

**PROTOTIPO PARA EL REGISTRO DE REPARACIONES DE LA EMPRESA “SDMOTOS &
REPARAUTOSRG”**

ARREDONDO GARCIA CRISTIAN FELIPE

**INSTITUCIÓN UNIVERSITARIA PASCUAL
BRAVO FACULTAD DE INGENIERÍA
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN
2022**

**PROTOTIPO PARA EL REGISTRO DE REPARACIONES DE LA EMPRESA “SDMOTOS &
REPARAUTOSRG”**

ARREDONDO GARCIA CRISTIAN FELIPE

**TRABAJO PARA OPTAR AL TÍTULO DE TECNOLOGIA EN DESARROLLO
DE SOFTWARE**

TUTORES:

**INGENIERO JAVIER ALBERTO SALDARRIAGA CANO
INGENIERA YUDY ANDREA QUINTERO TANGARIFE**

**INSTITUCIÓN UNIVERSITARIA PASCUAL
BRAVO FACULTAD DE INGENIERÍA
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
MEDELLÍN**

2022

Contenido

Contenido	3
Glosario	5
Resumen	6
Introducción	7
Capítulo 1: El problema	8
Capítulo 2: Justificación	9
Capítulo 3: Objetivos	10
<i>Objetivo general</i>	10
<i>Objetivos específicos</i>	10
Capítulo 4: Referentes teóricos	11
<i>Work Orders</i>	12
<i>Mitchell 1</i>	13
<i>Kordata</i>	14
Capítulo 5: Metodología	16
<i>Ciclo de vida del proyecto informático</i>	16
<i>Análisis</i>	16
<i>Diseño</i>	16
<i>Implementación:</i>	17
<i>Pruebas</i>	17
Capítulo 6: Resultados	18
<i>Los Requerimientos:</i>	18
<i>Requerimientos Funcionales:</i>	18
Tabla 1	18
<i>Requerimientos no Funcionales:</i>	20
<i>Diagramas:</i>	21
<i>Diagrama de casos de uso:</i>	21
<i>Diagrama de clases:</i>	29

<i>Diagrama de secuencia:</i>	31
<i>Modelos:</i>	33
<i>Modelo relacional:</i>	33
<i>Mockups</i>	34
Capítulo 7: Recomendaciones	39
Capítulo 8: Conclusiones	40
Bibliografía	41

Glosario

Aplicación: Programa informático que sirve como herramienta para permitir a un Usuario realizar una o varias tareas. Normalmente sirve para facilitarnos las cosas.

Diseñar: Boceto, bosquejo o esquema que involucra variadas dimensiones que van más allá del aspecto, la forma y el color, abarcando también la función de un objeto y su interacción con el usuario.

Herramienta: Instrumento diseñado para facilitar la realización de una tarea específica

Proyecto: Es la acción de pedir algo que se desea o se requiere, esto se hace comunicando una pretensión o una necesidad a alguien o algo, ese alguien o algo se encargará de satisfacerla.

Software: programa o varios programas como datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.

Usuarios: personas que utilizan una computadora, tablet, celular, entre otros dispositivos para acceder a productos o servicios.

Desarrollo: se refiere al proceso de cambio y crecimiento relacionado con una situación, individuo u objeto

Reparaciones: arreglo de una cosa estropeada, rota o en mal estado.

React.js: es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

Spring Boot: es una tecnología para crear aplicaciones autocontenidas.

MySQL: es un sistema de gestión de bases de datos relacional

Servicio: es un conjunto de actividades que buscan satisfacer las necesidades de un cliente.

Cliente: es la persona o empresa receptora de un bien, servicio, producto o idea, a cambio de dinero u otro artículo de valor.

Mockups: es una maqueta o es un modelo a escala o de tamaño completo de un diseño o dispositivo, que se utiliza para la enseñanza, demostración, evaluación del diseño, promoción y otros fines.

Requerimientos Funcionales: define una función del sistema de software o sus componentes.

Diagrama de clases: es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones, y las relaciones entre los objetos

Diagrama de Entidad Relación: es una herramienta para el modelo de datos, la cual facilita la representación de entidades de una base de datos.

Frontend: consiste en la conversión de datos en una interfaz gráfica para que el usuario pueda ver e interactuar con la información de forma digital usando HTML, CSS y JavaScript.

JavaScript: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.

Java: es un lenguaje de programación y una plataforma informática.

Backend: es un sistema corporativo que se utiliza para dirigir una web o empresa, tales como sistemas de gestión de pedidos, inventario y procesamiento de suministro

Resumen

La siguiente propuesta de mejoramiento se realizó un diseño para la recolección de registros dentro del taller “SDMOTO & REPARAAUTOS”. Esta empresa se dedica al sector automotor donde realiza reparaciones a vehículos como motos, camiones y carros. Para realizar sus registros de reparaciones diarios donde se administran mediante un cuaderno en donde podemos observar y llevan un control de reparaciones que se ejecutan diariamente.

Este procedimiento actual que tiene la empresa de administrar sus registros de reparaciones deja fallas tales como un descuadre a la hora de ver cuántos vehículos se repararon al mes fin, esto conlleva a la idea de crear un software que permite guardar dicha información y monitorear estos tipos de datos.

Introducción

En los últimos años los desarrollos tecnológicos han crecido considerablemente permitiendo que algunas tareas de la vida cotidiana se realicen de manera más eficiente, logrando un ahorro de tiempo y facilitando la vida de las personas.

Este proyecto de grado presenta una propuesta de software que tiene como propósito ofrecer comodidad al usuario permitiéndole llevar un registro de las reparaciones de los vehículos que ingresen al local diariamente, evitando errores tales como no haberlos registrados previamente o que se pierda estos registros a través del tiempo.

El software consta de un aplicativo web donde tendrá solo un único perfil que será el administrador y éste podrá registrar las reparaciones. Para poder realizar lo mencionado anteriormente, se partió de la creación de un conjunto de requerimientos, se propusieron los diferentes diagramas que describen el software y se muestra cómo sería la estructura de base datos, todo esto para dar una idea cómo funciona la aplicación, que información debería almacenar y cómo estaría conectadas las tablas para establecer las dependencias entre estas.

El presente trabajo está estructurado de la siguiente manera: en los capítulos uno y dos se enuncia y se justifica la problemática, seguidamente se muestra los objetivos y se presentan los referentes teóricos y la metodología. El capítulo 6, presentación de resultados, corresponde a la descripción completa del software propuesto. Finalmente se presentan las recomendaciones para la implementación de la solución de software y las conclusiones del trabajo desarrollado.

Capítulo 1: El problema

Con el transcurrir del tiempo el sector automotor ha incrementado mucho, el cual hace apuesta a los avances tecnológicos.

Para el presente estudio se ha realizado un análisis de los principales problemas de “SDMOTO & REPARAAUTOSRG”. La información fue recopilada con base en entrevista con un empleado

La empresa repara alrededor de 20 a 35 vehículos al día y en temporadas altas como: diciembre, enero, semana santa, junio y julio, el flujo de personas es muy alto por lo que es inevitable el desorden y mala facturación de las reparaciones. A lo cual muchos usuarios optan por llevar su vehículo a otro taller, con la App web se busca evitar dicho percance. Esto garantiza una mejor organización y un mejor desempeño del empleado.

Se pudo evidenciar que la empresa lidia con el problema de llevar un registro de sus servicios. Su situación actual es que cuenta con múltiples archivos de Excel donde llevan los registros de cada servicio. La empresa ha identificado el problema generado ya que los empleados mantienen un desorden con dichos archivos.

Con base en lo anteriormente expuesto, la empresa es consciente de la oportunidad de mejora que generaría al manejar estos registros de otra manera

Capítulo 2: Justificación

La finalidad del proyecto es llevar un control adecuado de los registros por medio de una App amigable y de una forma rápida, segura, confiable y sencilla.

Esta App sería de gran impacto para la empresa por el gran auge tecnológico que se vive en la actualidad, ya que los usuarios dispondrán de una herramienta que puede utilizar en cualquier momento, la cual le permitirá tener la seguridad de un registro.

La importancia de esta aplicación es la eficiencia de cómo administrar, guardar y dar una seguridad a los registros. El impacto de esta es reducir los errores humanos y mejorar el rendimiento laboral a la hora de hacer los registros

Los beneficios para la empresa:

- Registro histórico
- Seguridad
- Comodidad
- Eficiencia
- Estar a la vanguardia en el uso de las nuevas tecnologías

Capítulo 3: Objetivos

Objetivo general

Proponer una solución de tipo prototipo de software para la administración de los registros de reparaciones de los vehículos de la empresa “SDMOTO & REPARAAUTOS” con el fin de favorecer el registro de clientes, vehículos ingresados y reparaciones que se realizan en este taller.

Objetivos específicos

Construir los requerimientos funcionales y no funcionales para la creación del contenido que tendrá el diseño del software de registros.

Construir el diagrama de caso de uso, diagrama de clases y el diagrama de secuencia para la propuesta de software.

Proponer la estructura de la base de datos con sus respectivos modelos: relacional y entidad relación.

