eficaz y proactivo.

Vite

Figura 18:



Vite - (<https://worldvectorlogo.com/es/logo/vitejs>)

Herramienta de desarrollo frontend y backend que mejora la experiencia de desarrollo al ofrecer tiempos de carga más rápidos y una configuración sencilla. Se utiliza para crear y optimizar la aplicación, haciendo uso de su soporte para módulos ES y Hot Module Replacement (HMR).

JavaScript

Figura 19:



JavaScript - (<https://worldvectorlogo.com/es/logo/logo-javascript>)

JavaScript es un lenguaje de programación dinámico y versátil, utilizado principalmente para crear interactividad en las páginas web. Permite manipular el contenido del DOM (Document Object Model), gestionar eventos y realizar tareas tanto en el frontend como en el backend.

Jest

Figura 20:



Jest - (<https://worldvectorlogo.com/es/logo/jest-2>)

Jest es un framework de pruebas en JavaScript diseñado para realizar pruebas unitarias, de integración y de extremo a extremo. Facilita la simulación de funciones y módulos, soporta pruebas asíncronas y captura de snapshots. Ofrece reportes detallados y se integra bien con proyectos de React. Es una opción popular por su simplicidad y funcionalidad lista para usar.

Testing Library

Figura 21:



Testing Library – (<https://www.logiciels.pro/logiciel-saas/react-testing-library/>)

Testing Library es una colección de utilidades para probar interfaces de usuario en aplicaciones web. Se enfoca en realizar pruebas basadas en el comportamiento del usuario, priorizando la interacción con elementos como un usuario real lo haría. Es compatible con frameworks como React, Vue y Angular, y se integra con Jest para facilitar la ejecución de pruebas. Su objetivo es mejorar la legibilidad y mantener pruebas robustas y fáciles de mantener.

Vitest

Figura 22:



Vitest - (<https://seeklogo.com/vector-logo/434979/vitest>)

Vitest es un framework de pruebas para aplicaciones JavaScript y TypeScript, optimizado para trabajar con Vite. Ofrece pruebas unitarias, de integración y de extremo a extremo, con soporte nativo para módulos ES y funcionalidades similares a Jest, como mocks y spies. Es rápido, fácil de configurar y está diseñado para integrarse de manera eficiente con proyectos modernos basados en Vite.

CodeScene

Figura 23:



CodeScene - (<https://empresaytrabajo.coop/Nexus-Repository-Manager-Gatekeeper-Plugin-Snyk-User-Docs-655505.html>)

CodeScene es una herramienta de análisis de código que utiliza patrones de desarrollo y análisis de comportamiento para identificar áreas problemáticas en el código. Ayuda a detectar "deuda técnica", medir la calidad del código y evaluar la productividad del equipo. Utiliza visualizaciones para resaltar áreas complejas y de riesgo, facilitando la toma de decisiones en el mantenimiento y evolución del software. Es útil para priorizar mejoras y optimizar el desarrollo a largo plazo.

8 Detalle de la Arquitectura a implementar

8.1 Diagrama BPMN

Figura 24:

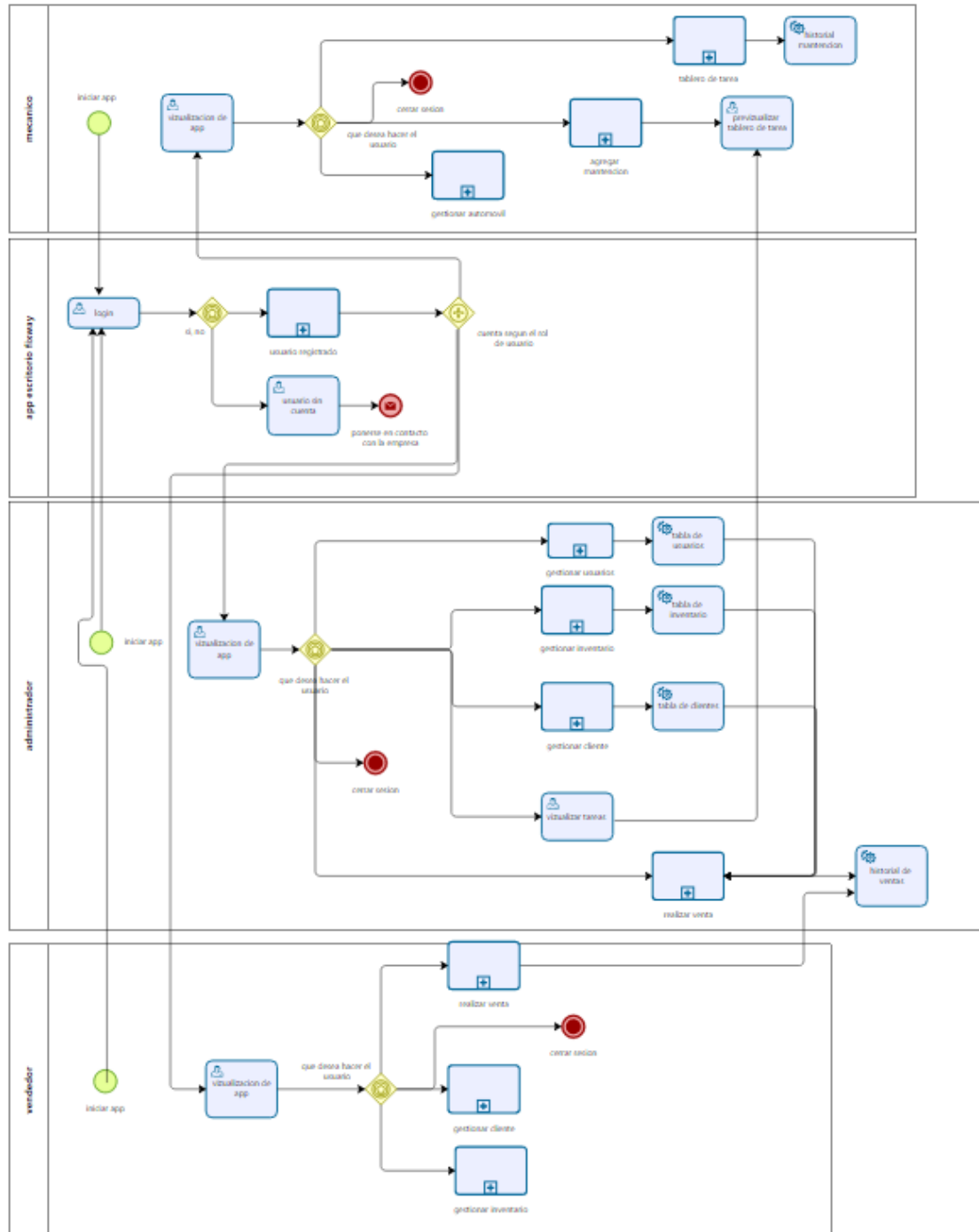


Diagrama BPMN 1 (Creación Propia)

Iniciar sesión, usuario registrado

Figura 25:

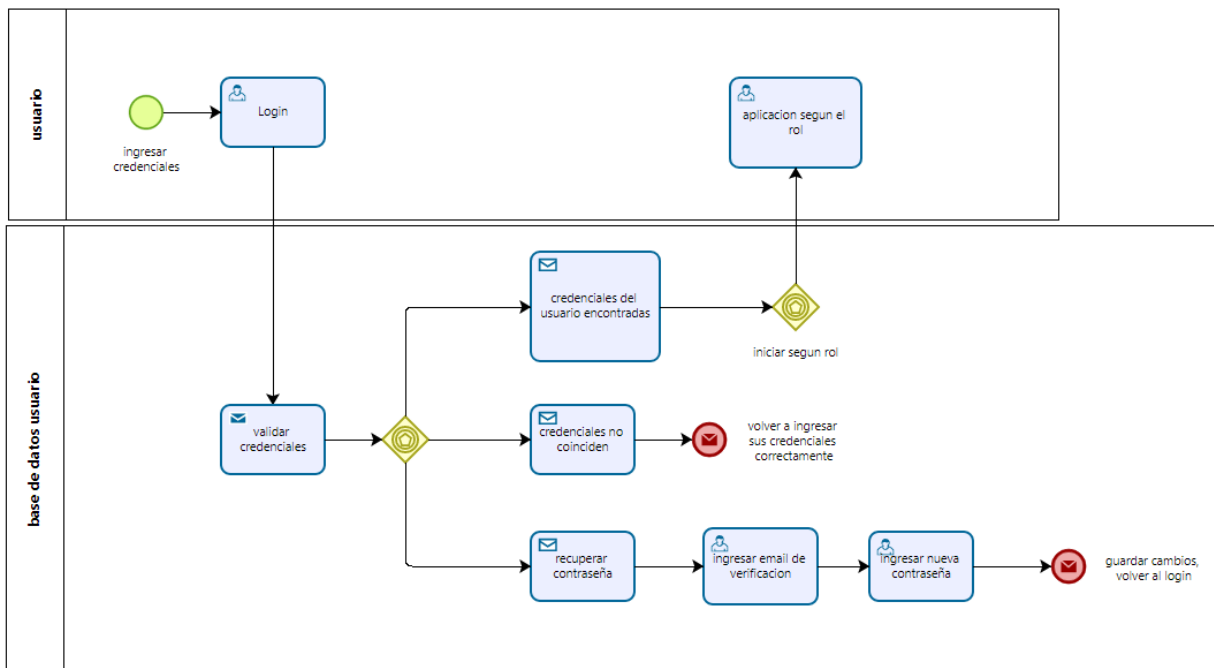


Diagrama BPMN 2 (Creación Propia)

Administrador-vendedor, gestionar cliente

Figura 26:

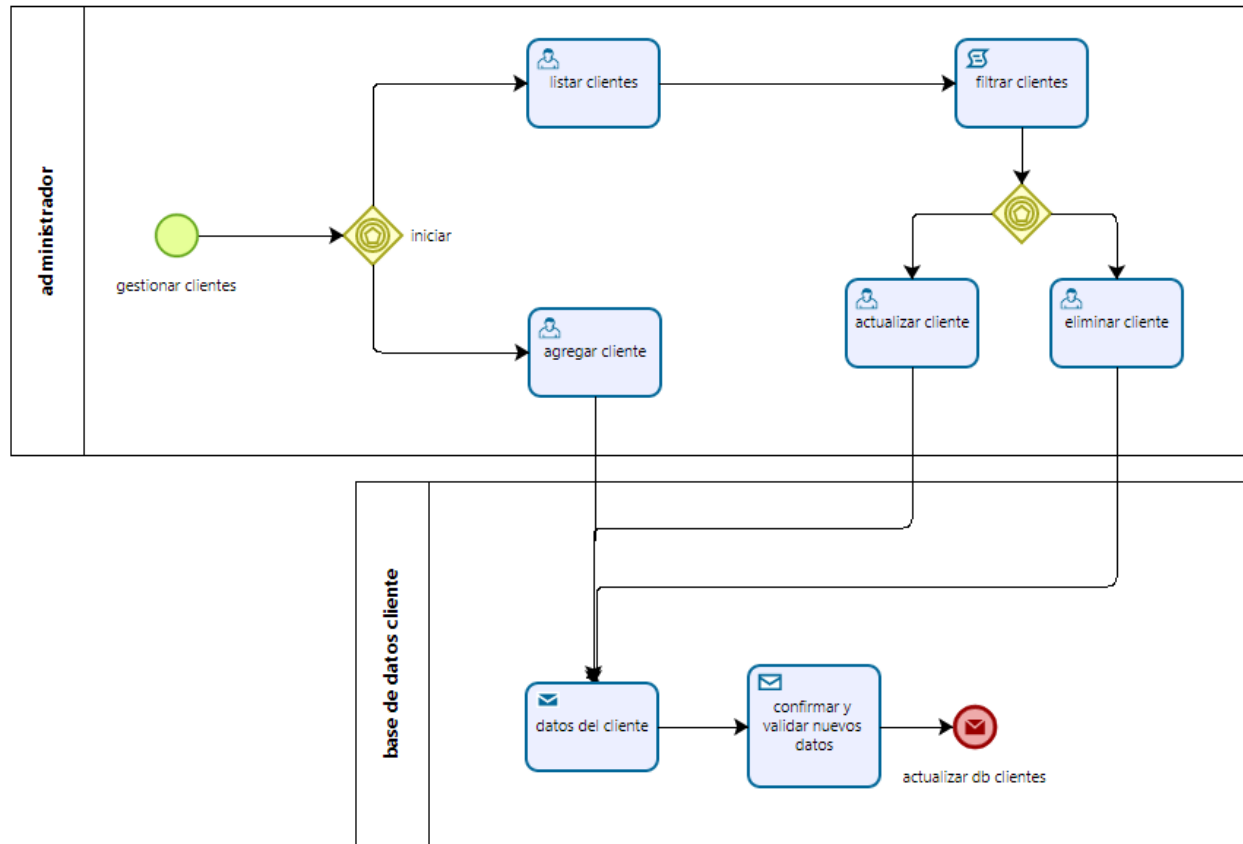


Diagrama BPMN 3 (Creación Propia)

Administrador-vendedor, gestionar inventario

Figura 27:

