



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



“TallerCom”

Profesor

Dorantes Gonzales Marco Antonio

INTEGRANTES:

Basilio Orihuela Levi Abdul

Figueroa de la Torre Heidi Fernanda

Pimentel Paulin Daniel Jezrael

Tello Gómez Lilia Anayeli

ÍNDICE

CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS	3
Introducción.....	3
Estado del arte.....	3
Justificación.....	4
Objetivos.....	4
CAPÍTULO 2. MARCO TEÓRICO.....	5
CAPÍTULO 3. ANÁLISIS.....	6
Descripción.....	6
Requerimiento técnicos.....	6
Arquitectura propuesta.....	6
Metodología de desarrollo.....	7
Entregables.....	9
Recolección de requisitos.....	9
Requerimientos funcionales.....	10
Requerimientos no funcionales.....	11
Actores.....	11
Diseño del sistema.....	12
Diagrama de casos de uso.....	12
Diagrama de clases	18
Diagrama de estados.....	21
Diagrama de secuencias.....	21
Diagrama de actividades.....	26
CAPÍTULO 4. DISEÑO.....	30
Pantalla de inicio de sesion.....	30
Sesión de Administrador	31
Sesión de Trabajador.....	36
Pantalla de ajuste.....	42
CAPÍTULO 5. CODIFICACIÓN Y PRUEBAS.....	46
CAPÍTULO 6. PRUEBAS.....	49
Plan de pruebas Plan001_TallerCom.....	49
Objetivo.....	49

Estrategia.....	49
Alcance.....	49
Propósito.....	49
Entorno de configuración de pruebas.....	49
Características a ser probadas.....	51
Criterios de aprobación y fallo.....	51
Criterios de suspensión y reanudación.....	51
CAPÍTULO 7. CONCLUSIONES.....	53
REFERENCIAS.....	53
ANEXOS.....	53

CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS

Nombre del proyecto:

TallerCom

Contactos:

Levi Basilio: levi13basilio@gmail.com

Heidi Figueroa: heidfer056@gmail.com

Daniel Pimentel: daniel.pim3ntel@gmail.com

Lilia Tello: bayte25@gmail.com

Introducción

La mayoría de los talleres mecánicos locales de nuestra región no cuentan con sistemas tecnológicos para llevar a cabo la información de su taller, lo que es una carga de trabajo extra ya que resguardar información importante como, los datos de los clientes, de manera convencional es arriesgado ya que estos datos pueden ser extraviados fácilmente o no llevar el control adecuado de estos, perdiendo información importante, lo cual puede traer problemas con el cliente, ya que si los datos de ajustes que se tienen que realizar a un automóvil son confundidos con los de otro cliente genera graves problemas

Los Talleres mecánicos buscan brindar un servicio automotriz integral de calidad enfocado a las necesidades de cada uno de sus clientes a un precio justo y asequible. Asegurando una excelencia de servicio con una atención especial y personalizada. Lo que puede llegar a ser abrumador para los trabajadores ya que hoy en día vivimos en un mundo rodeado de tecnología y los clientes tienen la necesidad de mantenerse notificados por medios tecnológicos acerca del avance o situación de su automóvil, para tener una seguridad de que su automóvil está siendo revisado de manera correcta.

Los trabajadores necesitan tener una buena organización de la información que reciben de sus clientes para poder elaborar los ajustes de manera correcta y no confundir los ajustes a realizar o piezas que se tienen que cambiar del automóvil de un cliente con los ajustes del automóvil de otro cliente, estas situaciones disminuyen la eficiencia de los talleres mecánicos, lo cual implicaría pérdidas de tiempo, las cuales son de vital importancia tanto para el cliente como para los trabajadores dentro del taller

Estado del arte

TallerCom	Taller Alpha	RMS Gestion	Taller GP
Permite llevar un registro de clientes, sus vehículos, reparaciones hechas y por hacer.	Permite operar desde la aplicación donde se puede gestionar la recepción y salida de	Diseñado para la gestión de cualquier servicio técnico, solo permite trabajar con impuestos de	Este programa permite crear un usuario para cada miembro del taller, de forma que cada uno

Notifica al cliente en tiempo real el avance de las reparaciones a su vehículo, además de notificar sobre posibles reparaciones a futuro. Dará al cliente una factura que desglosa los gastos de reparaciones y por refacciones.	vehículos. Tiene un panel administrativo desde el cual se puede controlar todo lo relacionado a los clientes. Cuenta con una sección de tienda donde puede ofrecer sus productos.	IVA.	lo pueda utilizar desde su propio dispositivo.
--	---	------	--

Tabla 1. Aplicaciones similares

Justificación

En el mercado se encuentran diversas aplicaciones enfocadas a la gestión de un taller mecánico o de cualquier servicio técnico, estas sirven para facilitar el manejo del mismo, pero ninguna de estas te permiten una interacción con el cliente o notificación que permita que el cliente sepa como va el mantenimiento de su vehículo, es por esto que esta aplicación aparte de ayudar y tener un historial concreto del cliente, permitirá notificar al cliente acerca de las reparaciones en tiempo real y futuras de su vehículo.

Objetivos.

General:

Llevar el control de clientes y sus vehículos, además de justificar los costos de las reparaciones ante el cliente de un taller mecánico para facilitar la gestión del taller.

Particulares:

- Llevar el registro de las de las reparaciones de cada auto de un cliente, para así tener un historial del mismo para justificar el trabajo hecho en los vehículos.
- Notificar al cliente mediante notificaciones a su correo electrónico donde se le dé a conocer el avance de las reparaciones en tiempo real de su vehículo.
- Informar y registrar las reparaciones pendientes que el automóvil del cliente podría tener posibles fallas, con el propósito de agendar una cita a futuro.
- Entregar de forma digital o impresa una nota que justifique la cantidad cobrada al cliente, con los precios de cada reparación y/o refacciones compradas.

CAPÍTULO 2. MARCO TEÓRICO

Un taller mecánico- automotriz es un establecimiento industrial donde se realizan reparaciones para establecer las condiciones normales del estado y funcionamiento de un vehículo automóvil o de sus equipos y componentes.

Dichas reparaciones son realizadas por uno o más técnicos especializados llamados mecánicos. Los talleres mecánico- automotriz pueden estar especializados en automóviles, motocicletas u otros vehículos.

La función de un taller de servicio automotriz es la de ejecutar trabajos de reparación, diagnóstico o sustitución en el sistema mecánico de un vehículo, incluyendo su equipo eléctrico, sus estructuras y sus equipos auxiliares de acondicionamiento, alumbrado, señalización e instrumentación de indicación y control

Para que un taller sea productivo, es necesario tener un flujo de trabajo óptimo . Éste se da cuando todas las operaciones de servicio del taller fluye uniformemente sin ninguna obstrucción, creando la más alta productividad posible.

Tipos de talleres mecánicos.

-Talleres de mecánica general:

Son los establecimientos encargados de realizar un servicio de mantenimiento o reparación de averías en la mayoría de los componentes de un vehículo como por ejemplo: cambio de amortiguadores, baterías, filtros, neumáticos, frenos, etc.

-Talleres de mecánica especializada.

Son los establecimientos que se dedican a ofrecer servicios más especializados ya sea mantenimiento, reparación o instalación de nuevos accesorios en el vehículo. Algunos ejemplos de talleres especializados pueden ser:

-Taller de pintura y chapistería:

Especializados en pintura y carrocería del automóvil ya sea reparación, sustitución o acoplamiento de nuevos componentes.

-Taller mecánico- eléctrico por autómatas:

Especializados en sistemas automotrices motivados por autómatas con personal cualificado y herramientas especiales para determinados vehículos y sistemas.

-Taller de accesorios, sonido y alarma:

Especializados en accesorios especiales y/o exclusivos.

-Talleres de Concesionario:

La mayoría de concesionarios de una o varias marcas de automóviles tienen un taller especializado para los automóviles de esta determinada marca o marcas. Confrontan un taller con técnicos especialistas en todas o casi todas las áreas y equipados con maquinaria específica.