Realizar el Mockup de la interfaz de usuario para mostrar cómo será la interacción y navegación a través de este

Capítulo 4: Referentes teóricos

La tecnología avanza cada día, haciendo que esta brinde solución a los problemas de la vida cotidiana, haciéndonos más fácil la vida. Internet es una red que conecta a otras redes y dispositivos para compartir información por medio de páginas, sitios, aplicaciones. El internet es algo muy popular hoy en día, incluso en algunas partes del mundo es considerado necesidad para el ser humano y es algo que la mayoría de las personas usan cada día ya que esto les vuelve la vida mucho más fácil y más entretenida, para algunos ya que contiene información de todo tipo, puede conectar varios lugares y personas, así están lejos y puede ser algo para todo tipo de personas. Gracias al internet podemos hacer funcionar aplicaciones webs.

Una aplicación web se puede acceder a desde cualquier dispositivo con acceso a internet y usada en smartphone (teléfono inteligente), tablet o computador. Estas aplicaciones webs se pueden acceder desde cualquier sistema operativo con un navegador web actual

Hay varias herramientas para crear este tipo de aplicaciones webs, en este caso mencionaremos el framework Spring Boot este se utiliza para desarrollar micro servicios o consumir servicios expuestos y se escribe en el lenguaje java. También para consumir estos servicios utilizaremos una librería llamada React utilizada para hacer toda la maquetación y la interactividad que se le va a presentar al usuario.

Hay varios entornos de desarrollo para crear este tipo de webs. En este caso mencionaremos las herramientas IntelliJ y Visual Studio Code para desarrollar.

IntelliJ y Visual Studio Code sirven para programar aplicaciones y páginas web para Windows y macOS. En IntelliJ se puede programar en los lenguajes de programación como Java, Kotlin, Groovy, Scala, Javascript, al igual que entornos de desarrollo web. Lo que se puede hacer en este IDEA con todo el entorno de programación es editar, diseñar, depurar y compilar.

Un programa de diseño gráfico es aquel que nos permite retocar o modificar imágenes, realizar dibujos, y guardarlos después en distintos formatos. Uno de estos programas es Figma, una herramienta de diseño vectorial, permitiéndonos producir diseños a Sketch sin limitaciones ni gastos.

Para la creación de un sistema de software se necesita tener requerimientos, estos requerimientos se clasifican en funcionales y no funcionales. Un requerimiento funcional es la descripción de actividades y servicios presentados, que el software deberá tener y hacer, son los requisitos que se refieren a las entradas y

salidas de todo el proceso, en cambio los requerimientos no funcionales son las expresan características o servicios externos o individuales del sistema, se trata de cómo se hace el sistema.

Un diagrama de casos de uso representa procesos y sistemas orientados a objetos UML no es un lenguaje de programación, sino un lenguaje de modelado, es decir, un método estandarizado para representar sistemas planificados o ya existentes. En este diagrama, todos los objetos involucrados se estructuran y se relacionan entre sí. Un diagrama de secuencia es un sistema UML mucho más interactivo haciendo énfasis a la hora de intercambiar objetos, este se realiza con ejes verticales y horizontales, donde los verticales representan el tiempo y los horizontales representan los objetos.

MySQL es una plataforma para el manejo de bases de datos que trabaja con Microsoft como su nombre lo indica. SQL permite crear una gráfica de la base de datos llamada diagrama de base de datos. Esta representación se denomina diagrama de la base de datos, este diagrama de estructura lógica de la base de datos donde se puede observar las tablas de la base de datos con sus nombres y composición, en la parte superior, y en la parte inferior se observan las acciones que va a tener cada tabla como, por ejemplo, que se pueda editar algún dato, modificar, eliminar, entre otras. Cada tabla contiene una clave primaria, las tablas son conectadas entre sí por medio de una clave foránea para indicar que los datos de una pertenecen a otra y se hace una relación de uno a muchos de muchos a muchos o de uno a uno dependiendo de la relación, ejemplo, si tenemos una aplicación de compra se puede hacer una relación entre dos tablas una llamada producto y otra llamada cliente, la relación entre estas dos tablas son de muchos a muchos ya que muchos clientes pueden comprar muchos productos y muchos productos pueden ser llevados por muchos clientes.

StarUML es una herramienta de modelado de software realizado con UML, este sirve para la realización de diagramas de uso, diagramas de clases, diagrama de secuencia, diagrama de colaboración, diagramas de estados, diagramas de actividad, diagramas de componentes, diagramas de despliegues y diagramas de composición estructural.

Work Orders

Este software está dirigido a proporcionar servicios de gestión de calidad para un taller de automóviles. El trabajo de administración consiste en mantener un registro de todos los archivos relacionados con el negocio.

Tiene un amplio espacio para almacenar datos además de que guía al personal del taller a buscar fácilmente el archivo relevante a través de un motor de búsqueda intuitivo.

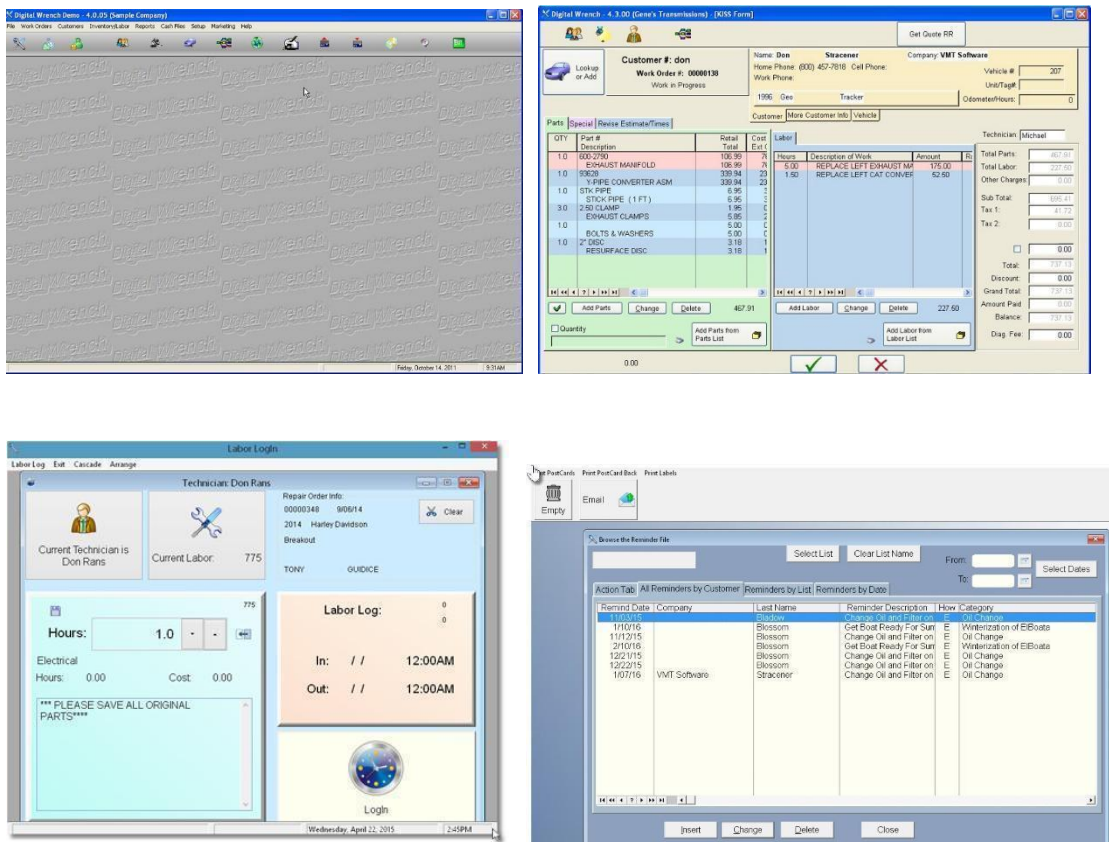


Figura 1. App Work Orders

Mitchell 1

Este software tiene herramientas fáciles de usar para manejar los problemas relacionados con el mantenimiento y la administración de los talleres de automóviles.

El software es útil para guardar los datos deseados y para obtener un detalle completo del trabajo de todo el día.

Mitchell 1 se está utilizando en grandes y pequeños talleres de automóviles y está dando los resultados deseados.

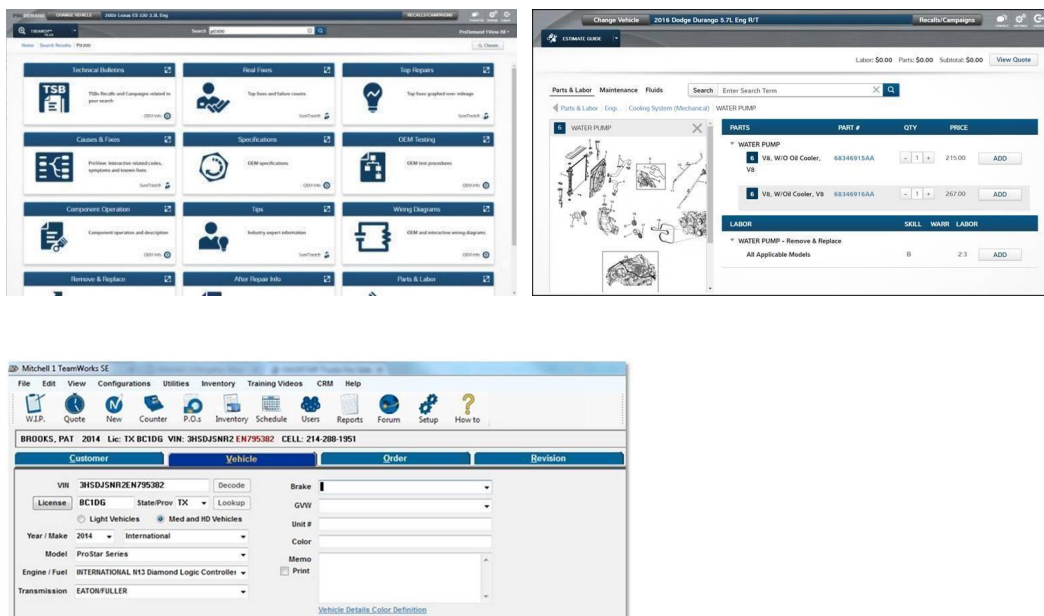


Figura 2. Mitchell 1

Kordata

Es el software más utilizado para la gestión administrativa de los talleres mecánicos, es de fácil acceso a través de una página web o aplicación móvil disponible en la App Store y en Play Store, y provee todo lo que un taller mecánico requiere en su operación diaria.