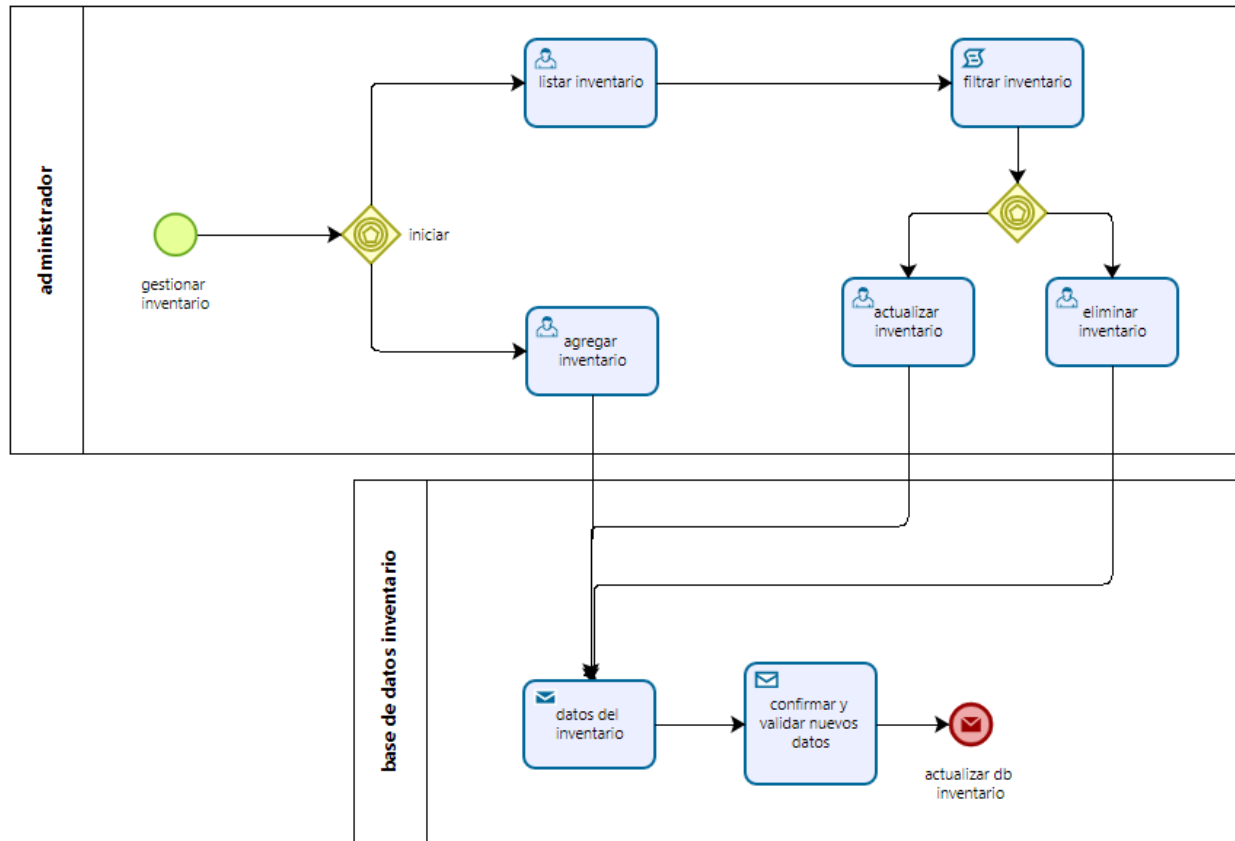


Diagrama BPMN 4 (Creación Propia)

Administrador-vendedor, gestionar cliente

Figura 28:

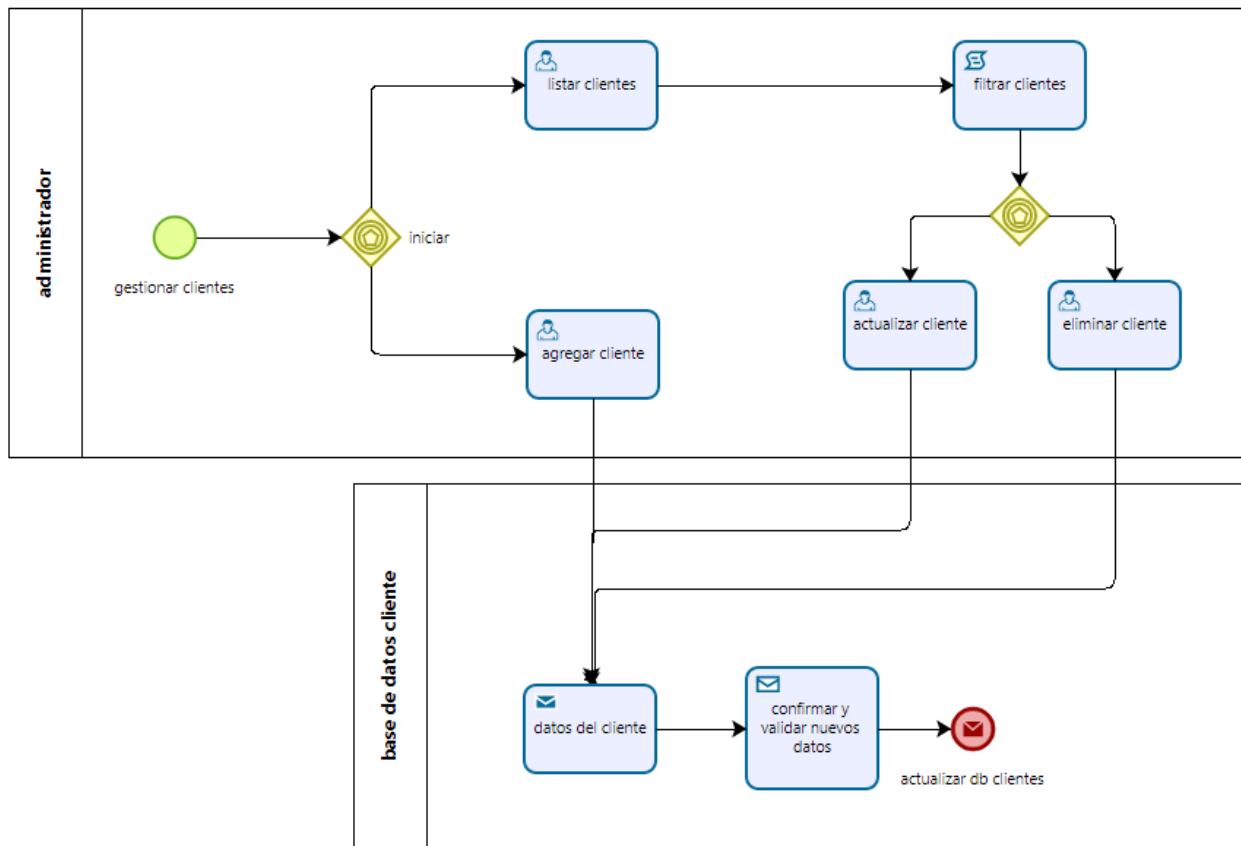


Diagrama BPMN 5 (Creación Propia)

Administrador-vendedor, realizar venta

Figura 29:

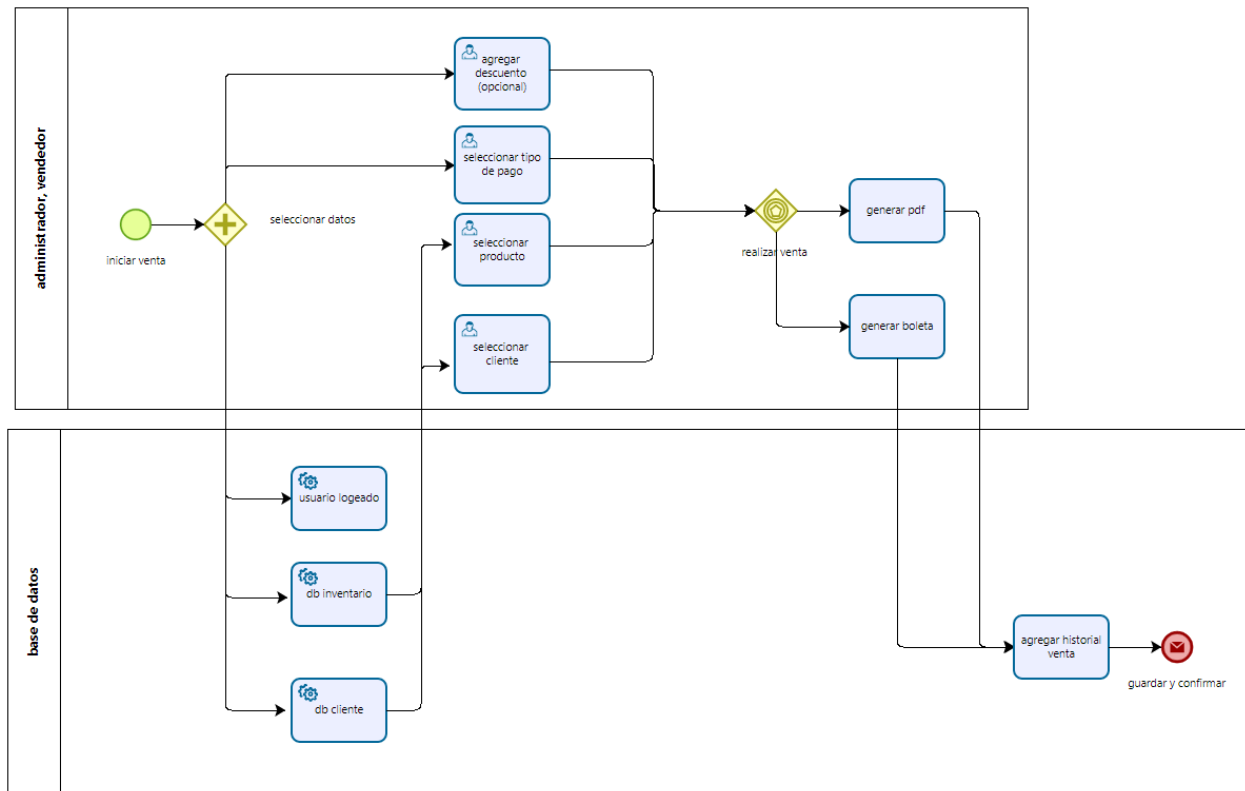


Diagrama BPMN 6 (Creación Propia)

Mecánico, gestionar automóvil

Figura 30:

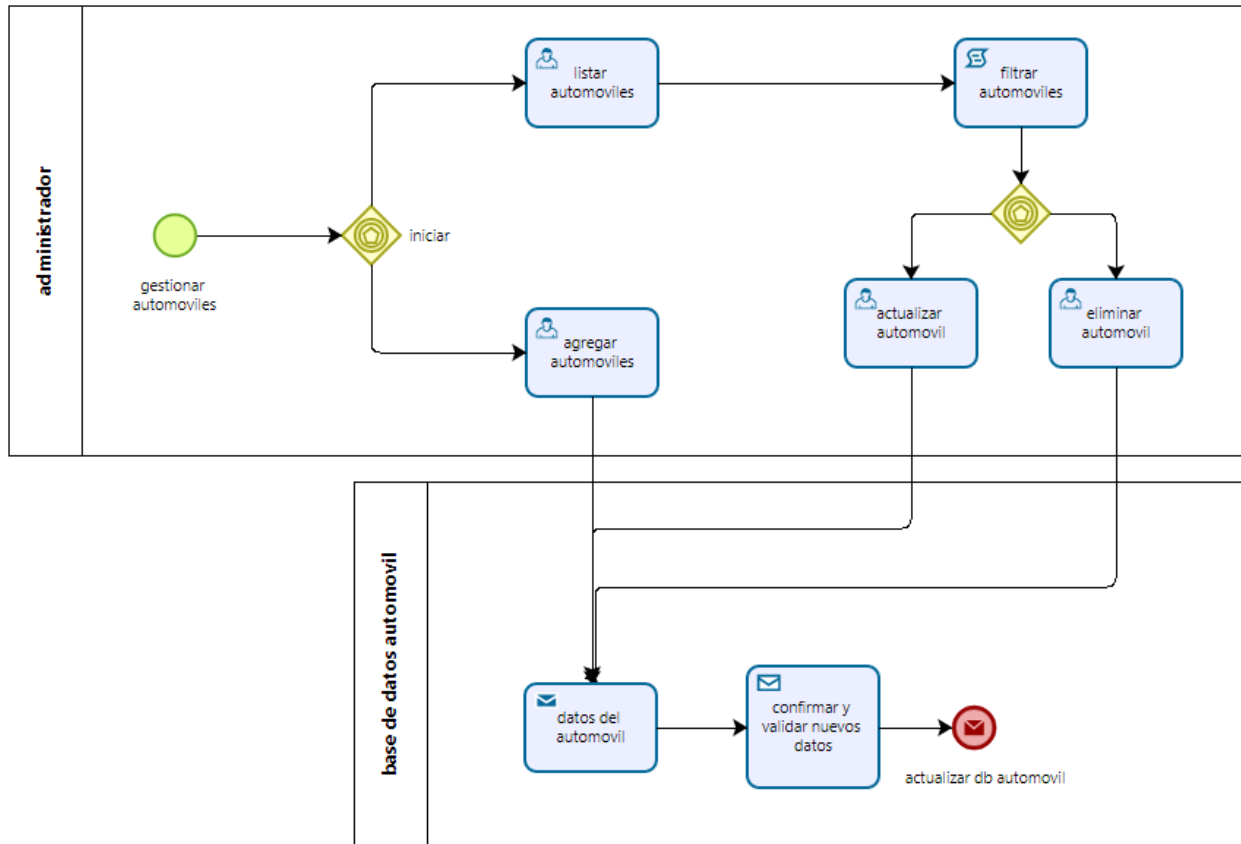


Diagrama BPMN 7 (Creación Propia)

Mecánico, agregar mantención

Figura 31:

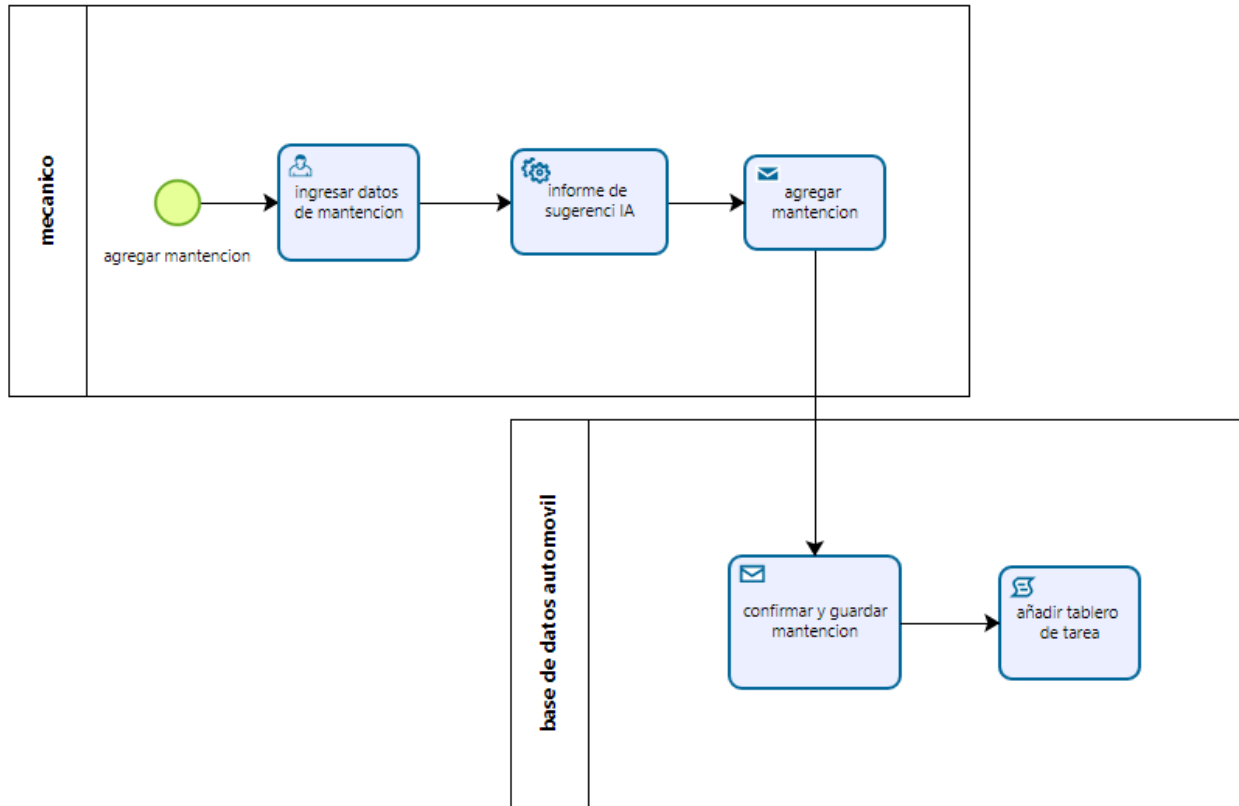


Diagrama BPMN 8 (Creación Propia)

Mecánico, tablero tareas

Figura 32:

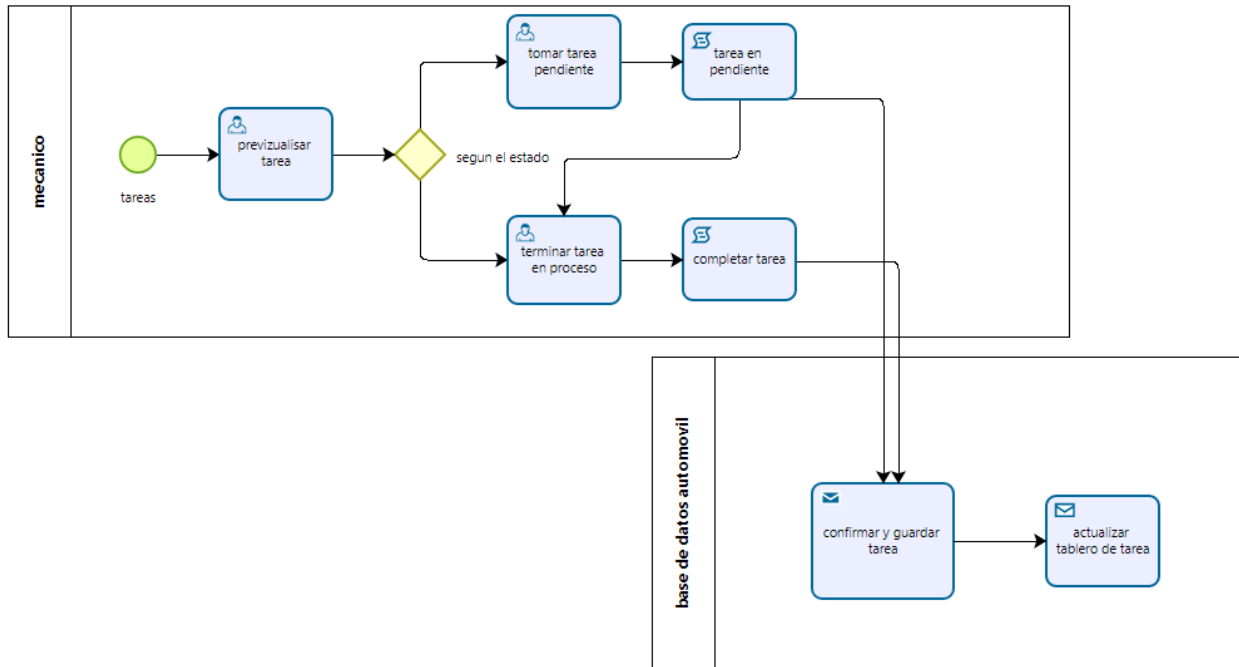


Diagrama BPMN 9 (Creación Propia)

8.2 Diagramas de caso de uso

Figura 33:

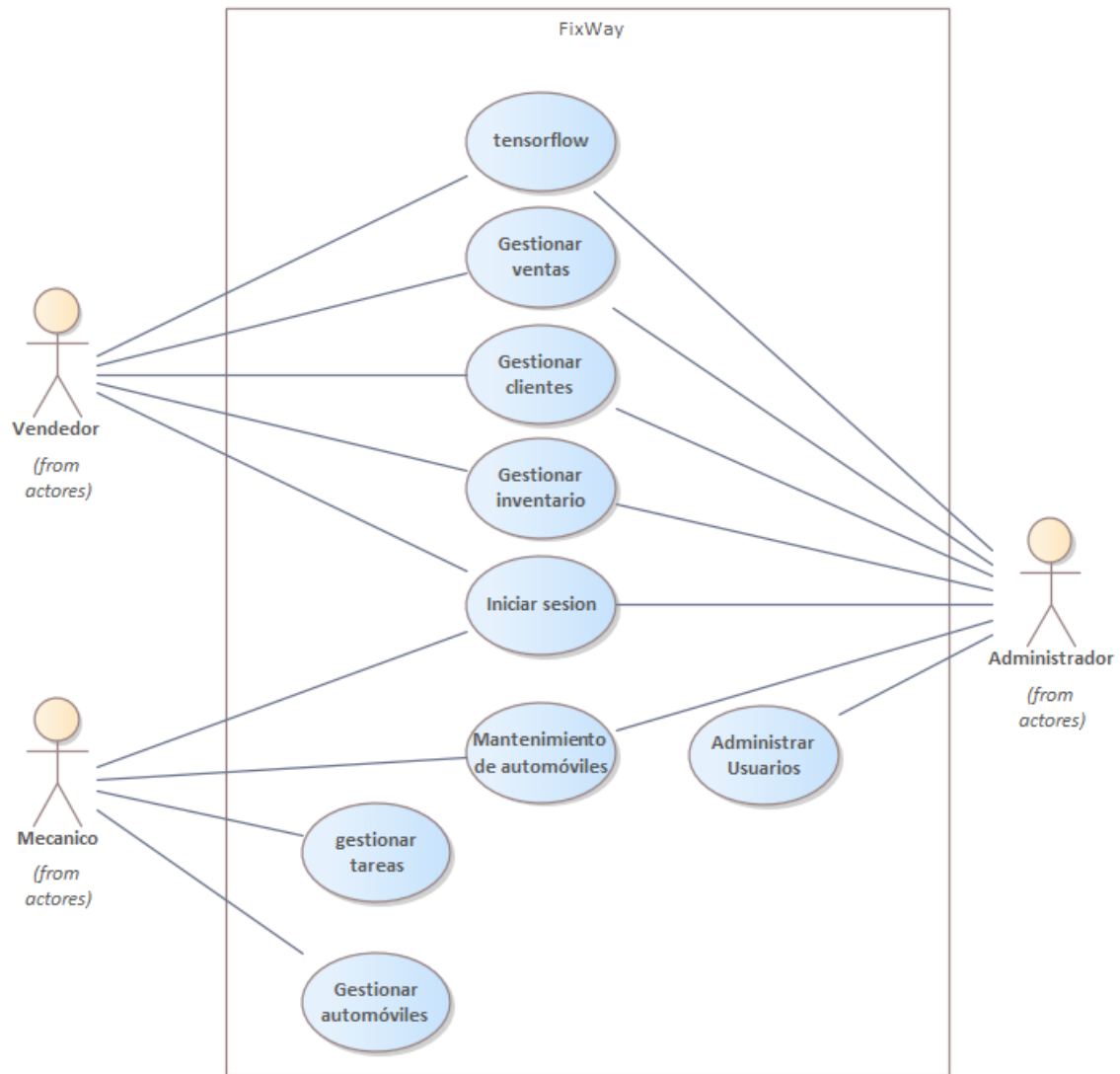


Diagrama de alto nivel (Creación Propia)

Figura 34:

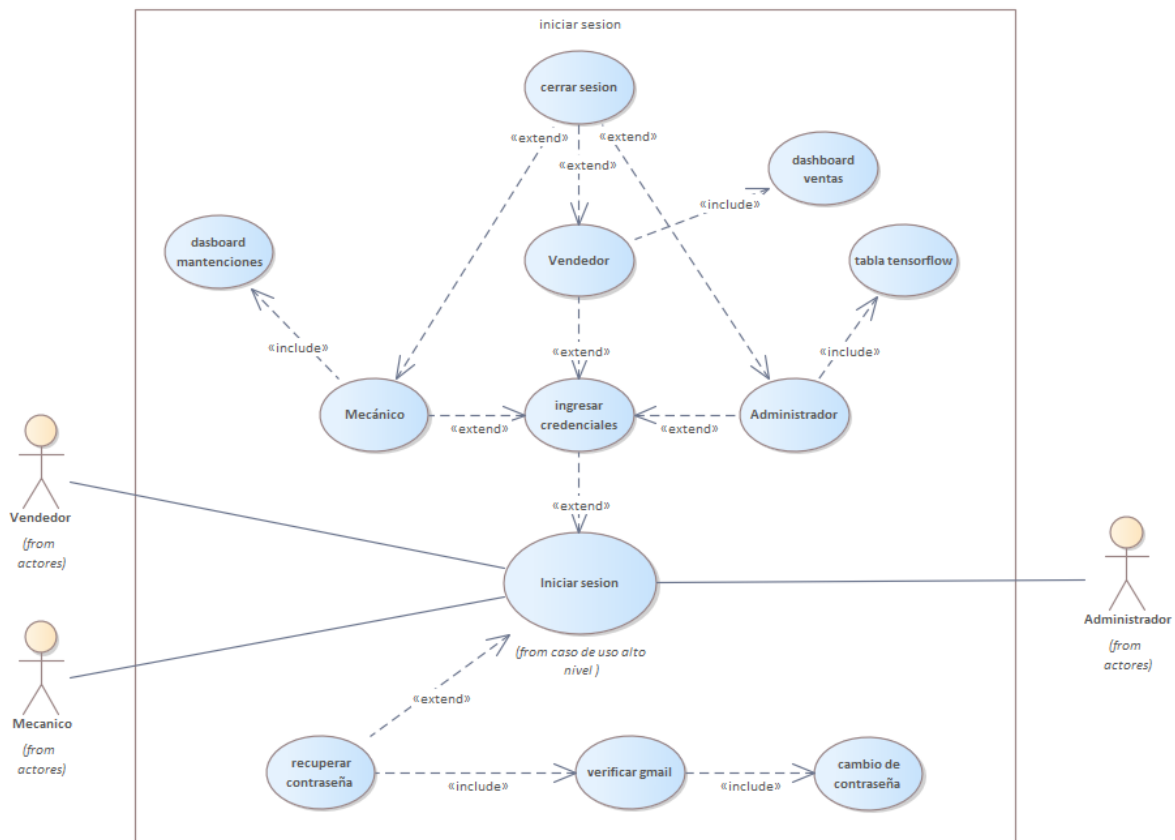


Diagrama de nivel detallado - iniciar sesión (Creación Propia)

Figura 35:

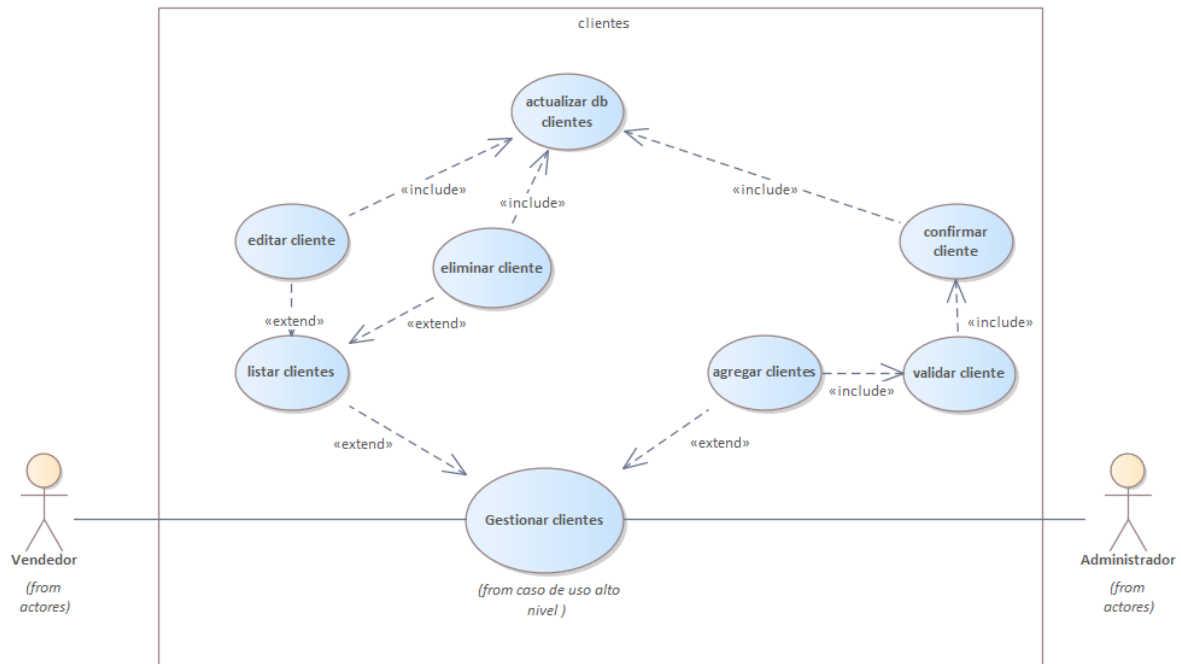


Diagrama de nivel detallado - clientes (Creación Propia)