CAPÍTULO 3. ANÁLISIS

Descripción

La aplicación consta de un sistema que lleva el control de talleres mecánicos, la cual tiene una interfaz que permite revisar los clientes registrados; los datos que se guardan del cliente son nombre, alias, número de carros, registro de autos, número de teléfono y correo electrónico. De cada auto se almacena su modelo, color, año, placas, mes de verificación y una lista de reparaciones. El registro se encarga de llenar una especie de formulario donde se colocan los tipos de reparaciones que se le hicieron al vehículo, una breve descripción, la fecha del ajuste y una lista de registros con reparaciones que quedaron pendientes (por razones ajenas al taller). La interfaz de notas; es posible crear una nota que permita al cliente identificar porque se le está cobrando la cantidad de dinero que solicita el taller, incluye los gastos de cada reparación y si es necesario el costo de las piezas que se compraron para reparar el automóvil. Finalmente, las interfaces anteriormente serán controladas por los trabajadores del taller, así se deberá tener un "login" para que el trabajador pueda acceder a la aplicación, del trabajador se guarda su nombre, contraseña, teléfono y correo electrónico

Requerimientos técnicos

Hardware:

- Computadora
- Acceso a internet

Software:

- MySQL: es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,¹² y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.
- Java: Java es un lenguaje de programación orientado a objetos, juega un papel importante en el desarrollo de páginas Web interactivas y con multimedia. Tiene prominencia al desarrollar aplicaciones para Internet e intranets.

Arquitectura propuesta

Descomposición modular:

Descomposición Modular es el proceso de descomposición de un sistema en un conjunto de elementos con un índice bajo acoplamiento (independientes) y alto índice de cohesión.

Consiste en descomponer el problema a resolver en módulos o tareas más simples. Cada tarea representa una actividad completa y se codifica de manera independiente. Facilita el diseño descendente del problema, centrándose cada vez más en la resolución de subproblemas de magnitud inferior. A la resolución de cada uno de estos subproblemas de complejidad inferior se denomina refinamiento por pasos. Los módulos pueden ser planificados, codificados, comprobados y depurados independientemente.

El diseño modular propone dividir el sistema en partes diferenciadas y definir sus interfaces.

Una descomposición modular debe poseer ciertas cualidades mínimas para que se pueda considerar suficiente valida, las cuales son las siguientes:

- Independencia funcional: Cada módulo debe realizar una función concreta o un conjunto de funciones afines. Es recomendable reducir las relaciones entre módulos al mínimo.
- Acoplamiento: El grado de acoplamiento mide la interrelación entre dos módulos, según el tipo de conexión y la complejidad de la interfaz.
- Cohesión: Es necesario lograr que el contenido de cada módulo tenga la máxima coherencia. Para que el n° de módulos no sea demasiado elevado y complique el diseño se tratan de agrupar elementos afines y relacionados en un mismo módulo.
- Comprensibilidad: Para facilitar los cambios, el mantenimiento y la reutilización de módulos es necesario que cada uno sea comprensible de forma aislada.
- Adaptabilidad: La adaptación de un sistema resulta más difícil cuando no hay independencia funcional, es decir, con alto acoplamiento y baja cohesión, y cuando el diseño es poco comprensible.

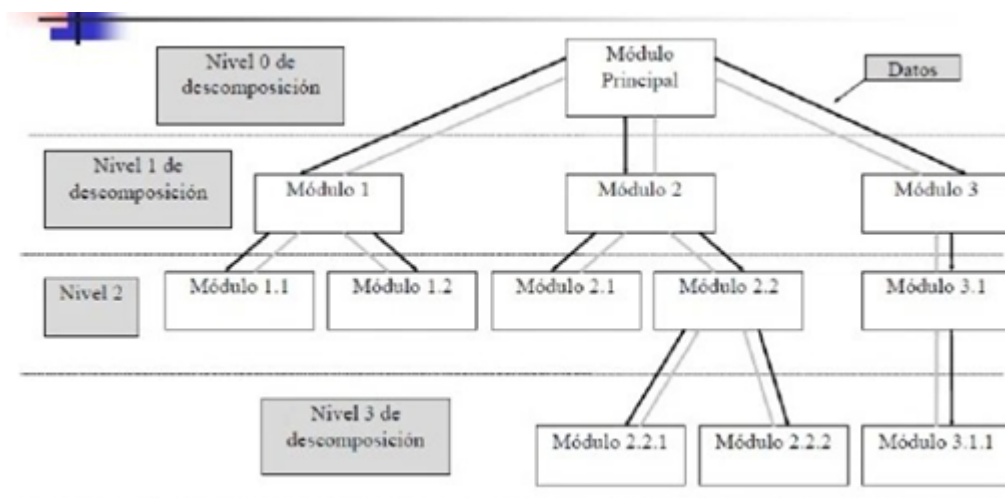


Figura 1. Arquitectura de descomposición modular

Metodología de desarrollo:

Metodología OMT(Object Modeling Technique):

La metodología OMT (Object Modeling Technique) fue creada por James Rumbaugh y Michael Blaha en 1991, mientras James dirigía un equipo de investigación de los laboratorios General Electric.

El significado de las siglas de esta metodología es Técnica de Modelado en Objetos (Object Modeling Technique), la definen como una de las metodologías de la Ingeniería de Software aplicable al desarrollo orientado a objetos en las fases de análisis y diseño.

Las fases que conforman a la metodología OMT son:

- **Análisis.** El analista construye un modelo del dominio del problema, mostrando sus propiedades más importantes. Se expande en 3 modelos:
 - Modelo de objetos. Describe la estructura estática de los objetos del sistema (identidad, relaciones con otros objetos, atributos y operaciones)
 - Modelo dinámico. Representa la interacción entre esos objetos representados como eventos, estados y transiciones. Se representa gráficamente mediante diagramas de estado.
 - Modelo funcional. Describe las transformaciones de valores de datos (funciones, correspondencias, restricciones y dependencias funcionales) que ocurren dentro del sistema. Se representa mediante diagramas de flujo de datos
- **Diseño del sistema.** Aquí se define la arquitectura completa del sistema. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta.
- **Diseño de objetos.** Aquí se define la arquitectura completa del sistema. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta.
- **Implementación.** Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta.

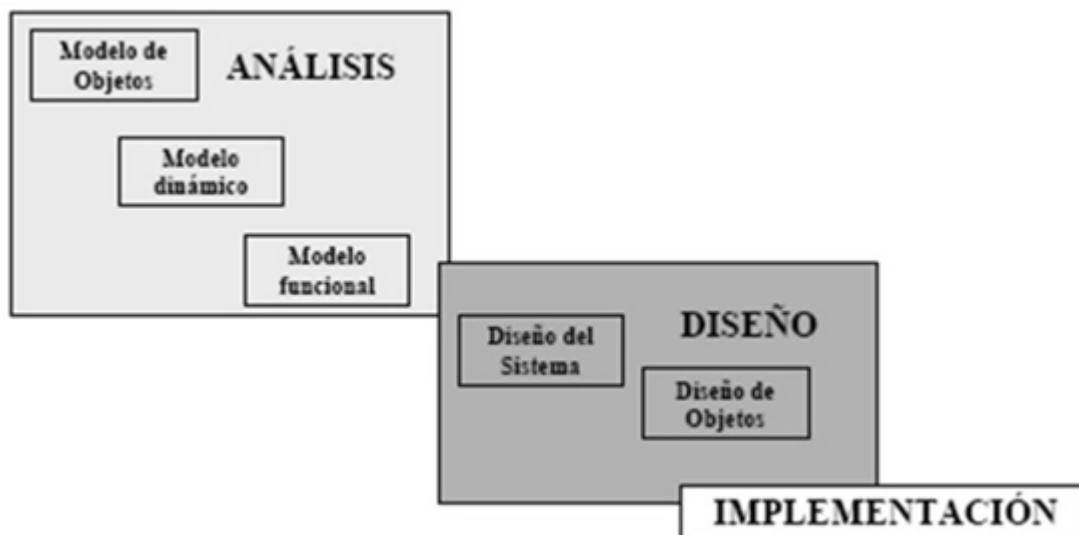


Figura 2. Fase de la Metodología OMT

Entregables:

- Manual técnico: Se entregará un manual técnico dirigido a la dirección de IT, al administrador del sistema y a otros desarrolladores que puedan darle mantenimiento en caso que se requiera.
- Manual de usuario: Se entregará un manual de usuario el cual estará destinado a dar asistencia a las personas que estén haciendo uso de este sistema.
- Tríptico: Se entregará un tríptico informativo para dar a conocer las principales características de la aplicación para dar promoción o incentivar el uso de este sistema.
- Poster: Se entregará un póster informativo y gráfico que promociona la aplicación dando a conocer los puntos importantes de esta y para quién está dirigida.