Toda gira a través de la orden de servicio, desde ahí Kordata toma el control y ayuda a los administradores y dueños a llevar una gestión correcta de la operación del día a día, desde la recepción de los vehículos, toma de evidencia fotográfica, control del inventario de las refacciones, costeo de los insumos, cierre, facturación y cobro de la orden de servicio.

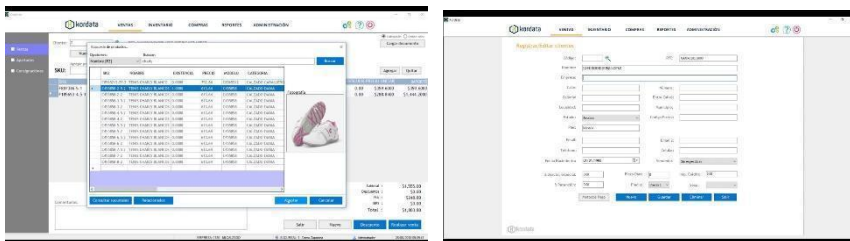


Figura 3. Kordata

Capítulo 5: Metodología

El ciclo de vida del desarrollo de software describe las fases de un proyecto informático desde el inicio hasta el final, con el propósito de definir las distintas fases que se va a tener durante el desarrollo de este para garantizar que el software cumpla los requisitos para la aplicación y verificaciones de los procedimientos del desarrollo.

Análisis:

Se recopiló información del negocio y sus procesos mediante los empleados del mismo, con preguntas realizadas a cada uno de ellos y algunos clientes que se encontraban en el lugar. Luego de recopilar la información se procedió a relacionarla para encontrar patrones, casos relevantes y posibles mejoras a implementar, con estos datos se llegó a plasmar el análisis de esta empresa y sus procesos principales y secundarios que conforman la totalidad de funciones.

Diseño:

Al realizar el análisis se realizó un diseño de las interfaces o mockups por medio de una herramienta de trabajo llamado figma, allí se plasmó las ideas con sus conceptos de negocio y demás factores que permitieran para el usuario un correcto enlace de comunicación con el sistema, que fuese de fácil acceso, trazabilidad y usabilidad.

En cuanto a los componentes y demás factores como los casos de uso, diagrama de secuencia y diagrama de clases, se diseñaron con la ayuda de una herramienta llamada StarUML.

Implementación:

Para la implementación del proyecto se inició con la base de datos, la cual se construyó en Mysql, se crearon las tablas y atributos de cada una de ellas con sus respectivas relaciones, luego se creó el backend de manera limpia y por componentes. El cual se implementó en el framework llamado spring boot con java 17, luego se procedió a crear el front end en el framework llamado react, creando las interfaces de usuario y componentes necesarios. Finalmente se unificó consumiendo los servicios puestos desde en el back end y su respectiva implementación.

Pruebas:

Las pruebas se realizaron al aplicativo web como pruebas funcionales es decir se validó que cumplieran con los requerimientos y métricas establecidas para el correcto funcionamiento del proyecto, se encontró en la realización de ellas un positivo resultado en caso de uso definido, cumpliendo lo anteriormente mencionado se da el visto bueno al desarrollo.

Capítulo 6: Resultados

Los Requerimientos:

Requerimientos Funcionales:

La figura 4 muestra los requerimientos funcionales, enunciado las operaciones que permitiera realizar el sistema, se muestra información que se necesita para ello y que actor ejecutara ese requerimiento.

Tabla 1 - Requerimientos funcionales

ID Requerimiento	Nombre	Descripción	Información Necesaria	Actor
RF-001	Login Usuario	El sistema debe permitir el ingreso por medio de un usuario y contraseña para acceder.	Usuario y contraseña	Usuarios que participen
RF-002	Registrar cliente	El sistema debe permitir registrar los clientes que lleguen al taller.	Documento, numero de contacto, email, apellido, nombre	Empleado
RF-003	Eliminar cliente	El sistema debe permitir eliminar el cliente que llegue al taller.	Documento, numero de contacto, email, apellido, nombre	Empleado
RF-004	Actualizar cliente	El sistema debe permitir actualizar el cliente.	Documento, numero de contacto, email, apellido, nombre	Empleado
RF-005	Registrar vehículo	El sistema debe permitir registrar las el	Modelo, tipo de vehículo, id(placas)	Empleado

		cliente que llegue al taller.		
RF-006	Eliminar vehículo	El sistema debe permitir eliminar el vehículo.	Modelo, tipo de vehículo, id(placas)	Empleado
RF-007	Actualizar vehículo	El sistema debe permitir actualizar el vehículo.	Modelo, tipo de vehículo, id(placas)	Empleado
RF-008	Almacenar	El sistema debe guardar la información de los clientes, vehículos y reparaciones, en una base de datos	documento, numero de contacto, email, apellido, nombre, Modelo, tipo de vehículo, id(placas), Observaciones, problema	Software y Base de Datos
RF-009	Registrar Reparación	El sistema debe permitir registrar las reparaciones.	Observaciones, problema, id (placas del vehículo), Fecha	Empleado
RF-010	Eliminar Reparación	El sistema debe permitir eliminar la reparación.	Observaciones, problema, id (placas del vehículo), Fecha	Empleado
RF-011	Actualizar Reparación	El sistema debe permitir actualizar la reparación.	Observaciones, problema, id (placas del vehículo), Fecha	Empleado

Requerimientos no Funcionales:

En la figura 5 se muestra los Requerimientos no funcionales con su descripción de lo que puede o no hacer el sistema, estos no están directamente relacionados con las funciones que realiza el sistema, pero permite saber si es fiable el diseño, que tiempo de respuesta podría tener, en donde funcionarían, las entradas y salidas, y la representación gráfica de imágenes de los datos que se va a utilizar en la interfaz de este sistema

Tabla 2. - Requerimientos no funcionales

ID Requerimiento	Nombre	Descripción
RF-001	Internet	La aplicación solo es funcional con internet y por eso los usuarios deben tener acceso a internet
RF-002	Dispositivo	Se debe disponer de un computador de escritorio o portátil, para el uso adecuado del sistema.
RF-003	Responsiva	El sistema se deberá ajustar a las resoluciones de los dispositivos
RF-004	Navegador	La aplicación web debe verse en cualquier navegador moderno.

Diagramas:

Los diagramas que se verán a continuación simplifican la información del funcionamiento del sistema de manera que más adelante se puedan realizar la implementación, desarrollo y puesta en marcha de este. Tales diseños brindan un entendimiento de cómo va conectado todo en conjunto, que es lo que contiene y cuáles son las funciones que se pueden hacer a quienes deben hacerlas, en el sistema.

Diagrama de casos de uso:

En la figura 6 se observa el Diagrama de Casos de uso y con él podemos ver la comunicación que hay entre el usuario y el sistema. Todo esto para saber cuál debería ser el punto de partida y el eje de apoyo del desarrollo del proceso diseño y análisis.

Figura 4

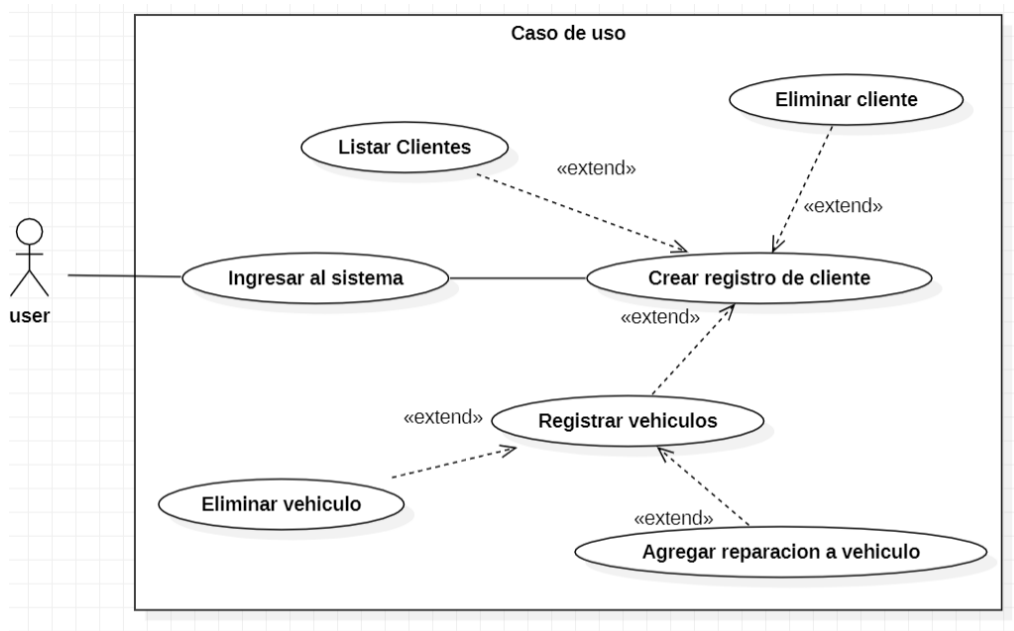


Figura 5

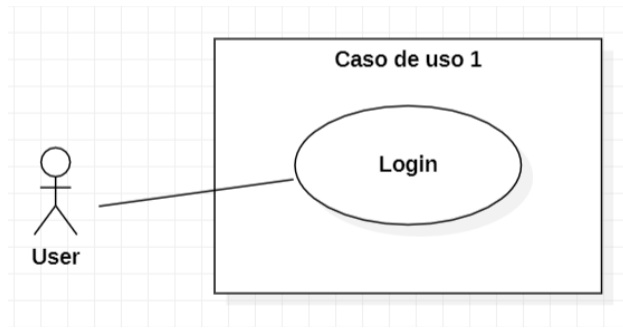


Tabla 3 - Caso de Uso CU01 (Login)