Figura 36:

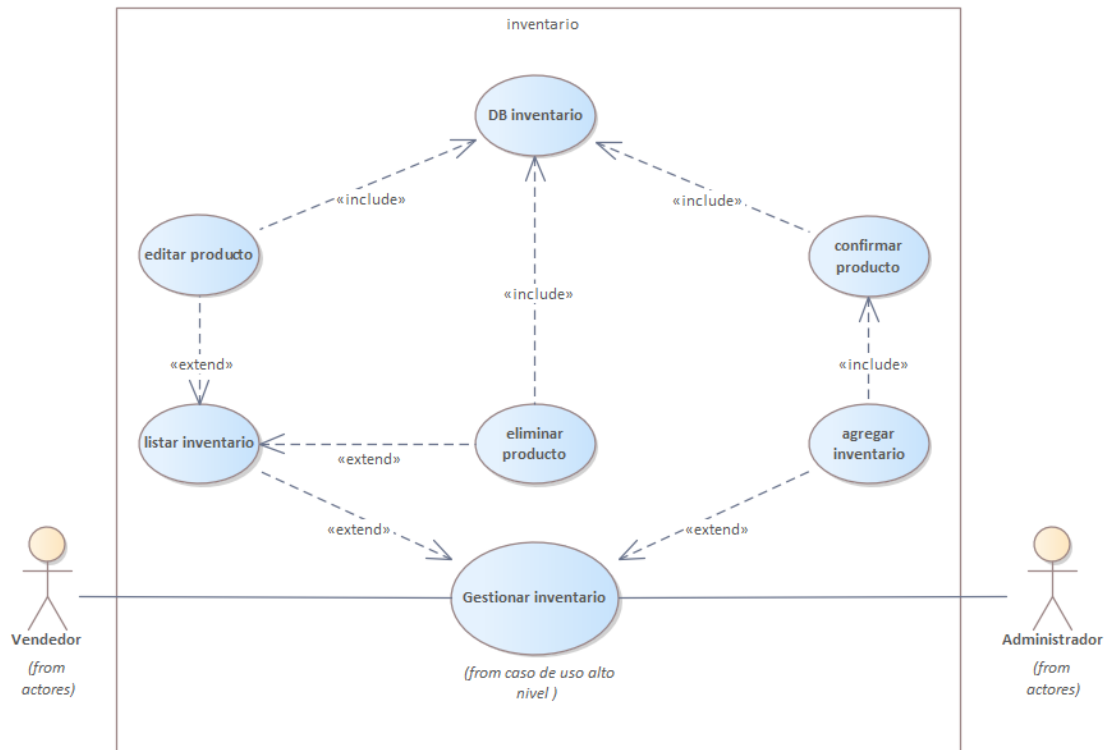


Diagrama de nivel detallado – inventario (Creación Propia)

Figura 37:

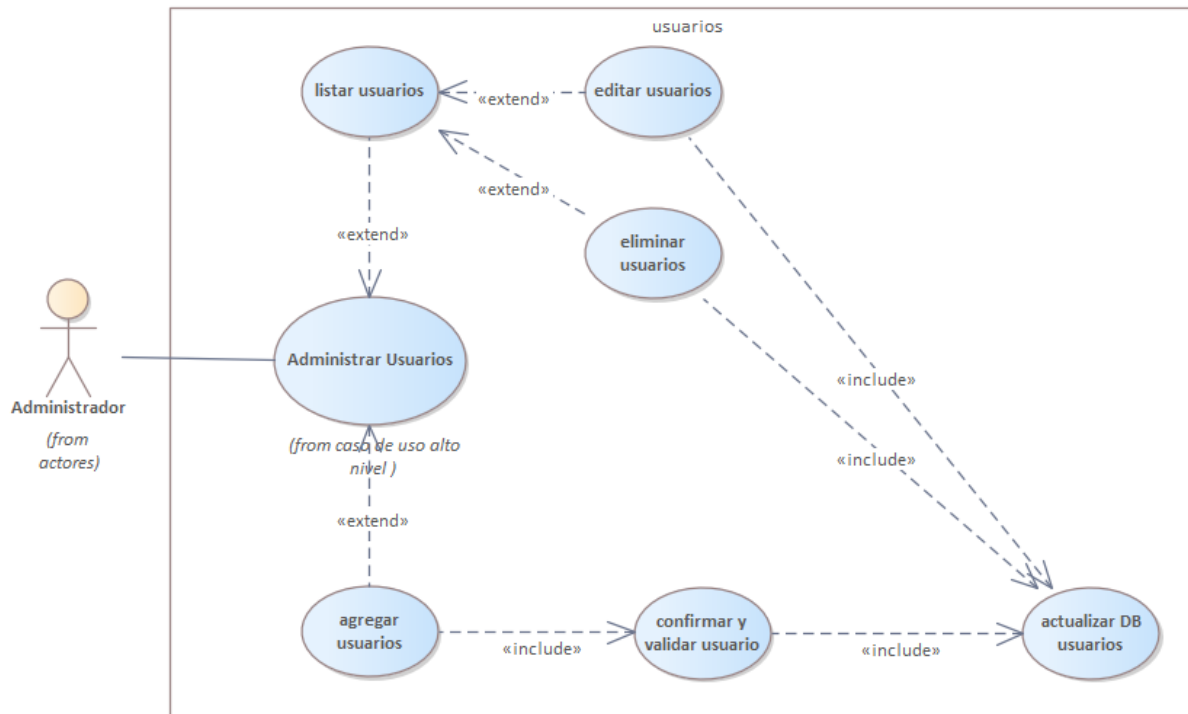


Diagrama de nivel detallado – usuario (Creación Propia)

Figura 38:

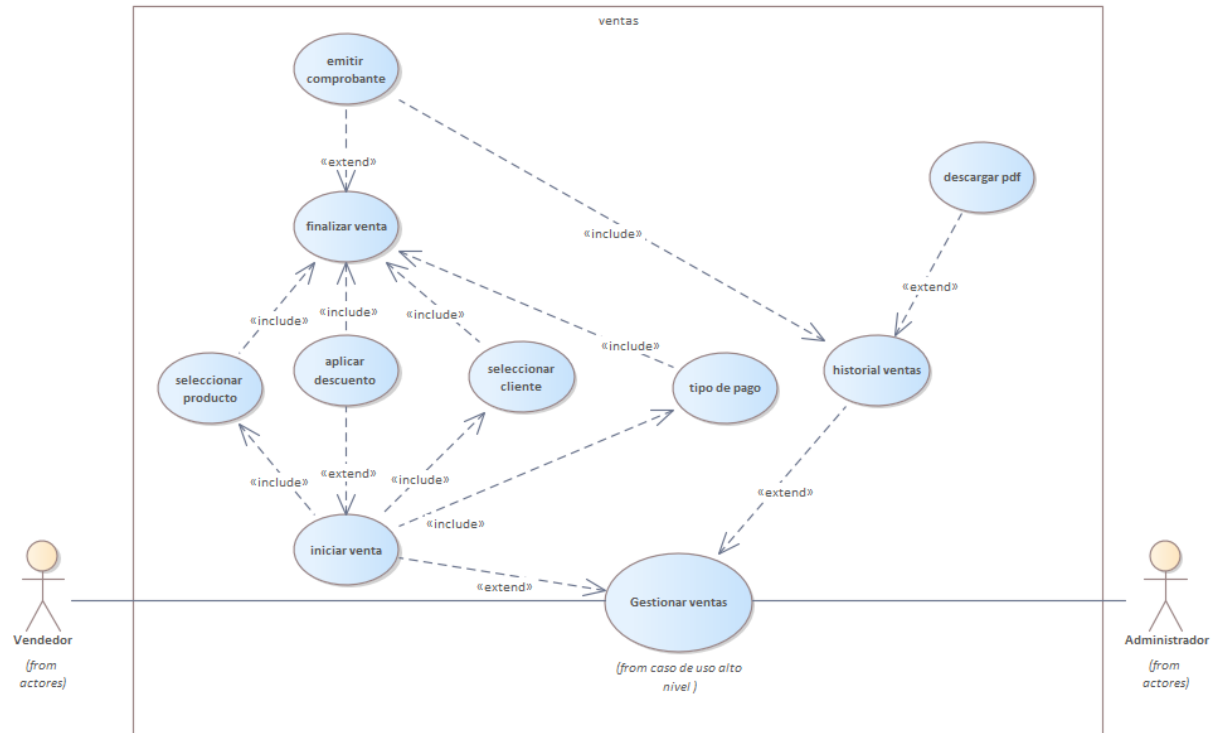


Diagrama de nivel detallado – ventas (Creación Propia)

Figura 39:

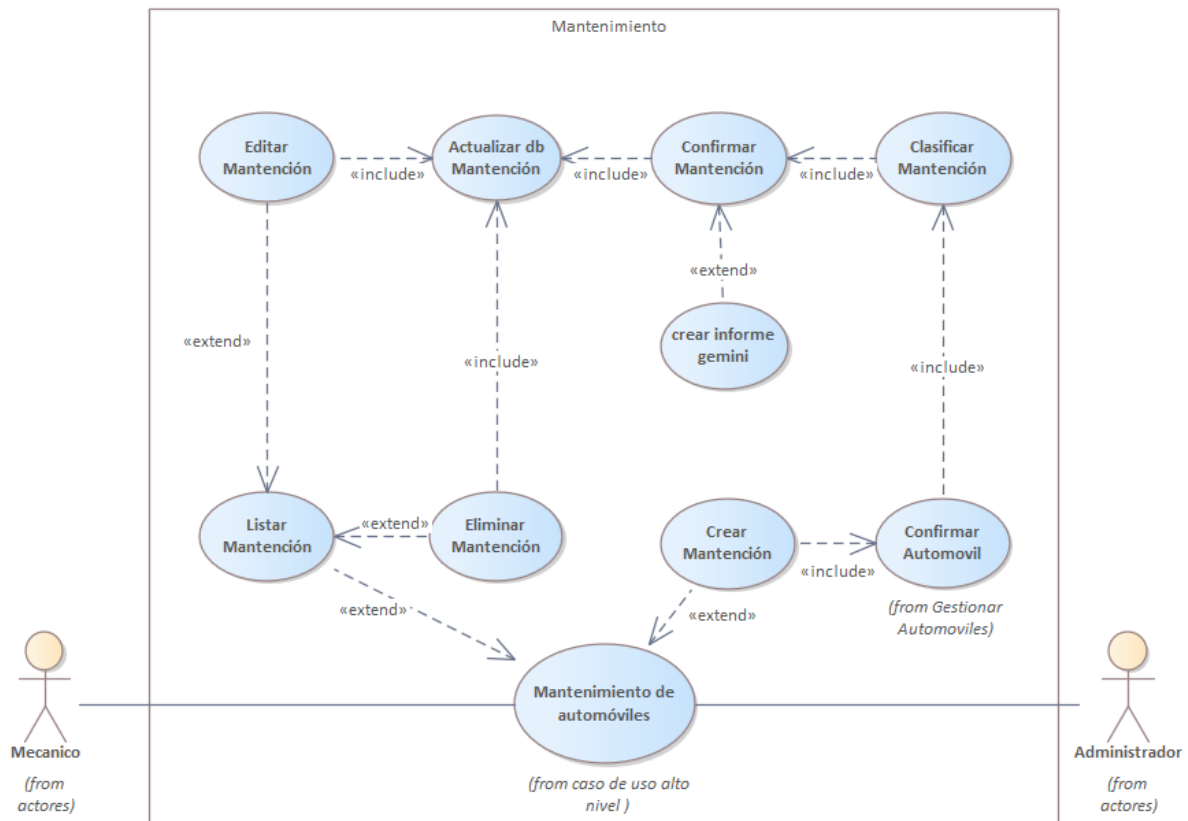


Diagrama de nivel detallado – mantenimiento (Creación Propia)

Figura 40:

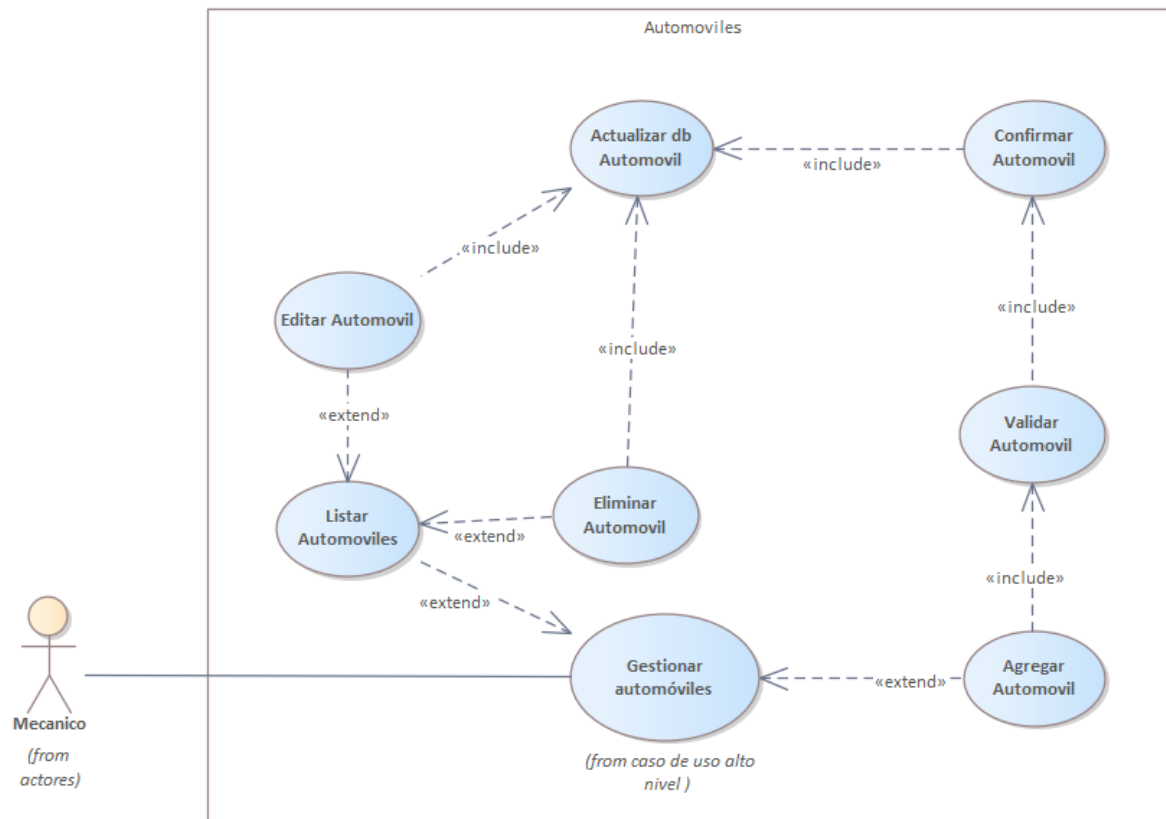


Diagrama de nivel detallado – automóviles (Creación Propia)

8.3 Diagrama de componentes

Figura 41:

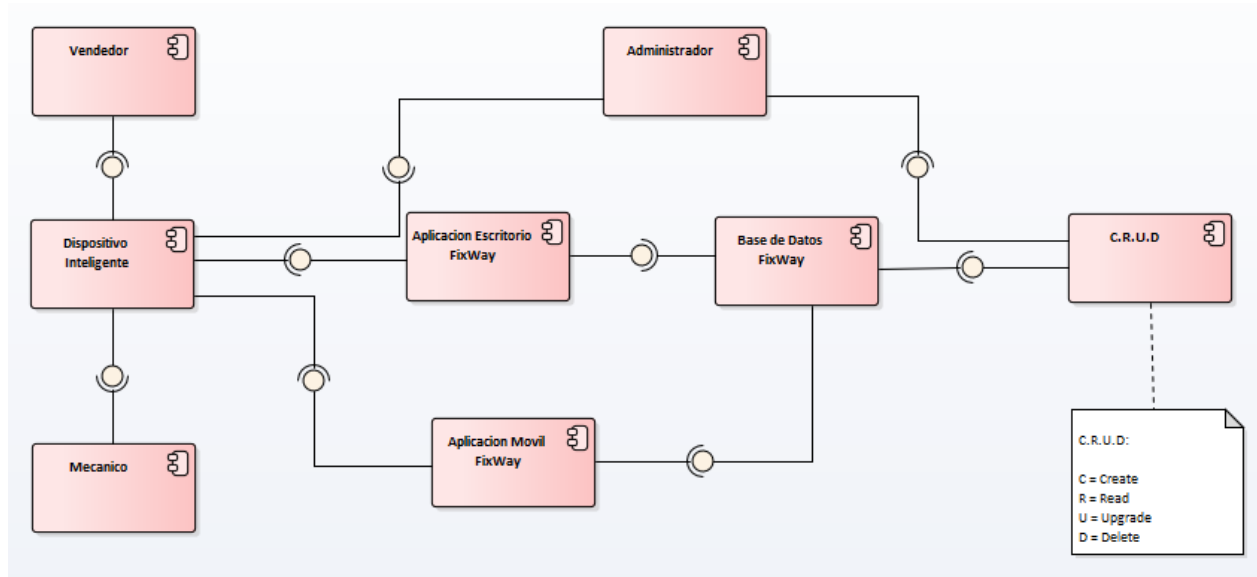
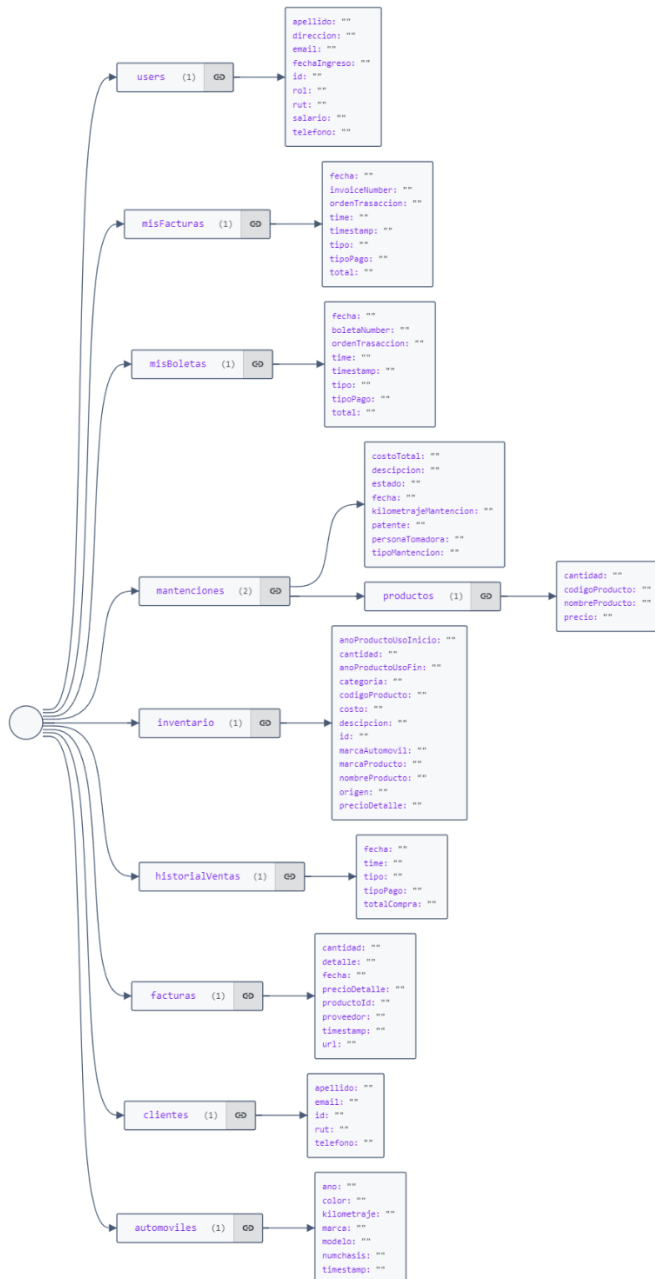


Diagrama de componentes (Creación Propia)

8.4 Modelo de datos

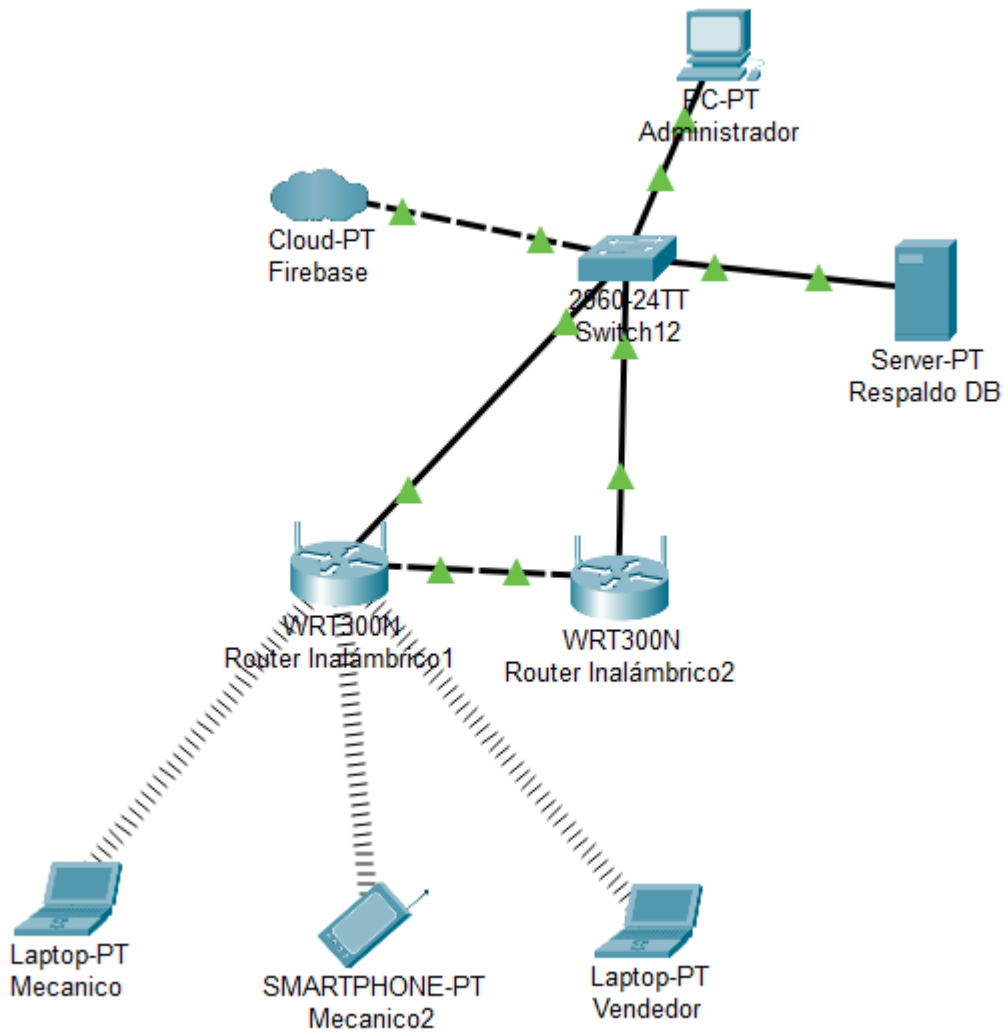
Figura 42:



Estructura de base de datos (Creación Propia)

8.5 Topologías de comunicación

Figura 43:



Topología de comunicaciones (Creación Propia)

8.6 Diagrama de Infraestructura

Figura 44:

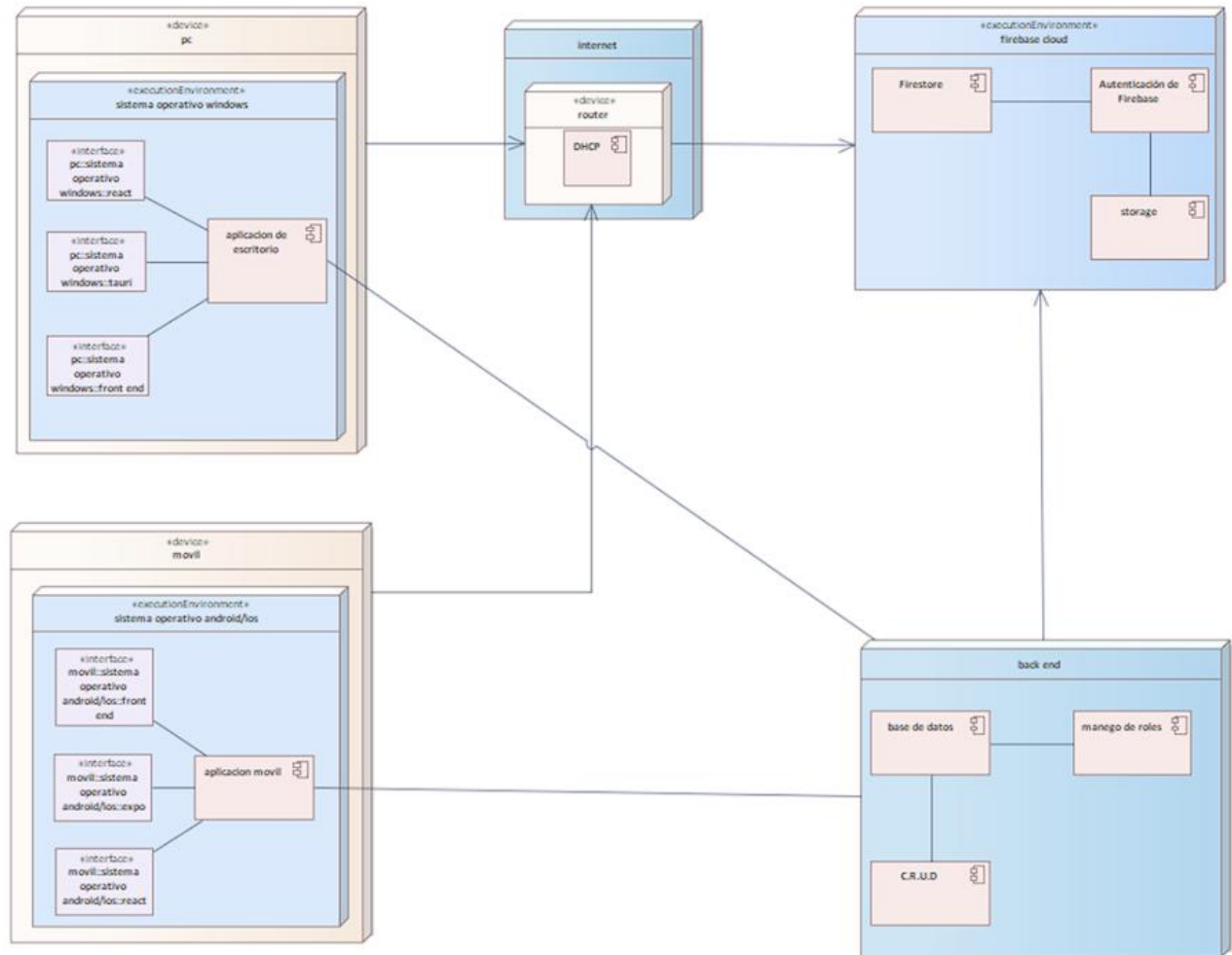


Diagrama de infraestructura (Creación Propia)