Recolección de requisitos

Para la recolección de requisitos el taller mecánico “Servicio integral de Mantenimiento Automotriz PONCE” nos solicitó las características que el sistema debía de cumplir de acuerdo a las necesidades de su taller mecánico, las cuales fueron las siguientes:



Servicio integral de Mantenimiento Automotriz PONCE

Se solicita una aplicación para una computadora de escritorio que permita a los trabajadores añadir clientes y sus vehículos, de los vehículos se requiere guardar las placas, el color, mes de verificación y los ajustes realizados a cada uno de los autos, cabe resaltar que de los ajustes es necesario guardar el día que se realizó, una descripción del ajuste y ajustes que se quedaron pendientes de hacer. Es importante que exista una cuenta que puede agregar e incluso eliminar trabajadores en caso de ser necesario.

También es necesario un archivo de Excel que el taller pueda modificar para generar notas e imprimirlas.

Sin más por el momento, un cordial saludo de parte del Servicio integral de Mantenimiento Automotriz PONCE.

Requerimientos funcionales:

La información utilizada para el análisis de los requerimientos fue recopilada gracias a la técnicas de recolección de requisitos mencionada anteriormente.

ID	Nombre	Descripción
RF1	Registrar trabajador	El administrador deberá introducir su contraseña para que se pueda registrar un nuevo trabajador en el sistema.
RF2	Actualizar los datos del trabajador	El administrador tendrá acceso para actualizar la contraseña o el correo del trabajador.
RF3	Autenticación de trabajador	El trabajador deberá registrar su contraseña y usuario correctos para poder acceder al sistema.
RF4	Crear cuenta cliente	Se crea una cuenta cliente, ingresando los datos necesarios para su registro
RF5	Actualizar datos del cliente	El trabajador podrá actualizar si es necesario los datos del cliente con su debida autorización
RF6	Registrar automóvil nuevo	El trabajador registrará el automóvil nuevo y a que cliente pertenece
RF7	Consultar automóvil	El trabajador podrá hacer consultas para observar el historial de cierto automóvil
RF8	Agendar cita de mantenimiento	El trabajador podrá agendar una cita de mantenimiento con la debida solicitud del cliente
RF9	Notificar al cliente de su cita	El sistema enviará un correo electrónico al cliente el día de la cita
RF10	Registro del ajuste que se realizará	El trabajador realizará los registros necesarios notificando que ajuste se realizó, en qué automóvil y quien es el cliente que lo solicitó
RF11	Realizar nota del servicio	Una vez terminado el servicio se realizará una nota la cual se le enviará al cliente
RF12	Realizar nota	Una vez ingresados las refracciones y ajustes realizados al auto el cliente recibirá su nota del servicio

Tabla 2. Requerimientos funcionales

Requerimientos no funcionales

ID	Nombre	Descripción
RNF 1	Interfaz	Se tendrá una interfaz fácil de utilizar e interactiva para el trabajador.
RNF 2	Restricción	Únicamente el administrador podrá registrar a nuevos trabajadores.
RNF 3	Confidencialidad	Los datos ingresados tanto de los trabajadores como los de los clientes serán resguardados.
RNF 4	Usabilidad	El sistema facilitará la administración de los talleres, además de que llevará un control de los vehículos de cada cliente, sus reparaciones, citas y pagos.
RNF 5	Disponibilidad	El sistema deberá funcionar mínimo 12 horas por día.

Tabla 3. Requerimientos no funcionales

Actores:

- Administrador: Persona que tiene acceso a todas las vistas del sistema, permitiendo agregar y eliminar trabajadores
- Trabajador: Persona que no puede agregar o eliminar a otros trabajadores y comparte las demás funciones con el administrador
- Cliente: Persona que proporciona información
- Automóvil: Recibirá el servicio del taller mecánico, el cual es solicitado por el cliente

Diseño del sistema

Diagrama de casos de uso

Utilizamos un diagrama de casos de uso para especificar el comportamiento de nuestro sistema mediante la interacción con nuestros actores, los cuales son el Cliente, Administrador, Trabajador y el Automóvil, y los casos de uso, los cuales representan las acciones que podrán llevar a cabo cada uno de los actores dentro de nuestro sistema

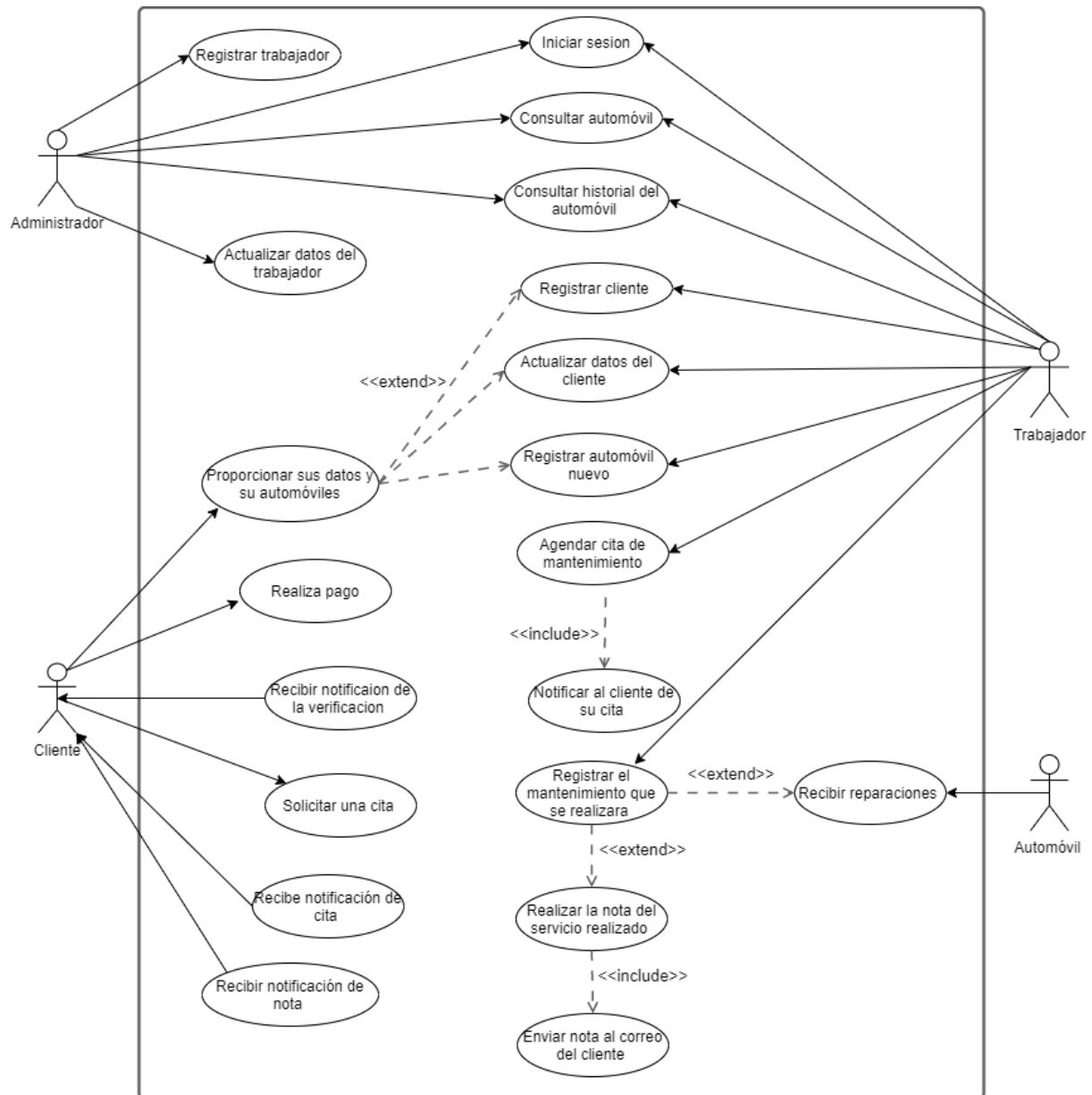


Figura 3. Diagrama de casos de uso general

Agendar cita de mantenimiento

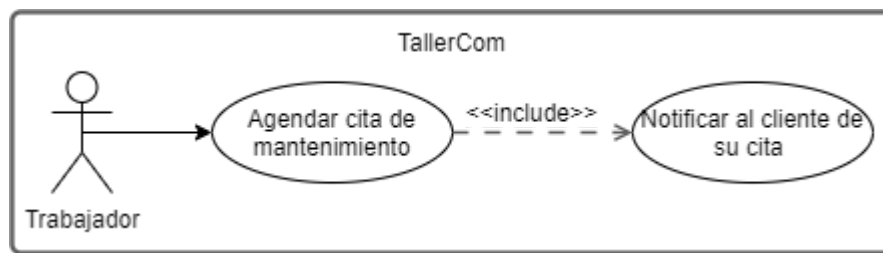


Figura 4. Requerimientos no funcionales

Caso de uso:	Agendar cita de mantenimiento
Versión:	1.0
Actor(es):	Trabajador
Propósito:	Registrar una cita para el mantenimiento del auto en cuestión
Resumen:	Proceso para agendar cita
Entradas:	-Cuadro de texto Cliente - Calendario de selección de la Fecha -Opciones de la hora -Botón de Agendar cita
Salidas:	Se agenda la cita
Precondiciones:	Tener un cliente registrado y disponibilidad del día y hora requerida
Postcondiciones:	Se agenda la cita y se envía un correo con esta

Tabla 4. Requerimientos no funcionales

Flujo de eventos:

Trayectoria principal:

1. El trabajador elige la opción “cita”.
2. El sistema muestra la pantalla “Agendar cita”.
3. El trabajador ingresa el cliente al que se le dará la cita.
4. El trabajador selecciona una fecha (Las fechas en color rojo no están disponibles).
5. El trabajador selecciona una hora.(Solo aparecen los horarios disponibles en la fecha seleccionada).
6. El trabajador presiona el botón “Agendar cita”.