ID- UC	CU01		
Nombre de UC:	LOGIN		
Creado por:	Desarrollador	Actualizado por:	
Fecha de la creación:		Fecha de la última Actualización:	

Actores:	Usuario	
Descripción:	El usuario podrá realizar la autenticación en la aplicación.	
Disparador:	En la página principal (login) el usuario ingresa las credenciales y le da clic al botón ingresar, el sistema deberá autenticar al usuario que intenta ingresar al sistema.	
Pre-Condiciones:	El usuario debe llenar los campos que se requieren para la autenticación.	
Pos-Condiciones	El servicio responderá con una respuesta.	
Flujo Normal:	1	En este caso de uso empieza en la página principal el usuario ingresa las credenciales y le hace click en el botón de ingresar.
	2	Ingresa la petición al servicio y se responde a la petición realizada.
	3	El sistema valida que todos los datos del usuario necesarios para realizar el registro del cliente

	4	El sistema valida la respuesta que todo esté bien.
Flujo Alternativo:	5	Error al realizar la petición.
	6	Error al no enviar campos requeridos.
Excepciones:	El usuario solo podrá.	
Inclusiones:	N/A	
Prioridad:	Importante	
Frecuencia de uso:	Siempre	
Requisitos especiales:	N/A	
Notas:	N/A	

Figura 6

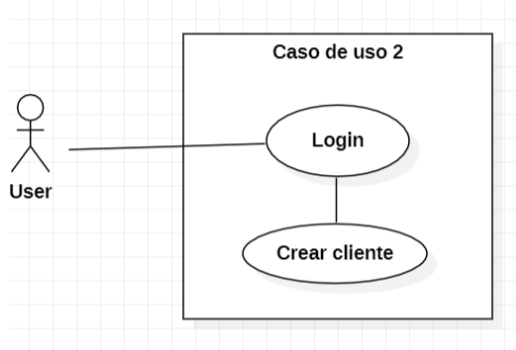


Tabla 4 - Caso de Uso CU02 (Creación de registro de usuario)

ID- UC	CU02		
Nombre de UC:	Creación de registro de cliente		
Creado por:	Desarrollador	Actualizado por:	
Fecha de la creación:		Fecha de la última Actualización:	

Actores:	Usuario	
Descripción:	El usuario(empleado) podrá realizar el registro del cliente en el sistema.	
Disparador:	En la pantalla de registro usuario, se debe realizar el registro de un cliente, se solicitan ingresar ciertos datos que se requieren para el registro.	
Pre-Condiciones:	El usuario debe llenar los campos que se solicitan para realizar el registro.	
Pos-Condiciones	El servicio responderá con una respuesta.	
Flujo Normal:	1	En este caso de uso empieza en la página principal el usuario ingresa las credenciales y le hace click en el botón de guardar.
	2	Ingresa la petición al servicio
	3	El sistema valida datos y devuelve respuesta.
	4	El sistema da una alerta al usuario.
Flujo Alternativo:	5	Error al realizar la petición.
	6	Error al no poder guardar el cliente.
Excepciones:	El usuario no podrá registrar al cliente solo podrá ingresar con el usuario correcto.	
Inclusiones:	N/A	
Prioridad:	Importante	
Frecuencia de uso:	Cada vez que se requiera registrar un nuevo cliente	
Requisitos especiales:	N/A	
Notas:	N/A	

Figura 7

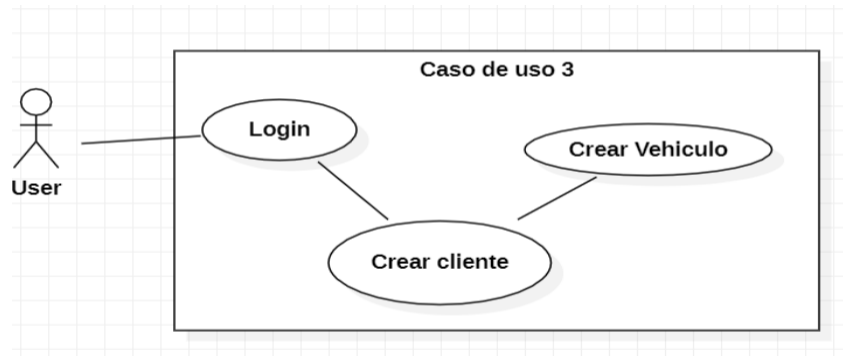


Tabla 5 - Caso de Uso CU03 (Creación de registro de vehículo)

ID- UC	CU03		
Nombre de UC:	Creación de registro de vehículo		
Creado por:	Desarrollador	Actualizado por:	
Fecha de la creación:		Fecha de la última Actualización:	

Actores:	Usuario	
Descripción:	El usuario(empleador) podrá realizar el registro de un vehículo en el sistema asociado a un cliente.	
Disparador:	En la pantalla de registro de un vehículo se solicitan algunos datos básicos, para así llevar el control en el sistema.	
Pre-Condiciones:	El usuario debe llenar los campos que se requieren para realizar el registro del vehículo.	
Pos-Condiciones	El servicio responderá con una respuesta.	
Flujo Normal:	1	Este caso de uso empieza en la pantalla de registro del vehículo, el usuario ingresa los datos solicitados en el formulario y le hace click en el botón de guardar.
	2	Ingresa la petición al servicio.
	3	El sistema valida la respuesta.

	4	El sistema da una alerta al usuario.
Flujo Alternativo:	5	Error al realizar la petición.
	6	Error al no poder guardar el vehículo.
Excepciones:	El usuario no podrá registrar el vehículo si le hace falta un campo por llenar.	
Inclusiones:	N/A	
Prioridad:	Importante	
Frecuencia de uso:	Cada vez que se requiera registrar un nuevo vehículo	
Requisitos especiales:	N/A	
Notas:	N/A	

Figura 8

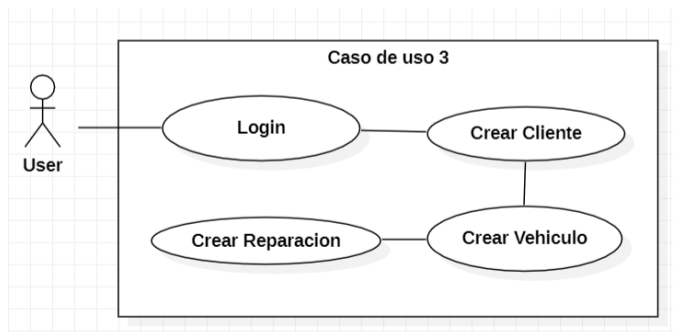


Tabla 6 - Caso de Uso CU04 (Agregar reparacion a un vehículo)

ID- UC	CU04		
Nombre de UC:	Agregar reparacion a un vehículo		
Creado por:	Desarrollador	Actualizado por:	
Fecha de la creación:		Fecha de la última Actualización:	

Actores:	Usuario
-----------------	---------

Descripción:	El usuario(empleador) podrá realizar el registro de una reparación de vehículo en el sistema asociado a un cliente.	
Disparador:	En la pantalla de registro de la reparación se solicitan algunos datos para así realizar el registro.	
Pre-Condiciones:	El usuario debe llenar los campos que se requieren para realizar el registro de la reparación.	
Pos-Condiciones	El servicio responderá con una respuesta.	
Flujo Normal:	1	Este caso de uso empieza en la pantalla del perfil del cliente donde se encuentran los vehículos, al darle click en el botón agregar reparación lo llevará a un formulario donde ingresará los datos solicitados en el formulario y le hace click en el botón de guardar.
	2	Ingresa la petición al servicio.
	3	El sistema valida la respuesta.
	4	El sistema da una alerta al usuario.
Flujo Alternativo:	5	Error al realizar la petición.
	6	Error al no poder guardar el registro de la reparación.
Excepciones:	El usuario no podrá registrar el vehículo si le hace falta un campo por llenar.	
Inclusiones:	N/A	
Prioridad:	Importante	
Frecuencia de uso:	Cada vez que se requiera registrar un nuevo vehículo	
Requisitos especiales:	N/A	
Notas:	N/A	

Figura 9

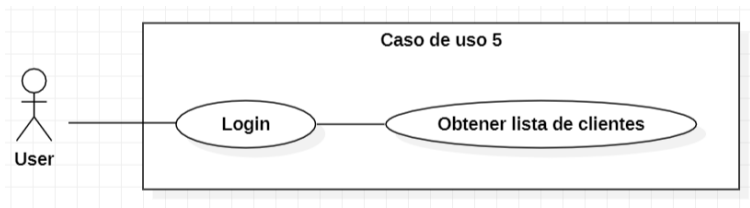


Tabla 7 - Caso de Uso CU05 (Obtener lista de clientes)

ID- UC	CU05		
Nombre de UC:	Obtener lista de clientes		
Creado por:	Desarrollador	Actualizado por:	
Fecha de la creación:		Fecha de la última Actualización:	

Actores:	Usuario	
Descripción:	El usuario(empleador) podrá ver el registro de todos los clientes	
Disparador:	En la pantalla de inicio (home) hay un botón que haciendo click en él nos llevará a la sección donde podremos ver el listado de clientes registrados .	
Pre-Condiciones:	El usuario debe iniciar sesión y seleccionar la opción de ver el listado.	
Pos-Condiciones	El servicio responderá con una respuesta.	
Flujo Normal:	1	Este caso de uso empieza en la pantalla de home seleccionando la opción ver listado de clientes
	2	Ingresa la petición al servicio
	3	El sistema valida la petición y devuelve respuesta.
	4	Se visualiza en una tabla la respuesta.