8.7 Diagrama de Arquitectura

Figura 45:

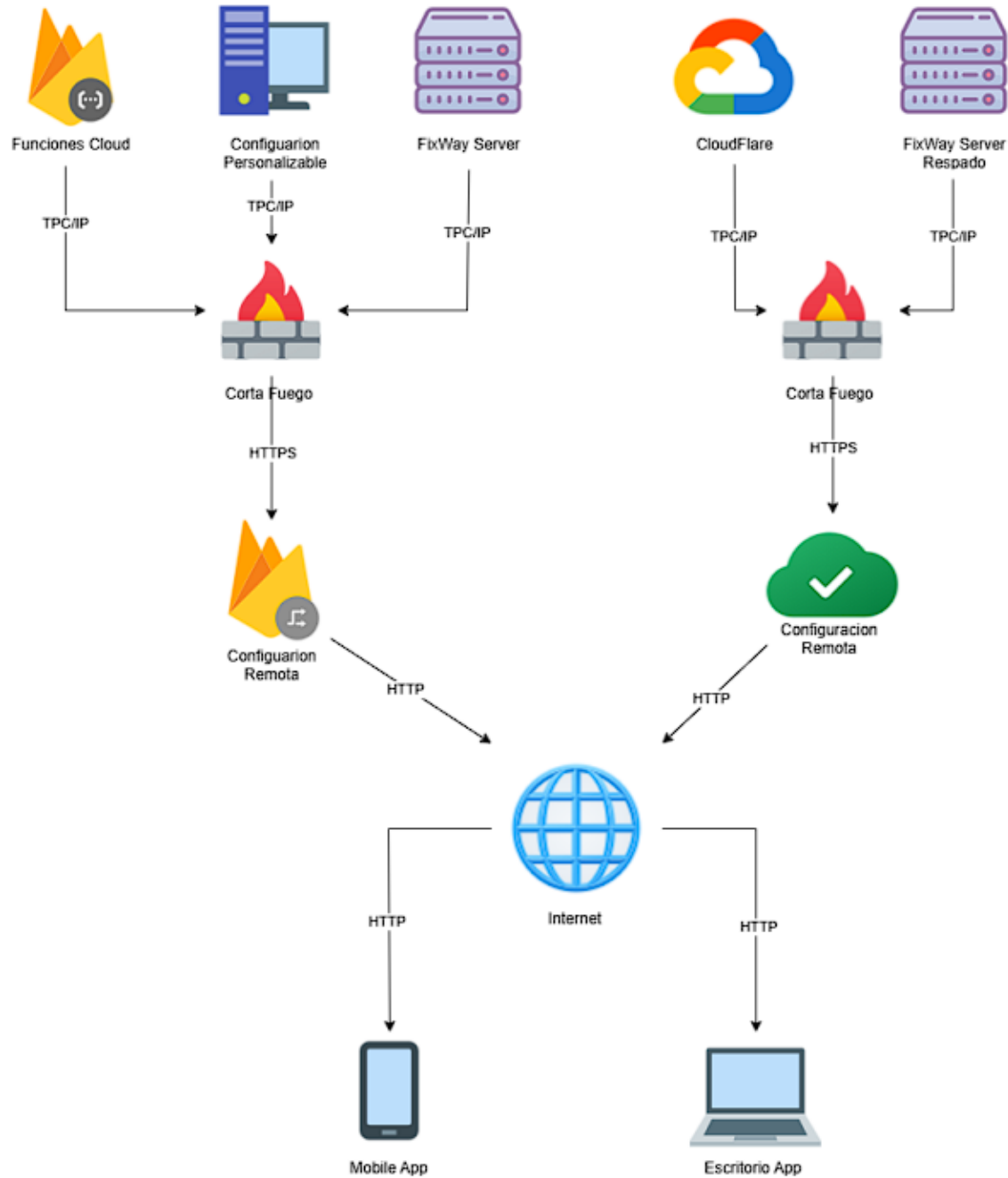


Diagrama de Arquitectura (Creación Propia)

9 Plan de Pruebas y Aseguramiento de Calidad

9.1 Plan de Pruebas

Durante el proceso de pruebas de software se emplearon tanto pruebas de caja negra como pruebas de caja blanca para asegurar la robustez, fiabilidad y calidad del sistema desarrollado. A continuación, se presenta un resumen de ambos enfoques y de las pruebas de estabilidad y conexiones activas realizadas:

1. Pruebas de Caja Negra

Para la ejecución de pruebas de caja negra se utilizaron los marcos de prueba **Jest** y **React Testing Library**, estableciendo un entorno de pruebas independiente, cuidadosamente separado tanto del entorno de desarrollo como del entorno de operación. Con estas herramientas, se implementaron técnicas de fijación de datos (mocking) para garantizar la estabilidad y consistencia de los datos de prueba, asegurando imparcialidad y objetividad en los resultados obtenidos.

Este enfoque permitió validar de manera integral todas las funcionalidades de la aplicación, verificando su correcto funcionamiento y la consistencia de los datos, lo que contribuyó a la robustez y fiabilidad del sistema. React Testing Library, en particular, fue utilizada para reflejar situaciones del mundo real, simulando eventos y cambios de estado específicos para probar cada función o característica por separado. Además, se emplearon las funciones de aserción de Jest para asegurar que las salidas de las pruebas fueran representativas del comportamiento esperado, garantizando así que los resultados se alinearan con los objetivos del sistema.

2. Pruebas de Caja Blanca

El análisis del código fuente se realizó mediante la herramienta **CodeScene**, configurada de manera personalizada para evaluar la calidad del código, identificar patrones de desarrollo y gestionar la deuda técnica. Este proceso incluyó una revisión detallada de los cambios recientes, revelando áreas de alta actividad y tendencias significativas que podrían afectar la arquitectura del proyecto.

La evaluación de la calidad del código se basó en métricas específicas proporcionadas por CodeScene, como complejidad, acoplamiento y duplicación de código. Esto facilitó la identificación de áreas críticas y la elaboración de estrategias concretas para la reducción de la deuda técnica y la mejora de la salud general del proyecto a largo plazo.

Además, se utilizó **Vitest**, un marco de prueba para JavaScript, conocido por su configuración sencilla y versatilidad, compatible con varios frameworks como React, Vue.js y Svelte. Vitest facilitó la escritura y ejecución de pruebas, proporcionando una interfaz de usuario interactiva que permitió obtener una visión clara de los resultados y simplificar la depuración y resolución de problemas.

3. Pruebas de Estabilidad y Conexiones Activas

Para asegurar que la aplicación pueda manejar distintos escenarios de carga y garantizar su rendimiento a largo plazo, se implementaron varias pruebas de estabilidad:

3.1.1. Prueba de Carga (Load Testing):

- **Objetivo:** Evaluar cómo responde la aplicación bajo condiciones normales de carga.
- **Implementación:** Se simuló múltiples usuarios interactuando con la interfaz de usuario para medir la capacidad de respuesta del sistema, identificando el rendimiento esperado en condiciones estándar de uso.

3.1.2. Prueba de Estrés (Stress Testing):

- **Objetivo:** Evaluar la resistencia de la aplicación bajo condiciones extremas.
- **Implementación:** Se aplicaron cargas significativamente superiores a las condiciones normales para verificar la estabilidad del sistema y detectar posibles puntos de fallo.

3.1.3. Prueba de Resistencia (Endurance Testing):

- Objetivo: Evaluar la estabilidad a largo plazo de la aplicación.
- Implementación: Se ejecutó la aplicación con una carga constante durante un período prolongado para identificar posibles problemas relacionados con la memoria o los recursos que podrían acumularse con el tiempo.

3.1.4. Prueba de Escalabilidad (Scalability Testing):

- Objetivo: Evaluar la capacidad de la aplicación para escalar al aumentar la carga o los recursos.
- Implementación: Se incrementó gradualmente la carga o los recursos para medir la capacidad de respuesta del sistema, identificar los puntos de saturación y determinar áreas de mejora en términos de escalabilidad.

9. Pruebas de tiempo de respuesta

1. Testing Aplicación Escritorio

Figura 46:

```
> HM@0.0.0 test
> vitest

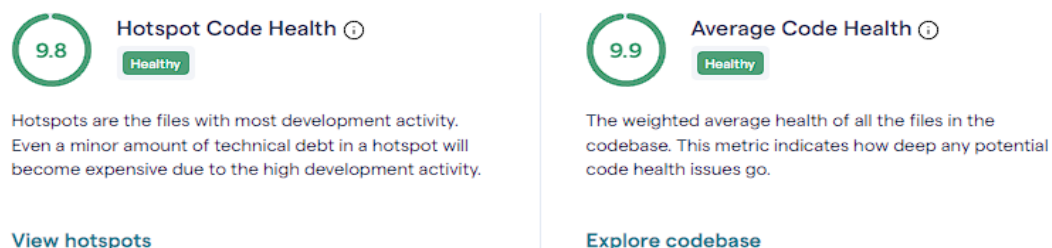
DEV v0.34.6 C:/Users/sgqui/OneDrive/Escritorio/HansMotors

✓ src/test/fizzbuzz.test.js (10)
✓ src/test/agregarInventario.test.jsx (1)
✓ src/test/listarInventario.test.jsx (1)
✓ src/test/admin.test.jsx (1)
✓ src/test/generarFactura.test.jsx (1)
✓ src/test/indexAdmin.test.jsx (1)
✓ src/test/generarListadoMantencion.test.jsx (1)
✓ src/test/listarInventarioMecanico.test.jsx (1)
✓ src/test/agregarFactura.test.jsx (1)
✓ src/test/gestionMantenciones.test.jsx (1)
✓ src/test/agregarUsuario.test.jsx (1)
✓ src/test/GenerarQR.test.jsx (1)
✓ src/test/gestionMantencionAdmin.test.jsx (1)
✓ src/test/listadoFactura.test.jsx (1)
✓ src/test/listarUsuario.test.jsx (1)
✓ src/test/mecanico.test.jsx (1)
✓ src/test/indexMecanico.test.jsx (1)

Test Files 17 passed (17)
Tests      26 passed (26)
Start at   22:02:47
Duration   5.70s (transform 910ms, setup 3ms, collect 7.48s,
PASS Waiting for file changes...
press h to show help, press q to quit
```

Testing App Escritorio 2023 (Creación Propia)

Figura 47:



Testing App Escritorio 2023 (Creación Propia)

Figura 48:

```
✓ src/Test/HistorialMantencion.test.jsx (2) 564ms
✓ src/Test/AgregarFactura.test.jsx (1) 563ms
✓ src/Test/AgregarInventario.test.jsx (1) 459ms
✓ src/Test/GestionMantencion.test.jsx (2) 611ms
✓ src/Test/GenerarFacturaVendedor.test.jsx (1) 441ms
✓ src/Test/ListarCliente.test.jsx (1) 475ms
✓ src/Test/AgregarAutomovil.test.jsx (1) 522ms
✓ src/Test/GenerarFactura.test.jsx (1) 518ms
✓ src/Test/ListarUsuario.test.jsx (1) 367ms
✓ src/Test/AgregarUsuario.test.jsx (1) 380ms
✓ src/Test/ListadoAutomovil.test.jsx (1) 411ms
✓ src/Test/ListadoFacturas.test.jsx (1) 417ms
✓ src/Test/GenerarQR.test.jsx (2) 1071ms
✓ src/Test/GenerarQRMecanico.test.jsx (2) 821ms
✓ src/Test/HistorialVentas.test.jsx (1)
✓ src/Test/AgregarCliente.test.jsx (1) 326ms
✓ src/Test/ListarInventario.test.jsx (1) 314ms
✓ src/Test/ListadoMisFacturas.test.jsx (1)
✓ src/Test/GestionMantencionesMecanico.test.jsx (1)
✓ src/Test/GraficoMisFacturas.test.jsx (1)
✓ src/Test/ProductoMasVendidoMecanico.test.jsx (1)
✓ src/Test/GraficoMisFacturasVendedor.test.jsx (1)
✓ src/Test/GenerarListadoMantencionMecanico.test.jsx (1)
✓ src/Test/GraficoMisBoletas.test.jsx (1)
✓ src/Test/GraficoMisBoletasVendedor.test.jsx (1)
✓ src/Test/GraficoTipoPagoVendedor.test.jsx (1)
✓ src/Test/GraficoTipoPago.test.jsx (1)
✓ src/Test/HistorialVentasVendedor.test.jsx (1)
✓ src/Test/AgregarAutomovilMecanico.test.jsx (1)
✓ src/Test/ListarInventarioMecanico.test.jsx (1)

Test Files 30 passed (30)
Tests 34 passed (34)
Start at 10:46:40
Duration 34.09s (transform 2.02s, setup 4ms, collect 125.15s, tests 10.23s, environment 15.73s, prepare 18.40s)

PASS Waiting for file changes...
press h to show help, press q to quit
```

Testing App Escritorio 2024 (Creación Propia)

Figura 49:



Testing App Escritorio 2024 (Creación Propia)