--Fin de trayectoria

Notificar al cliente de su cita

Caso de uso:	Notificar al cliente de su cita
Versión:	1.0
Actor(es):	Trabajador
Propósito:	Enviar una notificación al correo del cliente indicando la fecha de la cita para el mantenimiento de su auto
Resumen:	Entregar notificación de la cita al cliente
Entradas:	Botón Enviar cita
Salidas:	Se envía la cita al correo del cliente
Precondiciones:	Tener una cita registrada
Postcondiciones:	El cliente recibe su cita

Tabla 5. Requerimientos no funcionales

Flujo de eventos:

Trayectoria principal:

1. El trabajador oprime el botón “Enviar cita”.
2. El correo con la fecha de la cita es enviado vía correo electrónico al cliente correspondiente

--Fin de trayectoria

Registrar el mantenimiento que se realizará

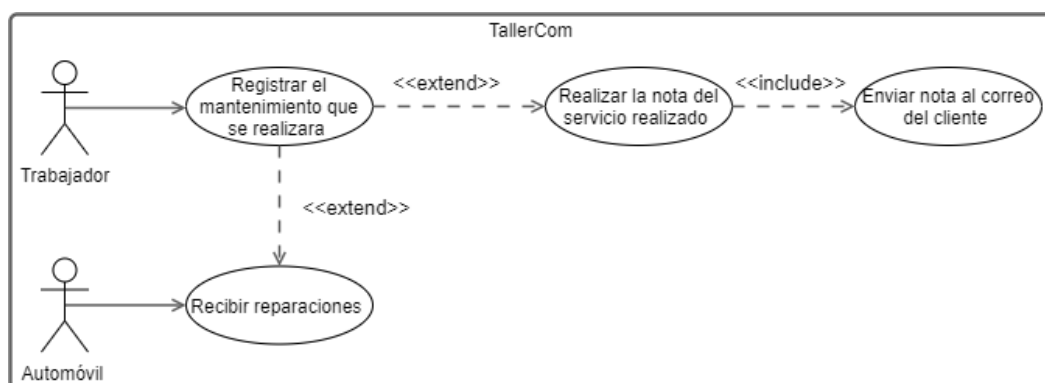


Figura 5. Requerimientos no funcionales

Caso de uso:	Registrar el mantenimiento que se realizará
Versión:	1.0

Actor(es):	Trabajador
Propósito:	Tener el registro del mantenimiento que se tendrá que realizar al automóvil del cliente
Resumen:	Proceso del registro del mantenimiento del auto
Entradas:	-Cuadro de texto Placas -Barra de selección Tipo de ajuste -Cuadro de texto Descripción -Cuadro de texto Pendientes -Cuadro de texto Realizó -Cuadro de texto Costo -Botón Agregar
Salidas:	Se agrega el ajuste a la base de datos
Precondiciones:	Tener un cliente y automóvil registrado
Postcondiciones:	Se guarda el ajuste en la base de datos

Tabla 6. Requerimientos no funcionales

Flujo de eventos:

Trayectoria principal:

1. El trabajador elige la opción “carro”.
2. El sistema muestra la pantalla “Carro”.
3. El trabajador elige la opción “Ajuste”.
4. El sistema muestra la pantalla “Ajuste”.
5. El trabajador elige la opción “Agregar ajuste”.
6. El sistema muestra la pantalla “Agregar ajuste”.
7. El trabajador ingresa el numero de placa de auto.
8. El trabajador selecciona el tipo de ajuste.
9. El trabajador ingresa la descripción del ajuste.
10. El trabajador ingresa los pendientes.
11. El trabajador ingresa el nombre de la persona que realizó el registro del ajuste.
12. El trabajador ingresa el costo del ajuste.
13. El trabajador oprime el botón “Agregar”.

--Fin de trayectoria

Realizar nota del servicio realizado

Caso de uso:	Realizar nota del servicio realizado
Versión:	1.0
Actor(es):	Trabajador

Propósito:	Generar un documento pdf en el cual se realizará la nota de todas las refracciones puesta y ajustes que se hicieron al automóvil del cliente
Resumen:	generar una nota en formato pdf
Entradas:	-Cuadro de texto Cliente. -Barra de selección Tipo de ajuste -Cuadro de texto Descripción -Cuadro de texto Costo -Cuadro de texto Número de piezas -Cuadro de texto Nombre -Cuadro de texto Costo(Refracción) -Botón Agregar ajuste -Botón Agregar refracción -Botón Generar PDF
Salidas:	Realización de la nota en formato PDF
Precondiciones:	Tener un cliente al que se le haya realizado un ajuste o puesto una refacción
Postcondiciones:	Se genera un formato PDF con la nota

Tabla 7. Requerimientos no funcionales

Flujo de eventos:

Trayectoria principal:

1. El trabajador elige la opción “Nota” en la pantalla de Ajuste.
2. El sistema muestra la pantalla “Nota”.
3. El trabajador agrega un ajuste. [Trayectoria A]
4. El trabajador agrega una refracción. [Trayectoria B]
5. El trabajador oprime el botón “Generar PDF”.

--Fin de trayectoria

Trayectoria A (El administrador agrega un ajuste)

1. El trabajador ingresa el nombre del cliente.
2. El trabajador selecciona el tipo de ajuste.
3. El trabajador ingresa ingresa la descripción del ajuste.
4. El trabajador ingresa el costo del ajuste.
5. El trabajador oprime el botón Agregar ajuste.

--Fin de trayectoria

Trayectoria B (El administrador agrega una refracción)

1. El trabajador ingresa el nombre del cliente.
2. El trabajador selecciona el tipo de ajuste.
3. El trabajador ingresa ingresa la descripción del ajuste.
4. El trabajador ingresa el costo del ajuste.

5. El trabajador oprime el botón Agregar ajuste.

--Fin de trayectoria

Enviar nota al correo del cliente.

Caso de uso:	Enviar nota al correo del cliente
Versión:	1.0
Actor(es):	Trabajador
Propósito:	Enviar via correo electronico la nota generada en formato PDF de los ajustes realizados y refacciones puestas al automóvil del cliente
Resumen:	Enviar la nota en formato PDF al correo del cliente
Entradas:	-Botón Enviar PDF
Salidas:	Envío de la nota formato PDF al correo del cliente
Precondiciones:	Tener una nota generada en formato PDF
Postcondiciones:	Se envía un formato PDF con la nota del cliente

Tabla 8. Requerimientos no funcionales

Flujo de eventos:

Trayectoria principal:

1. El Trabajador presiona el botón “Enviar PDF” en la pantalla de Nota

--Fin de trayectoria

Recibir reparaciones

Caso de uso:	Recibir reparaciones
Versión:	1.0
Actor(es):	Trabajador, Automóvil
Propósito:	El trabajador realizará los ajustes, que hayan sido previamente registrados en el sistema, al automóvil
Resumen:	El auto recibirá las reparaciones
Entradas:	ajustes registrados en el sistema
Salidas:	Confirmación de que se ha acabado de hacer el ajuste
Precondiciones:	Que el auto tenga reparaciones pendientes registradas en el sistema
Postcondiciones:	El auto es reparado

Tabla 9. Requerimientos no funcionales

Flujo de eventos:

Trayectoria principal:

1. El trabajador revisa los ajustes/mantenimiento pendiente del carro
2. El trabajador realiza los ajustes al automóvil

--Fin de trayectoria

Diagrama de clases.