Flujo Alternativo:	5	Error al realizar la petición.
	6	Error al no mostrar clientes.
Excepciones:	El usuario no podrá ver los registros de los clientes si no ha iniciado la sesión.	
Inclusiones:	N/A	
Prioridad:	Importante	
Frecuencia de uso:	Cada vez que se requiera visualizar los clientes	
Requisitos especiales:	N/A	
Notas:	N/A	

Figura 10

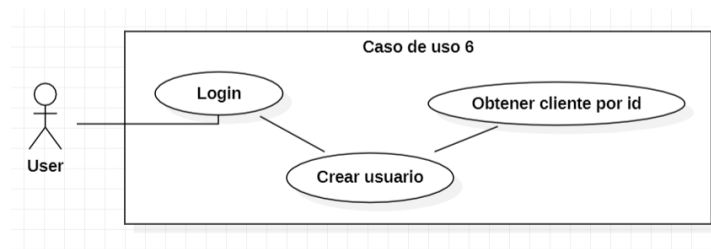


Tabla 8 - Caso de Uso CU05 (Obtener cliente por id)

ID- UC	CU05		
Nombre de UC:	Obtener cliente por id		
Creado por:	Desarrollador	Actualizado por:	
Fecha de la creación:		Fecha de la última Actualización:	

Actores:	Usuario
-----------------	---------

Descripción:	El usuario(empleador) podrá ver el registro de un cliente por el id dándole click en editar en la lista donde se pueden ver el listado de los clientes	
Disparador:	En la pantalla donde se muestra el listado de los clientes podremos ver que en la tabla hay un botón editar dándole click en este nos llevará al perfil de este y así podremos obtener la información completa de este cliente.	
Pre-Condiciones:	el cliente debe existir (haberlo registrado previamente) en la base de datos.	
Pos-Condiciones	El servicio responderá con una respuesta.	
Flujo Normal:	1	Este caso de uso empieza en la pantalla donde se puede visualizar el listado de clientes y haciendo click en el botón editar de algún cliente
	2	Ingresa la petición al servicio
	3	El sistema valida la petición y devuelve respuesta.
	4	Se visualiza la información del cliente.
Flujo Alternativo:	5	Error al realizar la petición.
	6	Error al no mostrar cliente.
Excepciones:	El usuario no podrá ver los registros del cliente si no ha iniciado la sesión y si no se encuentra en la base de datos.	
Inclusiones:	N/A	
Prioridad:	Importante	
Frecuencia de uso:	Cada vez que se requiera visualizar un cliente en específico	
Requisitos especiales:	N/A	
Notas:	N/A	

Figura 11

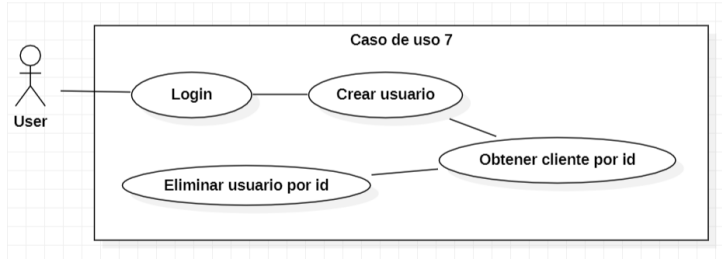


Tabla 9 - Caso de Uso CU06 (Eliminar un cliente por id)

ID- UC	CU06		
Nombre de UC:	Eliminar cliente por id		
Creado por:	Desarrollador	Actualizado por:	
Fecha de la creación:		Fecha de la última Actualización:	

Actores:	Usuario	
Descripción:	El usuario(empleador) podrá eliminar registro de un cliente por el id dándole click en eliminar en la lista donde se pueden ver el listado de los clientes	
Disparador:	En la pantalla donde se muestra el listado de los clientes podremos ver que en la tabla hay un botón eliminar dándole click en este se eliminará el registro del cliente.	
Pre-Condiciones:	el cliente debe existir (haberlo registrado previamente) en la base de datos.	
Pos-Condiciones	El servicio responderá con una respuesta.	
Flujo Normal:	1	Este caso de uso empieza en la pantalla donde se puede visualizar el listado de clientes y haciendo click en el botón eliminar de algún cliente
	2	Ingresa la petición al servicio

	3	El sistema valida la petición y devuelve respuesta.
	4	Aparece una alerta que se eliminó.
Flujo Alternativo:	5	Error al realizar la petición.
	6	Error al no eliminar el cliente.
Excepciones:	El usuario no podrá eliminar el registro si no ha iniciado sesión.	
Inclusiones:	N/A	
Prioridad:	Importante	
Frecuencia de uso:	Cada vez que se requiera eliminar un cliente en específico	
Requisitos especiales:	N/A	
Notas:	N/A	

Diagrama de clases:

En la figura 12 se observa el Diagrama de clases realizado para este proyecto en este se puede observar las entidades que interactúan en la aplicación y se presenta las relaciones con los demás objetivos dependiendo de su conectividad, cada entidad se representa por medio de una clase en la que se indica qué funciones va a realizar la entidad y los atributos que esta posee.

Figura 12

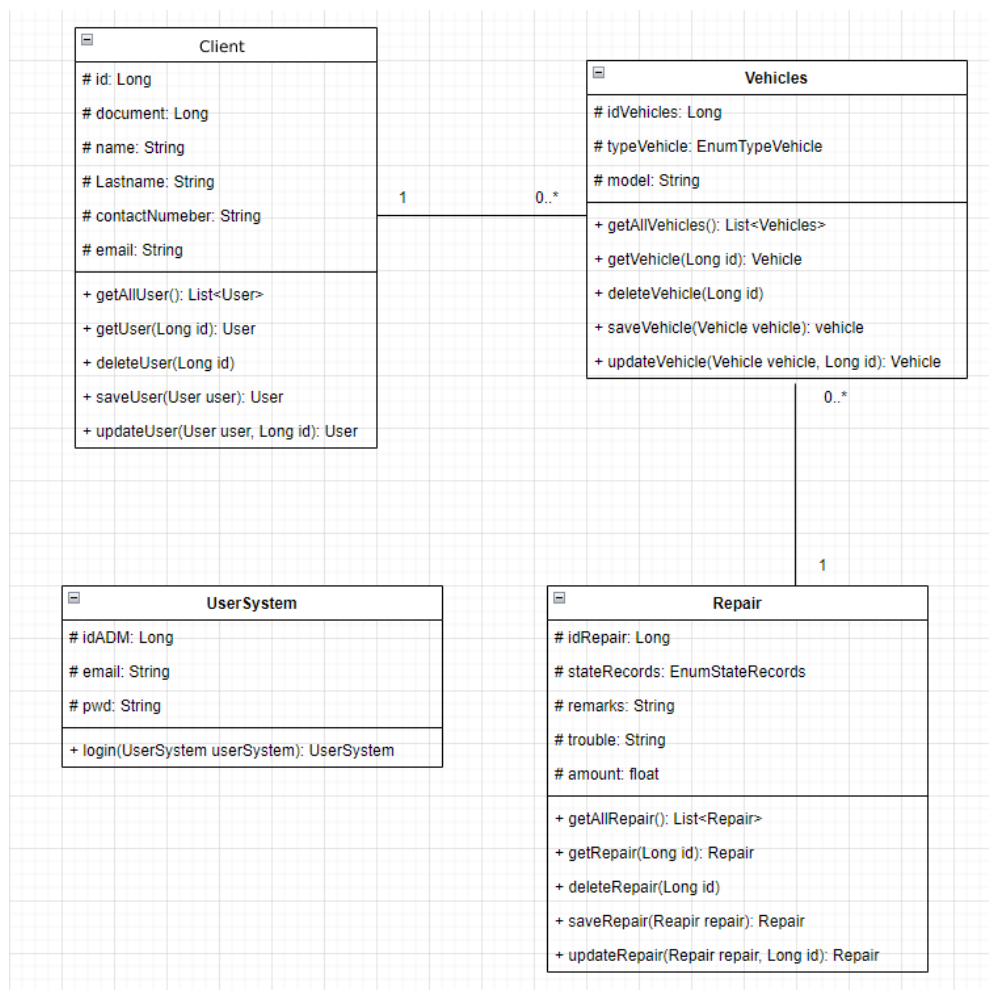


Diagrama de secuencia:

En la figura 13 encontramos el diagrama de secuencia del software donde podemos observar el comportamiento que va a tener el sistema de información, se realiza un diagrama de secuencia donde se puede encontrar el rol para los clientes y personas que prestan los servicios.