2. Testing Aplicación Móvil

Figura 50:

```

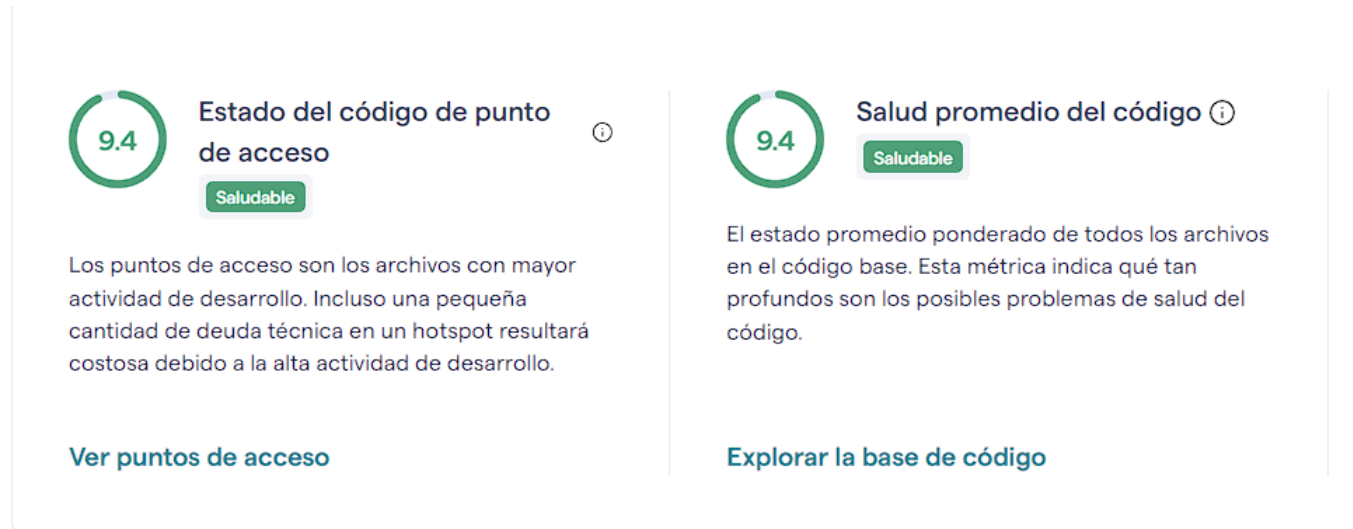
✓ Run expo doctor

1 Thu, 16 Nov 2023 22:20:54 GMT Running "expo doctor"
2 Thu, 16 Nov 2023 22:20:56 GMT Running 10 checks on your project...
3 Thu, 16 Nov 2023 22:20:56 GMT ✓ Check Expo config for common issues
4 Thu, 16 Nov 2023 22:20:56 GMT ✓ Check package.json for common issues
5 Thu, 16 Nov 2023 22:20:56 GMT ✓ Check dependencies for packages that should not be installed directly
6 Thu, 16 Nov 2023 22:20:56 GMT ✓ Check for common project setup issues
7 Thu, 16 Nov 2023 22:20:56 GMT ✓ Check npm/ yarn versions
8 Thu, 16 Nov 2023 22:20:56 GMT ✓ Check Expo config (app.json/ app.config.js) schema
9 Thu, 16 Nov 2023 22:21:00 GMT ✓ Check that packages match versions required by installed Expo SDK
10 Thu, 16 Nov 2023 22:21:01 GMT ✓ Check that native modules do not use incompatible support packages
11 Thu, 16 Nov 2023 22:21:01 GMT ✓ Check for legacy global CLI installed locally
12 Thu, 16 Nov 2023 22:21:02 GMT ✓ Check that native modules use compatible support package versions for installed Expo SDK
13 Thu, 16 Nov 2023 22:21:02 GMT Didn't find any issues with the project!

```

Testing App Móvil 2023 (creación Propia)

Figura 51:



Testing App Móvil 2023 – (Creación Propia)

Figura 52:

```

Run expo doctor 6s
1 Tue, 15 Oct 2024 22:45:24 GMT Running "expo doctor"
2 Tue, 15 Oct 2024 22:45:26 GMT Running 14 checks on your project...
3 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check Expo config for common issues
4 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check package.json for common issues
5 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check native tooling versions
6 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check if the project meets version requirements for submission to app stores
7 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check dependencies for packages that should not be installed directly
8 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check for common project setup issues
9 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check for app config fields that may not be synced in a non-CNG project
10 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check npm/ yarn versions
11 Tue, 15 Oct 2024 22:45:26 GMT ✓ Check for issues with metro config
12 Tue, 15 Oct 2024 22:45:27 GMT ✓ Check Expo config (app.json/ app.config.js) schema
13 Tue, 15 Oct 2024 22:45:29 GMT ✓ Check for legacy global CLI installed locally
14 Tue, 15 Oct 2024 22:45:29 GMT ✓ Check that native modules do not use incompatible support packages
15 Tue, 15 Oct 2024 22:45:30 GMT ✓ Check that packages match versions required by installed Expo SDK
16 Tue, 15 Oct 2024 22:45:30 GMT ✓ Check that native modules use compatible support package versions for installed Expo SDK
17 Tue, 15 Oct 2024 22:45:30 GMT Didn't find any issues with the project!

```

Testing App Móvil 2024 (Creación Propia)

Figura 53:



Testing App Móvil 2024 (Creación Propia)

9.2 Normas y Estándares

Estándares de Desarrollo de Software:

ISO/IEC 9126 - Calidad del Software:

- Este estándar internacional define un modelo de calidad del software que incluye características como la funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.
- Se recomienda medir cada característica para garantizar que el software cumpla con los requisitos y expectativas del usuario.

ISO/IEC 25000 - Sistemas y Software de Ingeniería de Software - Calidad del Producto:

- se centra en la calidad del software y sistemas, proporcionando un marco para evaluar y medir características de calidad. Define modelos de calidad, atributos y métricas que facilitan la evaluación objetiva del rendimiento del producto de software.

ISO/IEC 25010 - Calidad del Producto de Software:

- Este estándar amplía el modelo de calidad de ISO/IEC 9126 y proporciona directrices específicas para evaluar la calidad del producto de software.
- Incluye características como la adecuación funcional, eficiencia del rendimiento, compatibilidad, seguridad y usabilidad.

IEEE 730 - Estándar para la Calidad del Proceso de Software:

- Este estándar establece directrices para la gestión de calidad del proceso de software durante el ciclo de vida del proyecto.
- Define procesos para planificación, aseguramiento de la calidad, revisiones y auditorías.

Estándares de Seguridad:

ISO/IEC 27001 - Seguridad de la Información:

- Este estándar establece requisitos para un sistema de gestión de seguridad de la información (SGSI).
- Es crucial para proteger la integridad, confidencialidad y disponibilidad de la información almacenada en la aplicación

ISO/IEC 27002 - Controles de Seguridad de la Información:

1. Complementario a ISO/IEC 27001, este estándar proporciona un conjunto de prácticas recomendadas para la implementación de controles de seguridad.
2. Detalla medidas específicas de seguridad en áreas como la gestión de activos, control de acceso, seguridad física, y criptografía.

ISO/IEC 15408 - Evaluación de la Seguridad en Tecnología de la Información (Common Criteria):

- Conocido como Common Criteria, este estándar internacional establece un marco para evaluar y certificar la seguridad de productos y sistemas de TI.
- Permite evaluar la eficacia de los controles de seguridad implementados, asegurando que los productos cumplan con requisitos específicos de seguridad.

NIST SP 800-53 - Controles de Seguridad y Privacidad:

- Publicado por el Instituto Nacional de Estándares y Tecnología (NIST) de EE. UU., este estándar proporciona un catálogo exhaustivo de controles de seguridad y privacidad para los sistemas de información federales.
- Es ampliamente utilizado como referencia en organizaciones para la gestión de riesgos de seguridad y la implementación de un programa integral de protección de datos.

OWASP Top 10 - Amenazas de Seguridad en Aplicaciones Web:

- Aunque no es un estándar formal, es una referencia esencial para la seguridad de aplicaciones web.
- Proporciona una lista actualizada de las 10 principales amenazas de seguridad y vulnerabilidades en aplicaciones web, y recomendaciones para mitigar estos riesgos.

10 Conclusiones

El proyecto FixWay ha sido una iniciativa exhaustiva que involucra el diseño, desarrollo e implementación de un sistema robusto de gestión para talleres mecánicos. Desde su concepción hasta su fase final, este proyecto ha abarcado una amplia gama de tecnologías, enfoques de diseño y herramientas que, en conjunto, han permitido optimizar las operaciones del taller, mejorar la eficiencia del trabajo y facilitar la toma de decisiones a través del acceso a datos en tiempo real.

Selección de Tecnologías:

La decisión de utilizar Firebase como plataforma backend se fundamenta en su escalabilidad automática, seguridad avanzada y capacidad para manejar datos en tiempo real. Este tipo de infraestructura es crucial para un sistema que requiere gestionar múltiples datos de manera simultánea y accesible para varios usuarios. Comparado con tecnologías más tradicionales como MySQL o PostgreSQL, Firebase sobresale en términos de facilidad de configuración, integración con herramientas modernas y bajo mantenimiento, lo que se alinea con las necesidades de un taller mecánico en crecimiento.

Asimismo, el uso de Tauri para el desarrollo de la aplicación de escritorio es una decisión acertada. Tauri, al ser una plataforma emergente basada en Rust, ofrece una solución ligera, segura y eficiente en cuanto a consumo de recursos. Estas características son cruciales en un entorno donde la velocidad y la estabilidad de la aplicación son esenciales para las operaciones del día a día.

2. Arquitectura y Diseño del Sistema

La arquitectura propuesta en FixWay basada en microservicios facilita la escalabilidad y el mantenimiento. La implementación de diagramas BPMN y modelos de datos detallados garantiza que las operaciones del taller, desde la gestión de inventario hasta la emisión de cotizaciones, se realicen de manera estructurada y eficiente. Este tipo de diseño no solo facilita la comprensión y uso del sistema por parte de los usuarios, sino que también permite a los desarrolladores incorporar nuevas funcionalidades sin comprometer la estabilidad del software.

El uso de tecnologías de frontend como React y Vite demuestra un enfoque moderno y eficiente en el desarrollo de interfaces de usuario. La rapidez de compilación y la capacidad de recargar módulos sin interrupción mejora significativamente la experiencia de desarrollo y, en consecuencia, la calidad del producto final.

3. Implementación de KPI

Los Indicadores Clave de Desempeño (KPI) establecidos en el proyecto son fundamentales para asegurar que el sistema no solo cumple con las expectativas funcionales, sino que también optimiza el rendimiento y la satisfacción del cliente. La inclusión de indicadores como el Tiempo Promedio de Respuesta del Software y la Precisión en la Generación de Cotizaciones son esenciales para monitorear la eficiencia operativa del sistema.

Desde un punto de vista técnico, la implementación de KPI como la Disponibilidad del Sistema y la Tasa de Incidencias Críticas permite al equipo de desarrollo mantener un control estricto sobre el rendimiento del software en escenarios reales. Al medir la estabilidad y la robustez del sistema mediante estos KPI, se garantiza que el software funcione de manera óptima incluso bajo condiciones de alta demanda.

4. Implementación de SLA

Los Acuerdos de Nivel de Servicio (SLA) juegan un papel crucial en la definición de expectativas claras entre los usuarios del sistema y el equipo de desarrollo. Garantizar una disponibilidad del 99.5% y un tiempo de respuesta de soporte técnico de 2 horas son compromisos que refuerzan la confianza del cliente en el sistema y aseguran que las operaciones del taller no se vean afectadas por fallos imprevistos.

Estos acuerdos también reflejan un enfoque proactivo en la gestión de la continuidad del servicio, donde el mantenimiento preventivo y las actualizaciones programadas se realizan de manera transparente, minimizando el impacto en los usuarios. La implementación de medidas de seguridad de datos mediante auditorías y protocolos de respuesta rápida ante incidentes también garantiza que la información sensible esté protegida en todo momento, un aspecto crucial para cualquier sistema que maneje datos de clientes y vehículos.

5. Plan de Pruebas y Aseguramiento de Calidad

El Plan de Pruebas implementado para FixWay, utilizando tanto pruebas de caja negra como de caja blanca, es integral y está diseñado para asegurar que todas las funcionalidades del sistema operen según lo previsto. El uso de herramientas como Jest y Vitest para las pruebas automáticas de componentes y funcionalidades garantiza una cobertura completa del código y la identificación temprana de errores.

Además, la implementación de pruebas de carga, estrés y resistencia ha permitido evaluar la capacidad del sistema para manejar incrementos en la demanda y detectar posibles cuellos de botella. Estas pruebas son fundamentales para asegurar que el sistema mantenga su estabilidad incluso bajo condiciones extremas, garantizando así una experiencia de usuario consistente.

6. Gestión de la Disponibilidad y Continuidad

La gestión de la disponibilidad y continuidad en FixWay ha sido diseñada para maximizar la fiabilidad del sistema, mediante el uso de monitoreo en tiempo real y la implementación de infraestructura redundante. Esto asegura que cualquier fallo en el sistema sea detectado y corregido antes de que afecte significativamente a las operaciones del taller. Además, el plan de recuperación ante desastres asegura que los datos y configuraciones clave del sistema estén siempre respaldados, lo que permite una rápida restauración en caso de fallo crítico.

7. Innovación y Escalabilidad

Una de las fortalezas más destacables de FixWay es su enfoque en la escalabilidad. A medida que más talleres adopten el sistema, la arquitectura está preparada para crecer sin comprometer el rendimiento. Esta flexibilidad es fundamental en un sistema que tiene el potencial de expandirse a medida que se incorporan más usuarios y funciones. El uso de tecnologías como TensorFlow para la predicción de patrones de uso en el inventario también añade un nivel de innovación que mejora la eficiencia y permite anticipar necesidades, lo que posiciona a FixWay como una solución avanzada para la gestión de talleres.

8. Mantenimiento y Evolución del Sistema

El enfoque en el mantenimiento correctivo, preventivo y evolutivo asegura que FixWay no solo funcione adecuadamente en el corto plazo, sino que esté preparado para adaptarse a las demandas cambiantes del sector automotriz. Las actualizaciones periódicas y la inclusión de nuevas funcionalidades basadas en el feedback de los usuarios permiten que el sistema mantenga su relevancia y competitividad en el mercado.

9. Cumplimiento de Estándares y Normas

El compromiso del proyecto con los estándares internacionales de calidad y seguridad, como ISO/IEC 9126 e ISO/IEC 27001, refuerza la solidez del sistema desde el punto de vista técnico y de seguridad de la información. Estos estándares aseguran que FixWay no solo cumple con los requisitos funcionales, sino que también protege adecuadamente los datos sensibles de los usuarios y mantiene un alto nivel de calidad en todas las fases del ciclo de vida del software.