EL diagrama de clases de nuestro sistema cuenta con 7 clases, las cuales son: Administrador, Trabajador, Cliente, Autos, Registro, Nota y Pago. la Figura 3 muestra cómo es la estructura del diagrama de clases completo de nuestro sistema y posteriormente se explicara cada una de sus clases y relaciones a detalle

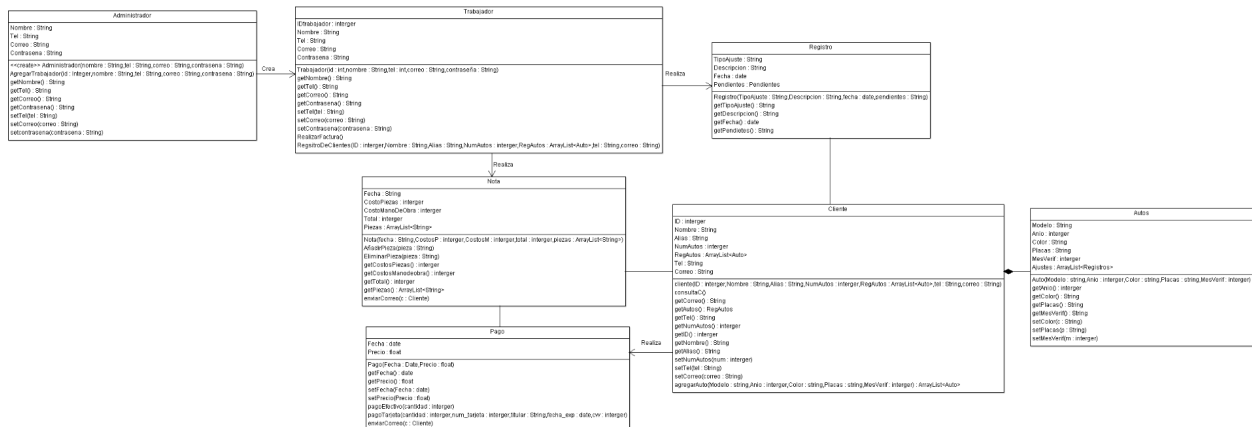


Figura 6. Diagrama de clases

En esta primera parte del diagrama (Figura 4) tenemos a la clase Administrador y la clase Trabajador.

La clase administrador cuenta con los atributos nombre, tel (el teléfono del administrador), correo y contraseña y son de tipo string, con estos atributos podemos identificar al administrador. Cuenta con los métodos agregar trabajador, actualizar datos del trabajador, consultar automóvil y sus setter y getter's.

La clase Trabajador cuenta con los atributos IDtrabajador, nombre, tel (el teléfono del trabajador), correo y contraseña y son de tipo string, Cuenta con los métodos registrar cliente, actualizar datos del cliente, registra automóvil nuevo, Agendar cita, registrar ajuste y sus setter y getter's.

La relación que hay entre estas dos clases es una asociación unidireccional la cual va del administrador al trabajador, el administrador es capaz de crear a un trabajador.

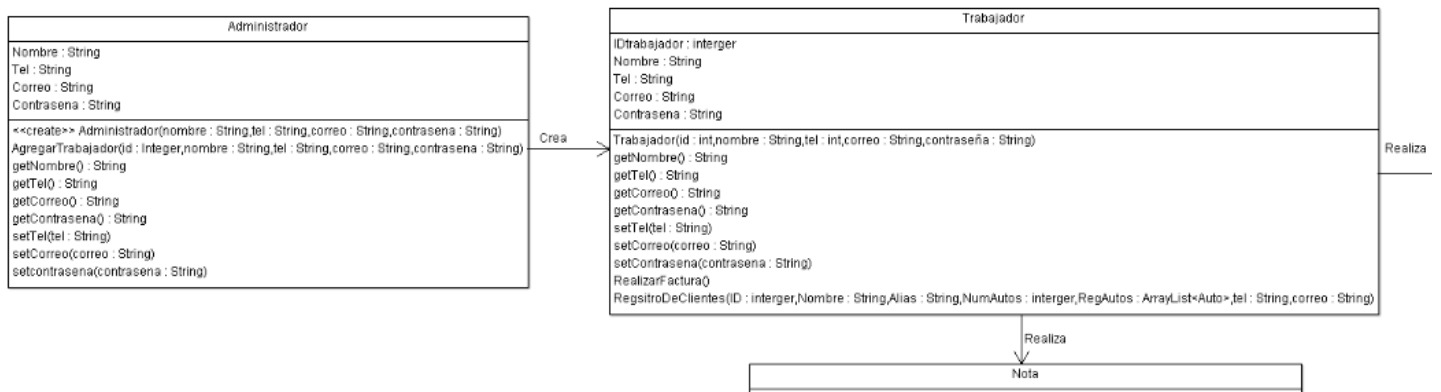


Figura 7. Diagrama de clases parte 1

En la segunda parte (Figura 5) se puede observar la relación “Realiza” que va desde el trabajador a la clase “Registró”, esta clase cuenta con los atributos TipoAjuste(string), Descripcion (string), Fecha(date), Pendientes(string)

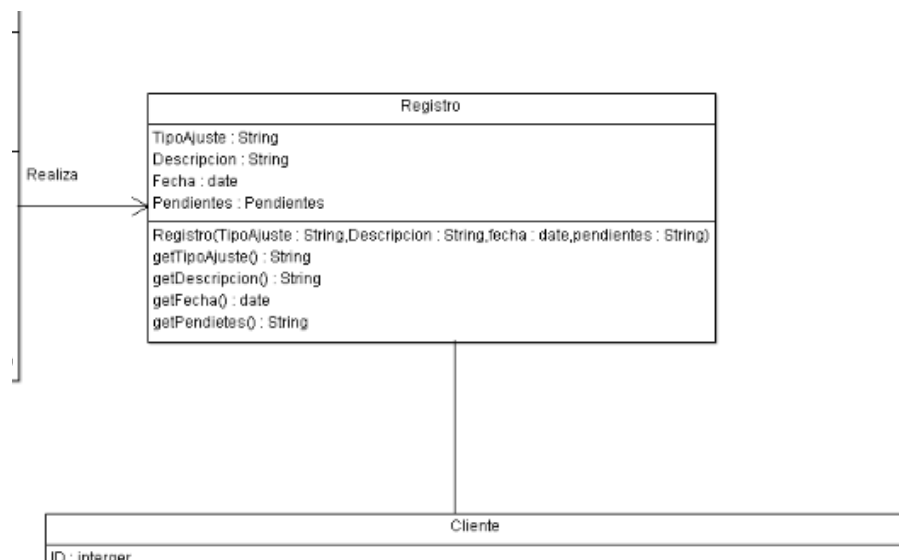


Figura 8. Diagrama de clases parte 2

En la tercera parte (Figura) tenemos a la clase Nota la cual tiene los atributos Fecha(date), Costodepieza (integer), Costomancodeobra (integer), Total(integer), piezas(string) y esta clase nos permitirá realizar la nota con los ajuste y piezas que se utilizaron en la reparacion del automovil.

En la parte superior se observa una relación “Realiza” la cual va de trabajador a nota.

La clase Pago está relacionada con la clase Nota, y esta clase pago cuenta con los atributos Fecha(Date) y Precio(integer) y tiene los métodos PagoEfecticvo el cual recibe una cantidad que será la cantidad que de como total en la nota, y un segundo método PagoTarjeta, el cual recibe el número de tarjeta, el titular, la fecha de expiración y el cvv, los cuales son datos necesarios para poder procesar el pago de la nota.