Figura 13

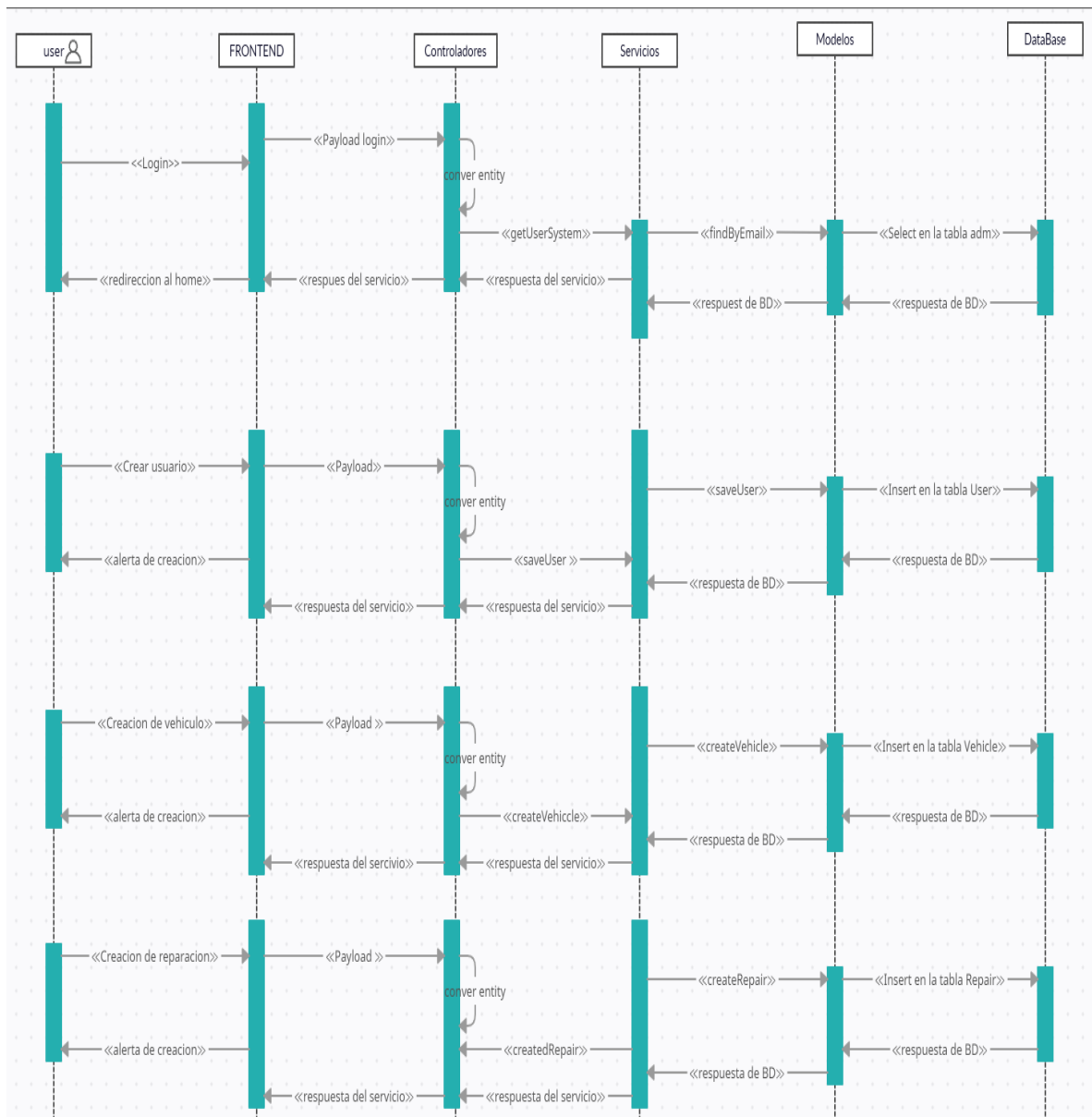
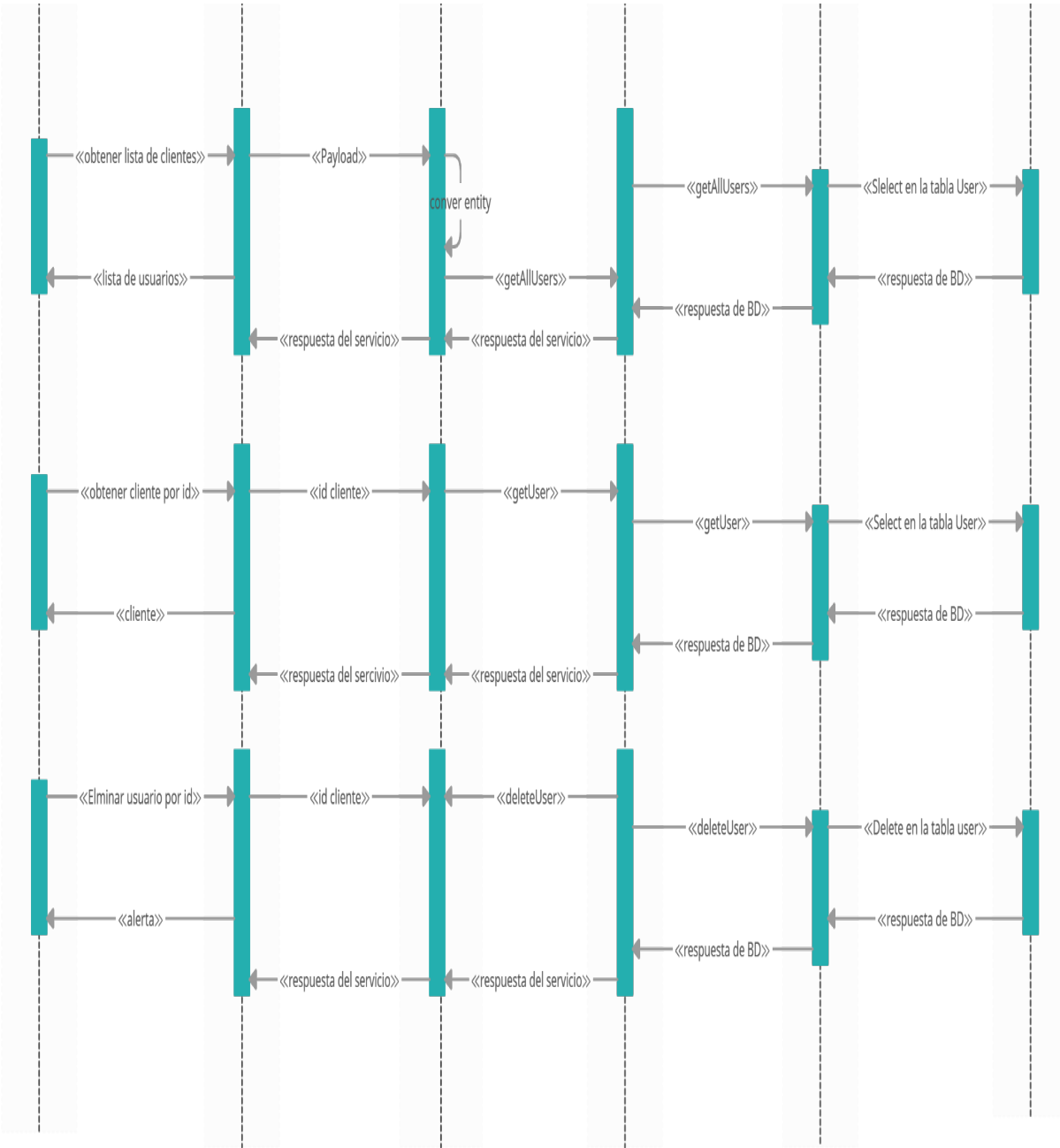