El proyecto FixWay es una solución integral que combina tecnología de vanguardia con prácticas sólidas de ingeniería informática para crear un sistema robusto, escalable y seguro. La implementación de tecnologías modernas como Firebase, Tauri, React, TensorFlow y Gemini, junto con un enfoque riguroso en la gestión de calidad y continuidad del servicio, garantiza que FixWay no solo optimiza las operaciones diarias de los talleres, sino que también ofrece un producto que puede evolucionar y crecer con las demandas del mercado.

Desde un punto de vista de ingeniería, FixWay destaca por su capacidad para integrar tecnologías eficientes, garantizar un alto nivel de servicio y ofrecer una plataforma preparada para el futuro. Los KPI y SLA definidos refuerzan este compromiso con la calidad y la satisfacción del cliente, asegurando que FixWay siga siendo una herramienta clave en la gestión moderna de talleres mecánicos.

11 Bibliografía

- Bright Development. (2023, July 14). React vs. Angular vs. Vue.js: Front-end framework comparison. Bright-development.com. <https://bright-development.com/2023/07/14/choosing-the-right-front-end-framework-react-vs-angular-vs-vue-js/>.
- Dearmer, A. (2023, March 14). Firebase vs. MySQL: Battle of the databases. Integrate.io. <https://www.integrate.io/blog/firebase-vs-mysql/>.
- Firebase vs MySQL: 13 critical differences. Data Stack Hub. Retrieved October 22, 2024, from <https://www.datastackhub.com/compare/firebase-vs-mysql/>.
- Poley, C. Angular vs React vs Vue: Detailed framework comparison. Blog by Tiny. Retrieved October 22, 2024, from <https://www.tiny.cloud/blog/vue-react-angular-js-framework-comparison/>.
- Saravanan, N. (2019, January 25). Escalado de base de datos: escalado horizontal y vertical. Hackernoon.com. <https://hackernoon.com/lang/es/escalado-de-base-de-datos-escala-horizontal-y-vertical-85edd2fd9944>.
- Vite. Vitejs. Retrieved October 22, 2024, from <https://vite.dev/>.
- Software DELSOL. (2020, December 9) Estándar ¿Qué es? <https://www.sdelisol.com/glosario/estandar/#:~:text=%C2%BFQu%C3%A9%20es%20est%C3%A1ndar%3F,rentable%20o%20no%20lo%20es>.
- Thompson, F. Cómo armar una verdadera arquitectura tecnológica. <https://cio.com.mx/armar-una-verdadera-arquitectura-tecnologica/>.
- ¿Qué es una arquitectura de aplicaciones? Tipos de arquitecturas de aplicaciones. <https://www.redhat.com/es/topics/cloud-native-apps/what-is-an-application-architecture>.
- Euroinnova Formación. (2022). actividades que se realizan en un taller mecánico automotriz. Euroinnova Business School. <https://www.euroinnova.cl/blog/actividades-que-se-realizan-en-un-taller-mecanico-automotriz>.
- ¿Qué es el modelo V en el desarrollo de software? Aativ. <https://www.aativ.com/es/tendencias/art%C3%ADculo/que-es-el-modelo-v-en-el-desarrollo-de-software#:~:text=El%20modelo%20V%20o%20ciclo,simb%C3%B3lica%20del%20flujo%20de%20desarrollo>.
- alfonros consultores. (2015, January 20). Cuadro de Mando Integral - Alfonso Ros Consultores. Alfonso Ros Consultores. <https://alfonros.com/cuadro-de-mando-integral/>.

- ¿Qué es son la Integración y el Despliegue Continuo? Aptiv. <https://www.aptiv.com/es/tendencias/art%C3%ADculo/qu%C3%A9-es-son-la-integraci%C3%B3n-y-el-despliegue-continuo>.
- Asana, T. (2023, April 24). Qué es un organigrama y cómo hacerlo con plantillas gratis [2023] • Asana. Asana. <https://asana.com/es/resources/organizational-chart>.
- Admin. (2021). Testing en Modelo V. Quassure. <https://quassure.com/testing-en-modelo-v/>.
- Fernandes, A. Z. (2022). Marco teórico. Significados. <https://www.significados.com/marco-teorico/>.
- Huet, P. (2023, April 13). Arquitectura de software: Qué es y qué tipos existen. OpenWebinars.net. <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>.
- Pursell, S. (2021, December 8). Sistemas de información en las empresas: tipos, funciones y ejemplos. HubSpot. Retrieved October 8, 2023, from <https://blog.hubspot.es/marketing/sistema-informacion#:~:text=Un%20sistema%20de%20informaci%C3%B3n%20es,de%20procesos%20y%20operaciones%20internas>.
- Martinekuan. Modelos de diseño de microservicios - Azure Architecture Center. Microsoft Learn. <https://learn.microsoft.com/es-es/azure/architecture/microservices/design/patterns>.
- Arquitectura de software: 5 patrones principales. (2023, March 10). Inesdi. <https://www.inesdi.com/blog/arquitectura-de-software-5-patrones-principales/>.
- ¿Qué son los microservicios? | AWS. Amazon Web Services, Inc. <https://aws.amazon.com/es/microservices/>.
- Atlassian. Comparación entre la arquitectura monolítica y la arquitectura de microservicios. <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>.
- Novoseltseva, E. (2021, June 9). 5 principales patrones de Arquitectura de Software. Apiumhub. <https://apiumhub.com/es/tech-blog-barcelona/principales-patrones-arquitectura-software/>.
- Cómo diseñar una arquitectura de software: consejos y prácticas recomendadas. (2021, April 8). Lucidchart. <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software>.
- Evaluación de la Arquitectura de Software. SG Buzz. <https://sg.com.mx/revista/31/arquitectura-evaluacion-la-arquitectura-software>.

- Definiciones y Conceptos sobre Evaluación de Arquitecturas - ` - Arquitectura de Gobierno. https://centroderecursos.agesic.gub.uy/web/arquitectura-de-gobierno/arquitectura-para-tramites/-/wiki/Arquitectura+para+Tr%C3%A1mites/Definiciones+y+Conceptos+sobre+Evaluaci%C3%B3n+de+Arquitecturas/pop_up.
- EcuRed. Técnicas de Evaluación de Arquitectura de Software - EcuRed. https://www.ecured.cu/T%C3%A9nicas_de_Evaluaci%C3%B3n_de_Arquitectura_de_Software.
- Software DELSOL. (2020, December 9) Estándar ¿Qué es? <https://www.sdelsol.com/glosario/estandar/#:~:text=%C2%BFQu%C3%A9%20es%20est%C3%A1ndar%3F,rentable%20o%20no%20lo%20es>.
- Thompson, F. Cómo armar una verdadera arquitectura tecnológica. <https://cio.com.mx/armar-una-verdadera-arquitectura-tecnologica/>.
- ¿Qué es una arquitectura de aplicaciones? Tipos de arquitecturas de aplicaciones. <https://www.redhat.com/es/topics/cloud-native-apps/what-is-an-application-architecture>.
- Euroinnova Formación. (2022). actividades que se realizan en un taller mecánico automotriz. Euroinnova Business School. <https://www.euroinnova.cl/blog/actividades-que-se-realizan-en-un-taller-mecanico-automotriz>.
- ¿Qué es el modelo V en el desarrollo de software? Aptiv. <https://www.aptiv.com/es/tendencias/art%C3%ADculo/que-es-el-modelo-v-en-el-desarrollo-de-software#:~:text=El%20modelo%20V%20o%20ciclo,simb%C3%B3lica%20del%20flujo%20de%20desarrollo>.
- alfonros consultores. (2015, January 20). Cuadro de Mando Integral - Alfonso Ros Consultores. Alfonso Ros Consultores. <https://alfonros.com/cuadro-de-mando-integral/>.
- ¿Qué es son la Integración y el Despliegue Continuo? Aptiv. <https://www.aptiv.com/es/tendencias/art%C3%ADculo/qu%C3%A9-es-son-la-integraci%C3%B3n-y-el-despliegue-continuo>.
- Asana, T. (2023, April 24). Qué es un organigrama y cómo hacerlo con plantillas gratis [2023] • Asana. Asana. <https://asana.com/es/resources/organizational-chart>.
- Admin. (2021). Testing en Modelo V. Quassure. <https://quassure.com/testing-en-modelo-v/>.
- Fernandes, A. Z. (2022). Marco teórico. Significados. <https://www.significados.com/marco-teorico/>.

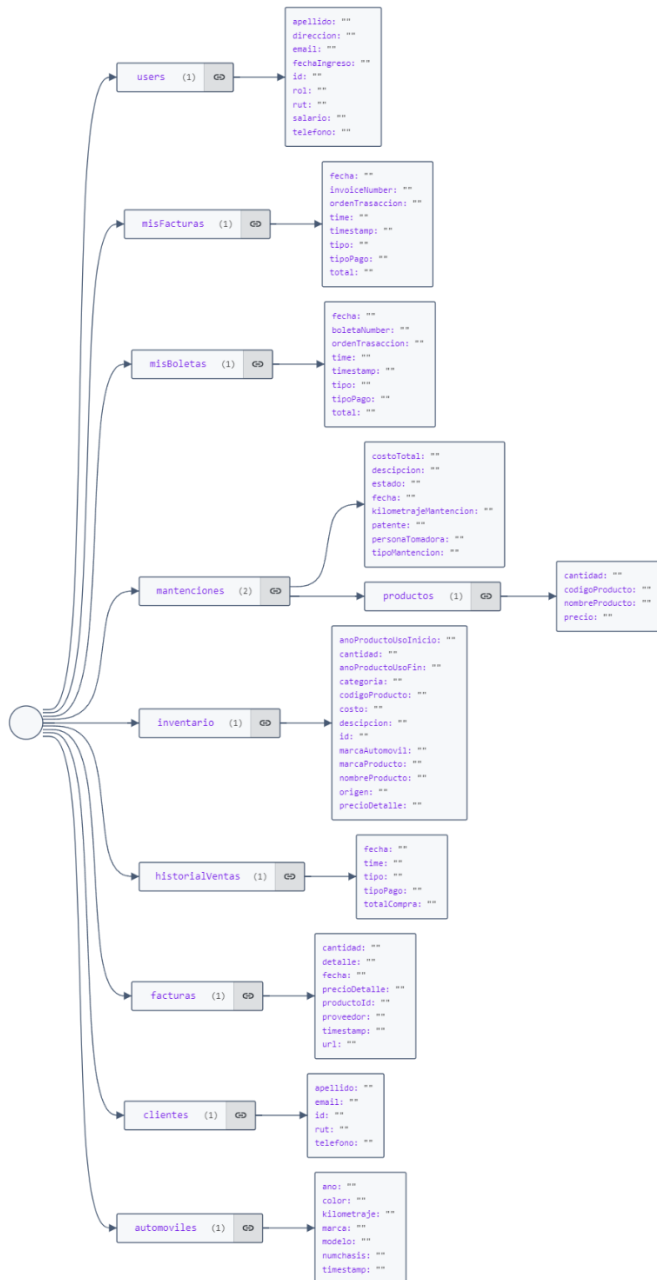
- Huet, P. (2023, April 13). Arquitectura de software: Qué es y qué tipos existen. OpenWebinars.net. <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>.
-
-
- Pursell, S. (2021, December 8). Sistemas de información en las empresas: tipos, funciones y ejemplos. HubSpot. Retrieved October 8, 2023, from <https://blog.hubspot.es/marketing/sistema-informacion#:~:text=Un%20sistema%20de%20informaci%C3%B3n%20es,de%20procesos%20y%20operaciones%20internas.>
- Martinekuan. Modelos de diseño de microservicios - Azure Architecture Center. Microsoft Learn. [https://learn.microsoft.com/es-](https://learn.microsoft.com/es-es/azure/architecture/microservices/design/patterns)
- [es/azure/architecture/microservices/design/patterns.](https://learn.microsoft.com/es-es/azure/architecture/microservices/design/patterns)
- Arquitectura de software: 5 patrones principales. (2023, March 10). Inesdi. <https://www.inesdi.com/blog/arquitectura-de-software-5-patrones-principales/>.
- ¿Qué son los microservicios? | AWS. Amazon Web Services, Inc. <https://aws.amazon.com/es/microservices/>.
- Atlassian. Comparación entre la arquitectura monolítica y la arquitectura de microservicios. [https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith.](https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith)
- Novoseltseva, E. (2021, June 9). 5 principales patrones de Arquitectura de Software. Apiumhub. <https://apiumhub.com/es/tech-blog-barcelona/principales-patrones-arquitectura-software/>.
- Cómo diseñar una arquitectura de software: consejos y prácticas recomendadas. (2021, April 8). Lucidchart. [https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software.](https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software)
- Evaluación de la Arquitectura de Software. SG Buzz. [https://sg.com.mx/revista/31/arquitectura-evaluacion-la-arquitectura-software.](https://sg.com.mx/revista/31/arquitectura-evaluacion-la-arquitectura-software)
- Definiciones y Conceptos sobre Evaluación de Arquitecturas - ` - Arquitectura de Gobierno. [https://centroderecursos.agesic.gub.uy/web/arquitectura-de-gobierno/arquitectura-para-tramites/-](https://centroderecursos.agesic.gub.uy/web/arquitectura-de-gobierno/arquitectura-para-tramites/)
- [/wiki/Arquitectura+para+Tr%C3%A1mites/Definiciones+y+Conceptos+sobre+Evaluaci%C3%B3n+de+Arquitecturas/pop_up.](https://centroderecursos.agesic.gub.uy/web/arquitectura-de-gobierno/arquitectura-para-tramites/)

- EcuRed. Técnicas de Evaluación de Arquitectura de Software - EcuRed.

https://www.ecured.cu/T%C3%A9nicas_de_Evaluaci%C3%B3n_de_Arquitectura_de_Software.

12 Anexos

Figura 54:



Estructura de base de datos (Creación Propia)