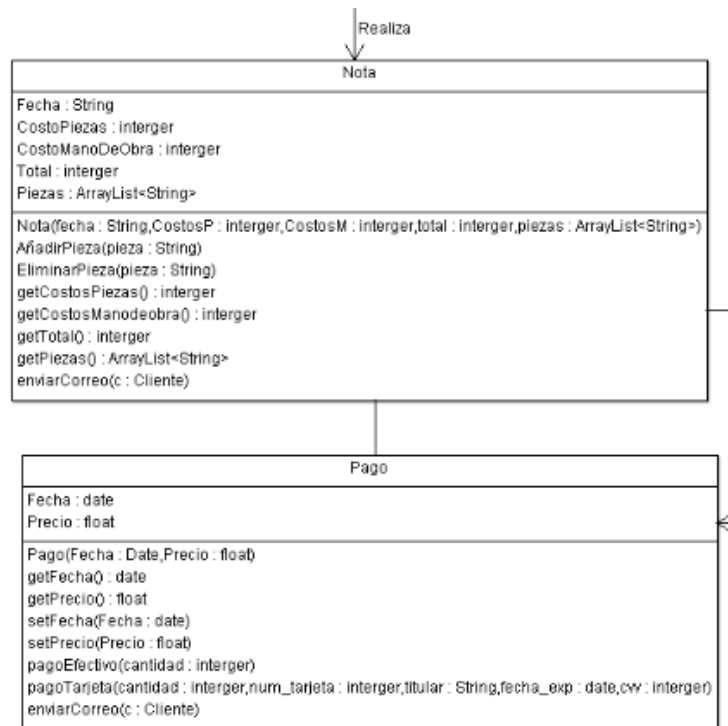


Figura 9. Diagrama de clases parte 3

Por último, en la tercera parte (Figura) tenemos la clase cliente, la cual cuenta con los atributos ID (integer), nombre (string), tel (el teléfono del trabajador), correo (string) y contraseña (string), Alias (string) y NumAutos (integer) y tiene su métodos getter's y setter's

También está la clase auto, la cual cuenta con los atributos Modelo (string), color (string), placas (string), Ajustes (ArrayList<Registros>)

Estas dos clases tienen una relación de composición

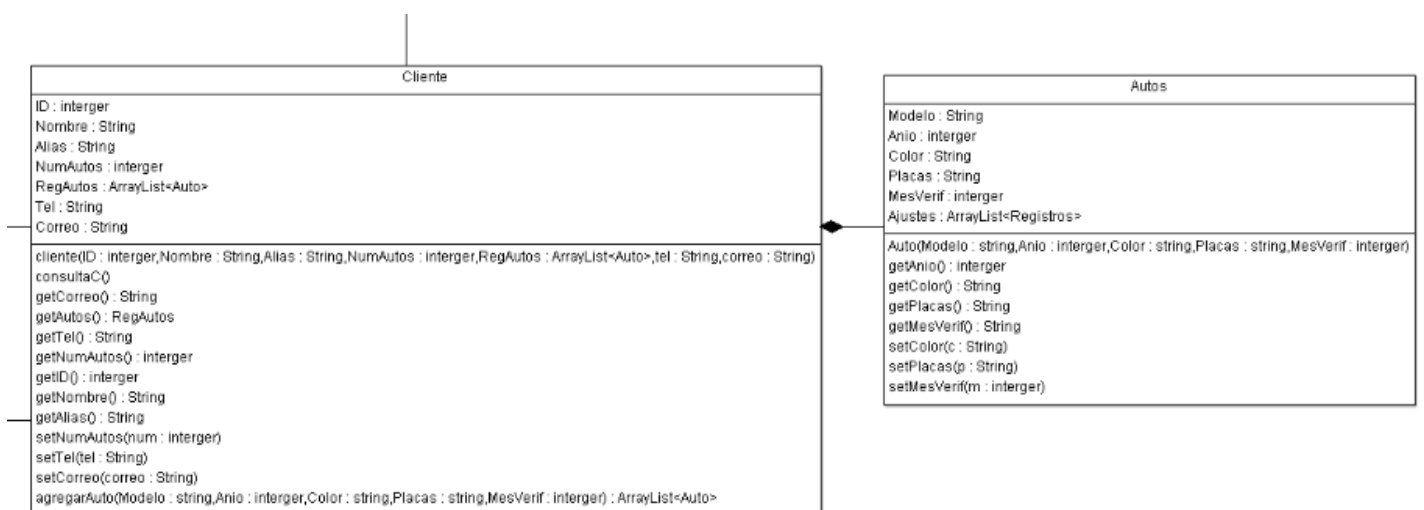


Figura 10. Diagrama de clases parte 4

Diagrama de Estados

En el diagrama de estados se analizan las distintas transiciones que suceden en ciertas actividades desarrolladas a lo largo de los procesos, en este caso se realizó el diagrama de estados del automóvil (Figura) y el de la generación de las citas (Figura 8).

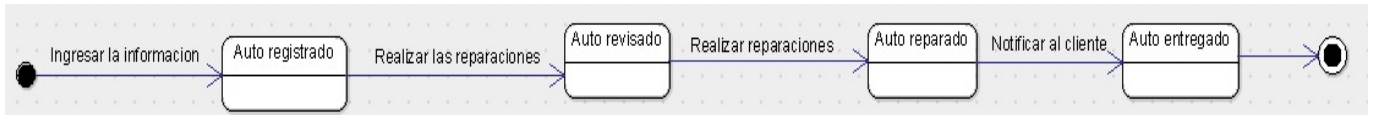


Figura 11. Diagrama de estados: Automóvil

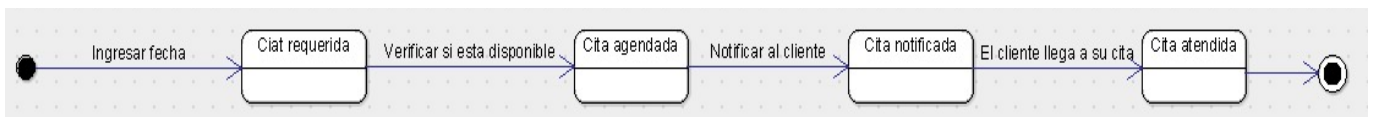


Figura 12. Diagrama de estados: Cita

Diagrama de Secuencias

Este diagrama es usado para modelar la interacción entre objetos en un sistema según UML, se centran específicamente en líneas de vida o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine.

En este primer diagrama se muestra el inicio de sesión en nuestro sistema, el inicio de sesión puede ser realizado por el administrador o el trabajador. En el que el administrador proporciona los datos requeridos por la Interfaz de usuario “Inicio de sesión” y llegan a la base de datos para verificar que los datos enviados pertenezcan a la base y así enviar una confirmación o error según sea el caso

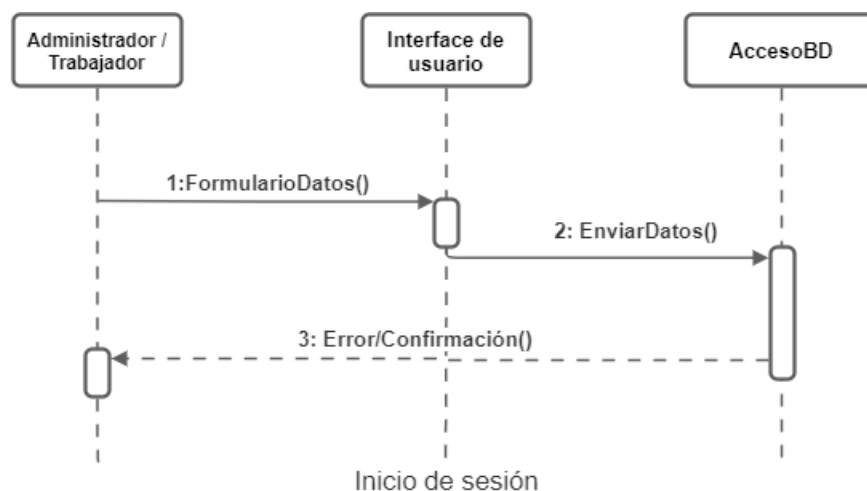


Figura 13. Diagrama de secuencias: Inicio de sesión

Este diagrama muestra el proceso de alta de cliente o agregación de un cliente nuevo, en el que el cliente proporciona sus datos y estos a través de la interfaz de usuario “Agregar cliente” son transmitidos a la base de datos para poder ser guardados.

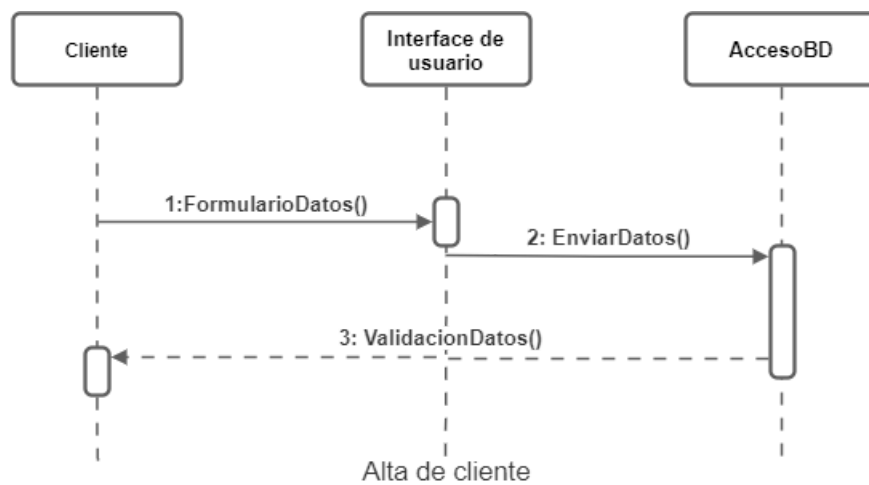


Figura 14.Diagrama de secuencias: Alta de cliente

Este diagrama muestra el proceso de alta de carro o agregación de un carro nuevo, en el que el cliente proporciona los datos del carro en cuestión y estos a través de la interfaz de usuario “Agregar auto” son transmitidos a la base de datos para poder ser guardados.