Figura 14



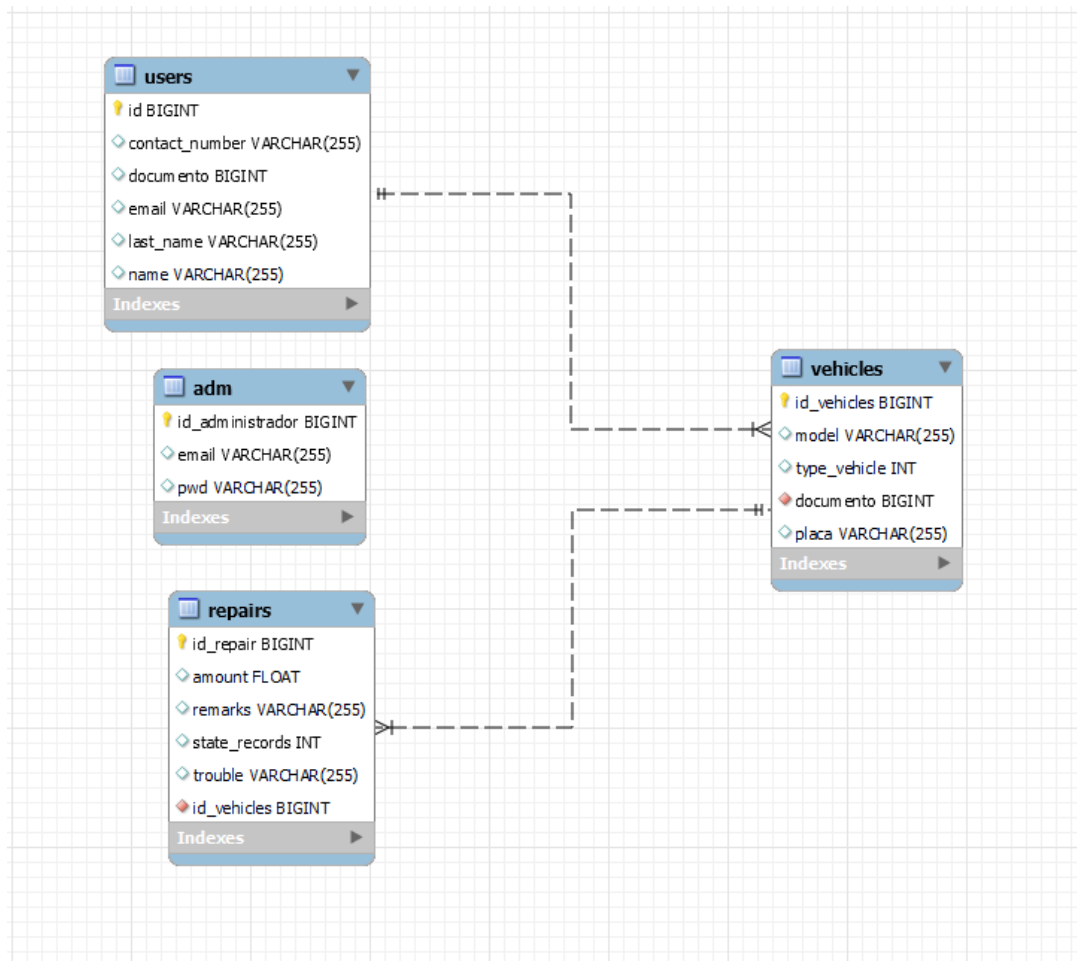
Modelos:

Se propuso un Modelo relacional y un modelo entidad - relación para la realización más sencilla y al saber cuál es el patrón del proyecto.

Modelo relacional:

En la figura 15 podemos observar el modelo racional del sistema, siendo una representación lógica del esquema de entidad- relación que aparece en la siguiente imagen. Este modelo contiene algunos registros para el uso del sistema

Figura 15



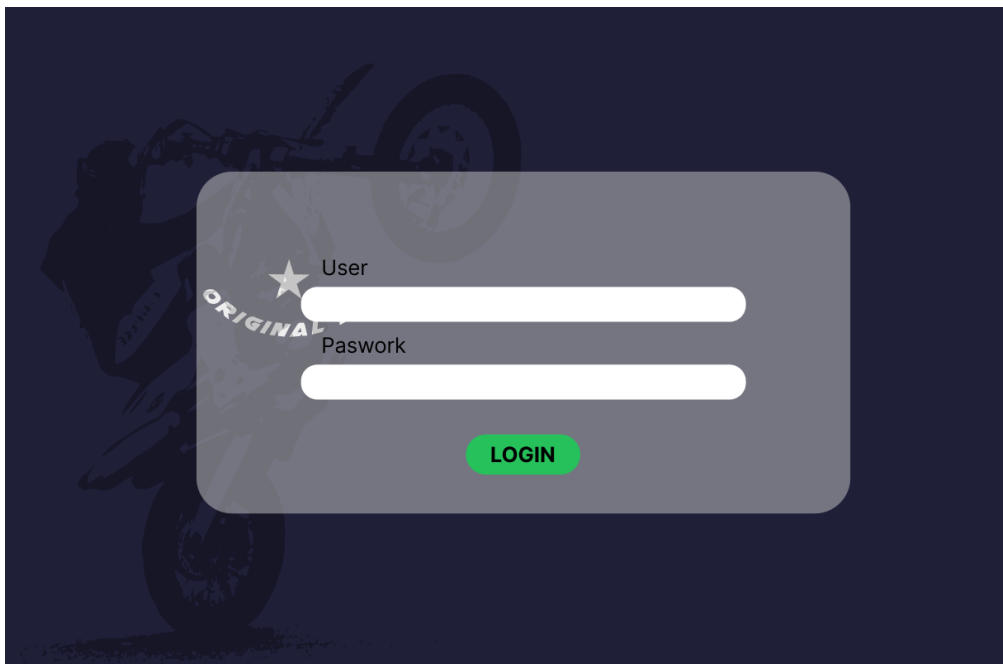
Mockups

Los diagramas propuestos anteriormente se convirtieron en punto de partida para identificar qué contenido va a tener Mockup y cómo se va a diseñar la Propuesta del trabajo de grado.

Aplicación de Móvil:

Login Usuario

Figura 16



En esta ventana el usuario puede ingresar al sistema con un usuario asignado

Home

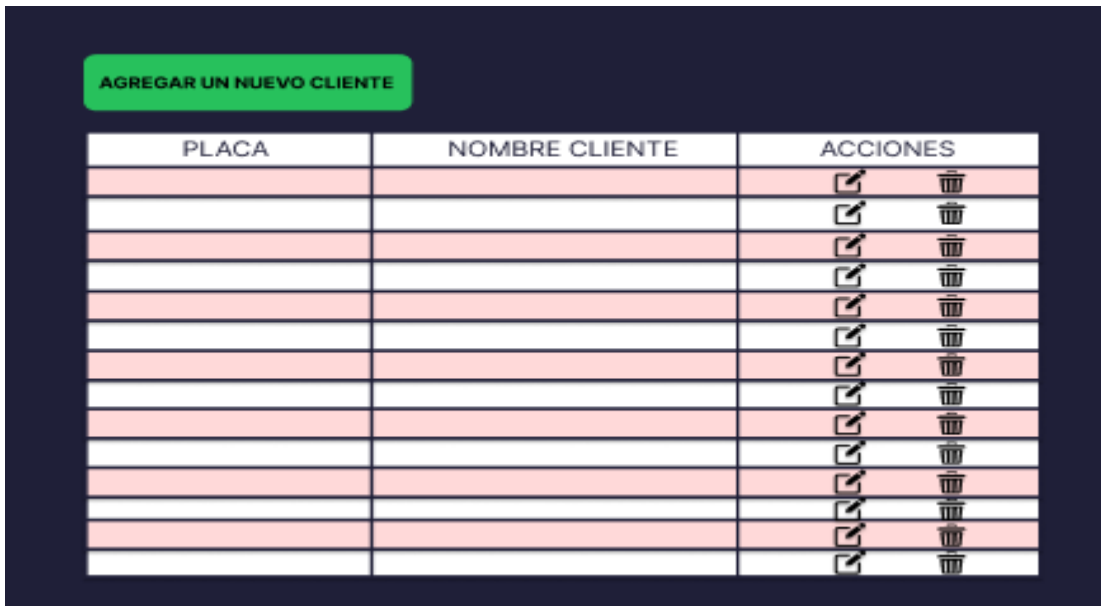
Figura 17



























Al ingresar al sistema nos mostrará un menú con múltiples opciones no disponibles aún, la única opción disponible es ver el listado de cliente

Lista de usuarios

Figura 18



The screenshot shows a dark blue interface. At the top left, there is a green button labeled "AGREGAR UN NUEVO CLIENTE". Below it is a table with three columns: "PLACA", "NOMBRE CLIENTE", and "ACCIONES". The table contains 12 rows, each with a pencil icon and a trash can icon in the "ACCIONES" column.

PLACA	NOMBRE CLIENTE	ACCIONES
		 
		 
		 
		 
		 
		 
		 
		 
		 
		 
		 
		 

En esta ventana nos encontraremos un listado de los vehículos registrados donde tendremos 2 opciones y es editar y borrar el vehículo, también podremos agregar un nuevo cliente desde esta ventana