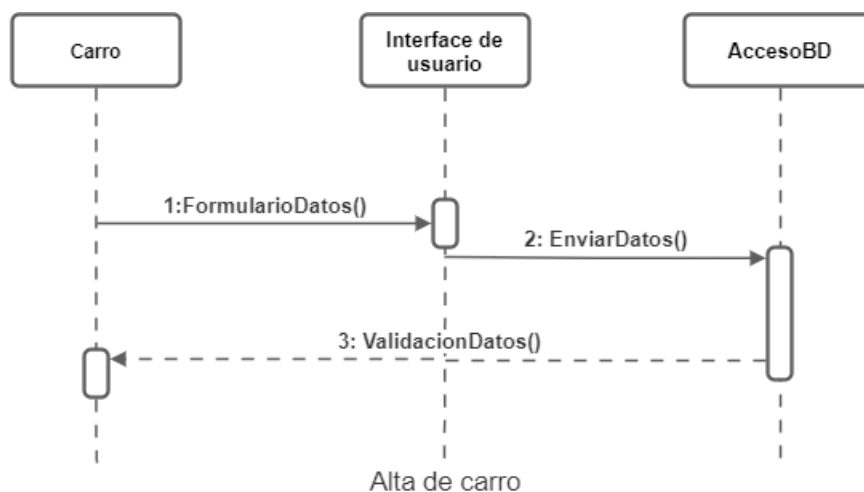


Figura 15.Diagrama de secuencias: Alta de carro

Este diagrama muestra el proceso de consultar el historial del automóvil, en el que el trabajador proporciona los datos requeridos en la interfaz de usuario “Consultar auto”, posteriormente esos datos se revisa si están en la base de datos y esta devuelve los datos requeridos por la operación a la interfaz de usuario “Consultar auto” mostrando así los datos al trabajador.

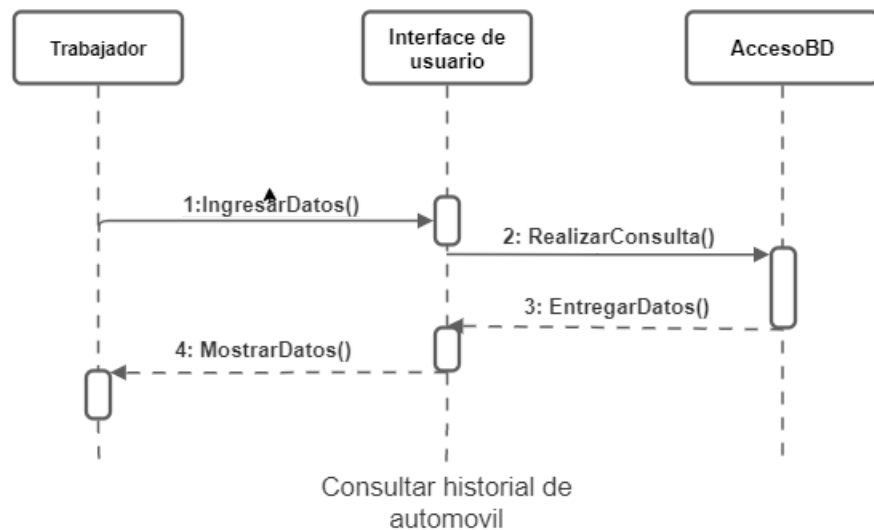


Figura 16. Diagrama de secuencias: Consultar historial del automóvil

En este diagrama para agendar una cita el cliente proporciona los datos de la cita al trabajador y este llena el formulario solicitado por la interfaz de usuario “Agendar cita” y se transportan estos datos a la base de datos para ser guardados y al final se valida que la operación se pudo realizar exitosamente.

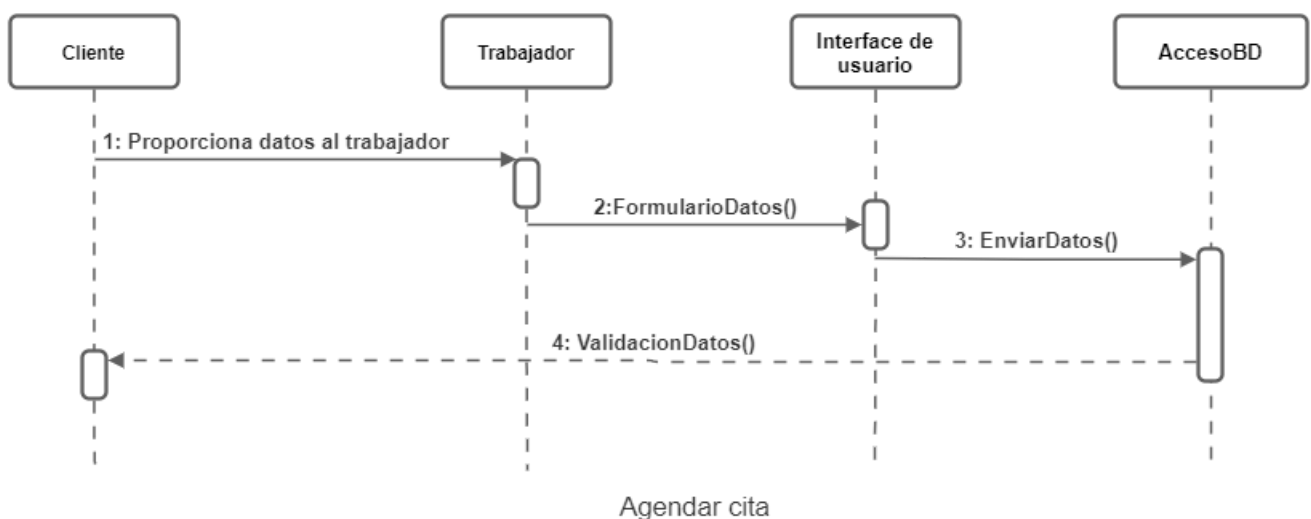


Figura 17. Diagrama de secuencias: Agendar cita

En este diagrama del Pago el cliente especifica el método del pago al trabajador y este llena el formulario para llenar el pago en la interfaz de usuario “Nota” y una vez generada la nota en formato PDF se envía al correo electrónico del cliente.

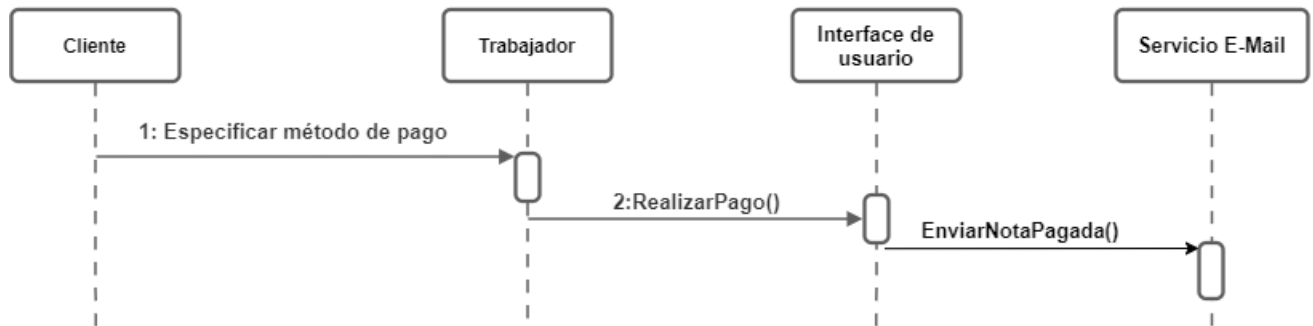


Figura 18. Diagrama de secuencias: Pago

En este diagrama de los Avances el Trabajador registra los avances en la interfaz de usuario y posteriormente este avance se envía al correo del cliente