Agregar un usuario

Figura 19

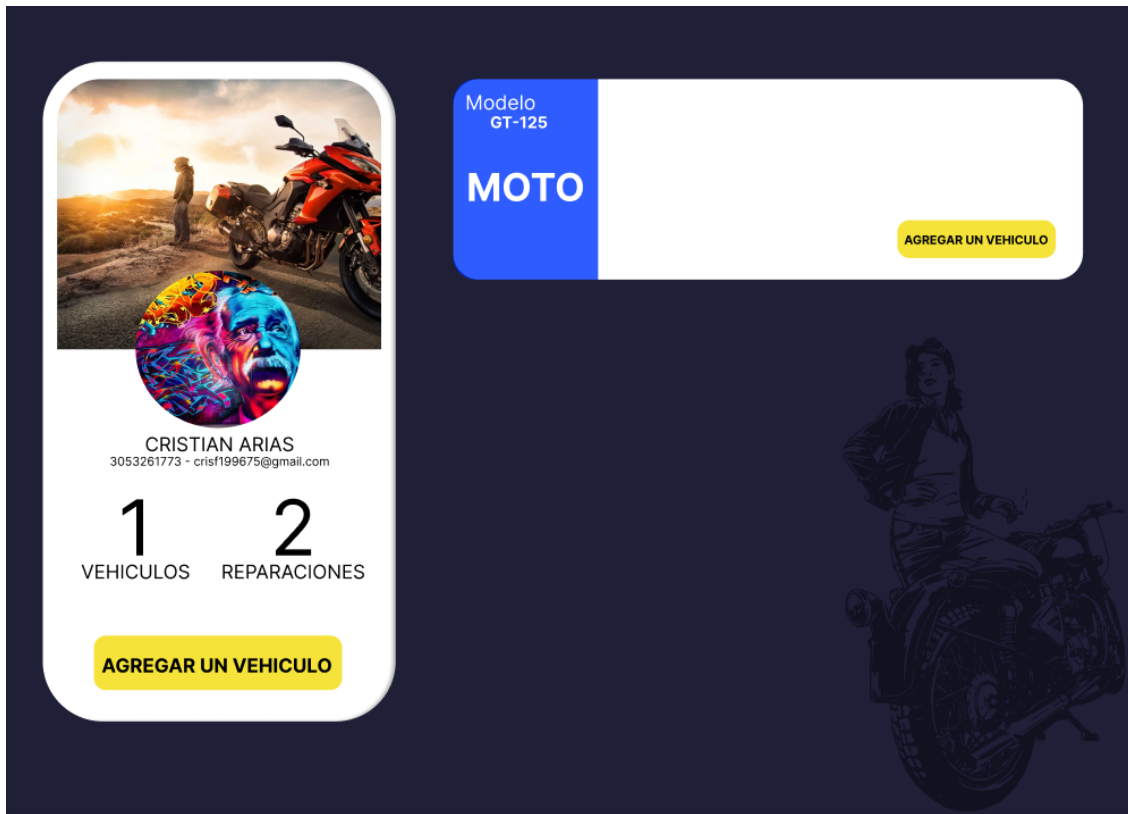


The screenshot shows a dark blue interface with a light gray rounded rectangle containing a form titled "AGREGAR USUARIO NUEVO". The form has four input fields: "Nombres", "Apellidos", "Email", "Celular", and "Documento". At the bottom left, there are two buttons: "CANCELAR" (yellow) and "GUARDAR" (green). On the right side, there is a faint illustration of a person riding a motorcycle.

Podremos registrar un usuario a través de esta sección y que aparezca en el listado de usuarios

Perfil de usuario registrado

Figura 20

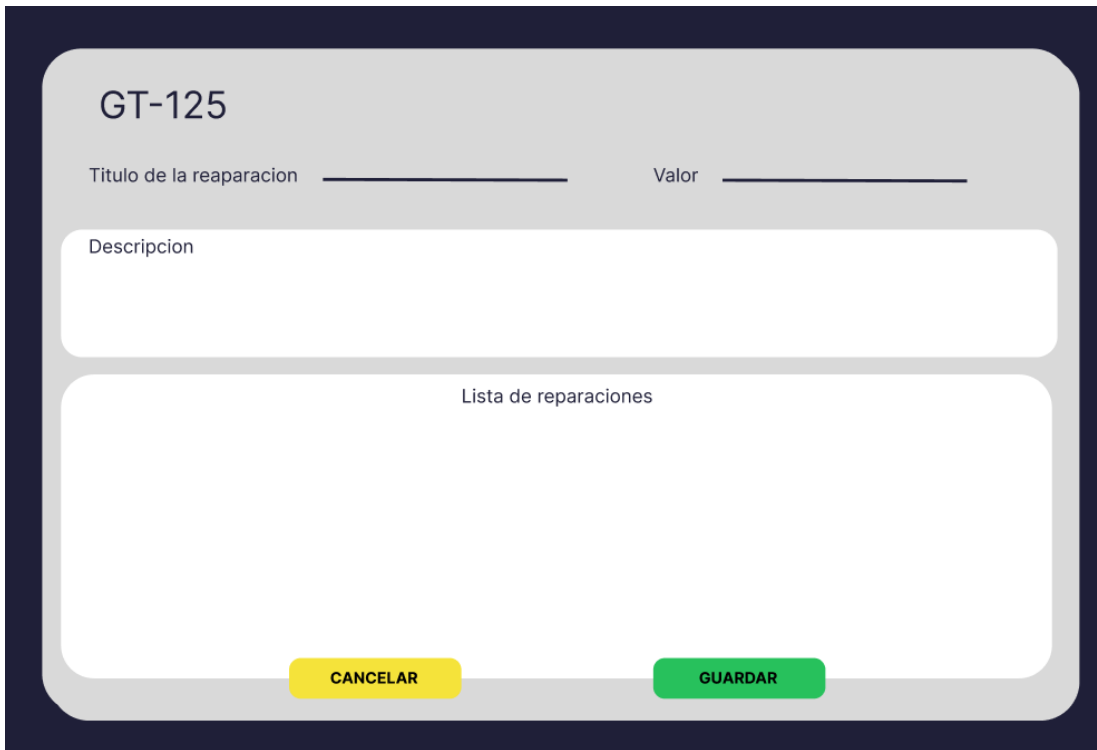


En esta pantalla podremos ver algunos datos de contacto del cliente registrado, cuántos vehículos posee registrados a su nombre y un total de reparaciones.

También veremos una card con una pequeña información sobre el vehículo y esta posee un botón donde nos llevara a ver más sobre las reparaciones del vehículo

Ver más detalles sobre Reparaciones del Vehículo

Figura 21



The screenshot shows a form titled "GT-125" for recording a vehicle repair. At the top, the text "GT-125" is displayed. Below it, there are two input fields: "Titulo de la reparacion" and "Valor". A large text area labeled "Descripcion" is positioned below these fields. At the bottom of the form, there is a section labeled "Lista de reparaciones" which is currently empty. Two buttons are located at the bottom: a yellow "CANCELAR" button and a green "GUARDAR" button.

En esta sección se muestra un listado de las reparaciones hechas con un estado y se podrá registrar una nueva reparación

Agregar Vehículo

Figura 23



The screenshot shows a form titled "AGREGAR VEHICULO NUEVO". It contains three input fields: "Placa", "Model", and "Tipo de vehiculo". The "Tipo de vehiculo" field is currently empty. At the bottom, there are two buttons: a yellow "CANCELAR" button and a green "GUARDAR" button. On the right side of the form, there is a grayscale illustration of a person riding a motorcycle.

Capítulo 7: Recomendaciones

Se recomienda a la hora de desarrollar el proyecto, que lo haga una persona que tenga conocimientos en programación Web, además conocimientos en React.js, MySQL, Spring Boot.

Se recomienda que desarrollen el proyecto en una computadora que contenga Visual Studio Code, MySQL e IntelliJ. Esta computadora debe tener como mínimo una memoria de 35 GB de RAM para tener un rendimiento estable.

Este Proyecto se recomienda hacerlo para mejorar el control de los registros que se realizan a diario dentro del taller.

Al realizar un proyecto de software es importante que las personas que participen en su implementación tengan los conocimientos adecuados para que el software cumpla con todas las necesidades que se identificaron, además se base en una metodología y arquitectura que brinde funcionalidad y calidad.

Capítulo 8: Conclusiones

Se construyó el diseño del proyecto con la realización de un Mockup que muestra cada ventana de la aplicación Web, propuesta para el sistema “SDMOTO & REPARAAUTOS”. Este Mockup fue realizado con la herramienta de diseño Figma.

Se construyeron los requerimientos funcionales y no funcionales que debía tener el Mockup ayudando para detectar qué contenido debería tener, qué actividades deberían hacer y que se necesita para realizar esas actividades.

Se realizó cada Diagrama para representar gráficamente los procedimientos, procesos, ideas, soluciones, y mecanismos para que se pueda comprender el proyecto en un futuro para su desarrollo.

Se construyó el diseño de la Estructura de la Base de datos con los modelos Relacional e Identidad Relación, para hacer la propuesta de esta estructura, para que en un futuro cuando se vaya a desarrollar el proyecto, se tenga idea de cómo va construido y conexión entre tablas.

Se implementa la base de datos utilizando MYSQL, como motor de base de datos, para el almacenamiento de los datos.

Se implementa tecnologías como React.js para desarrollar el frontend y para el backend Spring Boot, siendo las más adecuadas por estabilidad, escalabilidad y rendimiento.

Bibliografía

Work Orders. (9 de abril de 2022) Obtenido de <https://www.workordersoft.com/repairshopscreentour.htm>

Mitchell 1. (9 de abril de 2022) Obtenido de <https://mitchell1.com/alt-home/index.html>

Kordata. (9 de abril de 2022) Obtenido de <https://www.kordata.mx/>

Colaboradores de Wikipedia. (2022, 24 octubre). *Framework*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Framework>

Spring Boot. (s. f.). spring. <https://spring.io/projects/spring-boot>

El diagrama de casos de uso en UML. (2020, 24 julio). IONOS Digital Guide. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>

Rvillarroel16, V. T. L. E. de. (2017, 20 enero). Requerimientos Funcionales y No Funcionales. <https://ingenieriadesoftwareutmachala.wordpress.com/2017/01/20/requerimientos-funcionales-y-no-funcionales/>

Zamenfeld, S. (2011, 1 julio). *StarUML una herramienta para modelado*. <https://www.brainlabs.com.ar/novedad/staruml-una-herramienta-para-modelado/>

React | Qué es, para qué sirve y cómo funciona | Descúbrelo todo. (2021, 8 noviembre).

Tribalyte Technologies. <https://tech.tribalyte.eu/blog-que-es-react>

Jetbrains (s. f.). IntelliJ <https://www.jetbrains.com/es-es/idea/features/>

Bracey, K. (2022, 13 mayo). ¿Qué es Figma? Web Design Envato Tuts+. <https://webdesign.tutsplus.com/es/articles/what-is-figma--cms-32272>

Java. (s. f.). https://www.java.com/es/download/help/whatis_java.html

Qué es Visual Studio Code y qué ventajas ofrece. (2022, 3 noviembre). OpenWebinars.net. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>