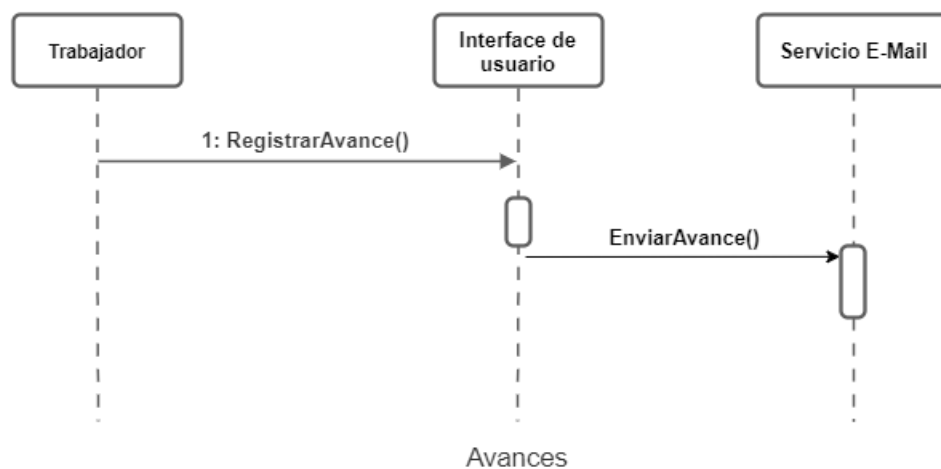


Figura 19. Diagrama de secuencias: Avances

En este Diagrama de recuperación de contraseña o actualización de contraseña el cliente llena el formulario con el cambio de contraseña en la interfaz de usuario “Actualizar datos del cliente” y estos datos se confirman en la base de datos y posteriormente se envía un correo al correo del usuario enviando un codigo de confirmacion y después de que el cliente lo ponga este podrá introducir su contraseña nueva

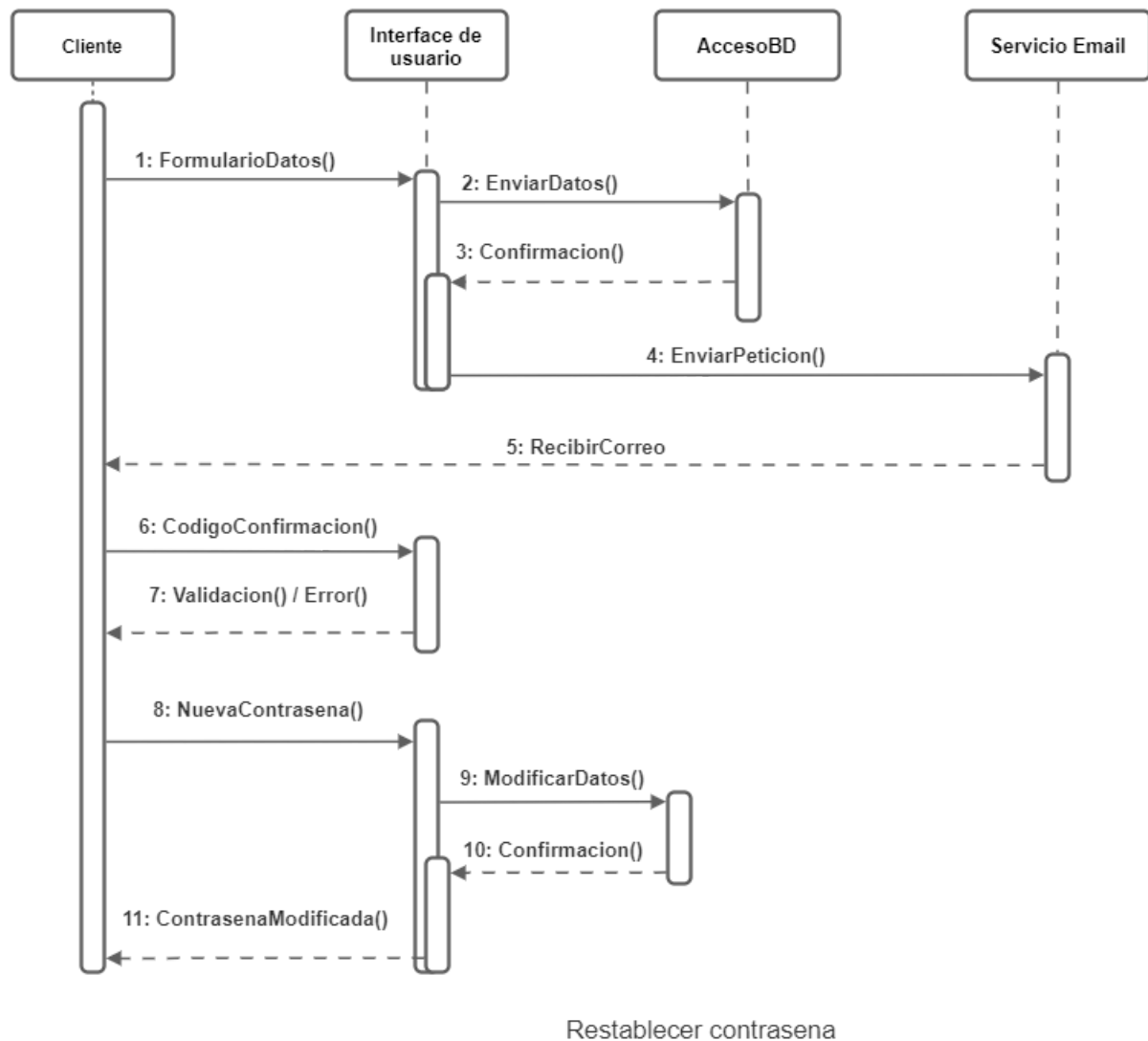


Figura 20. Diagrama de secuencias: Restablecer contraseña

Diagramas de Actividades

Los diagramas de actividades ayudan a que las personas en las áreas de negocios y desarrollo de una organización se integren para comprender el mismo proceso y comportamiento. Se utilizan símbolos especializados, incluidos aquellos para pasos de inicio, finalización, fusión y recepción en el flujo. Las interfaces gráficas utilizadas en los carriles son las que se muestra en el titular del diagrama.

En el diagrama de registrar cliente tenemos los carriles: Cliente, Trabajador, Interfaz gráfica y Base de datos en donde se realiza una validación de los datos en la base de datos con un nodo de decisión para poder realizar el registro.

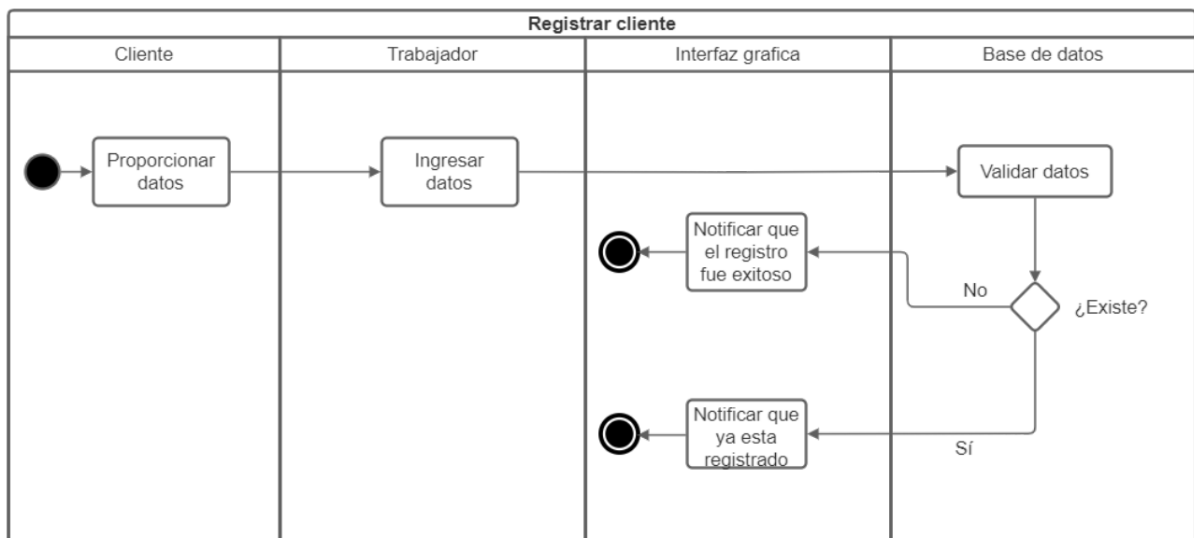


Figura 21. Diagrama de Actividades: Registrar cliente

En el diagrama de registrar al trabajador tenemos los carriles: Trabajador, Administrador, Interfaz gráfica y Base de datos en donde se realiza una validación de los datos en la base de datos con un nodo de decisión para poder realizar el registro.

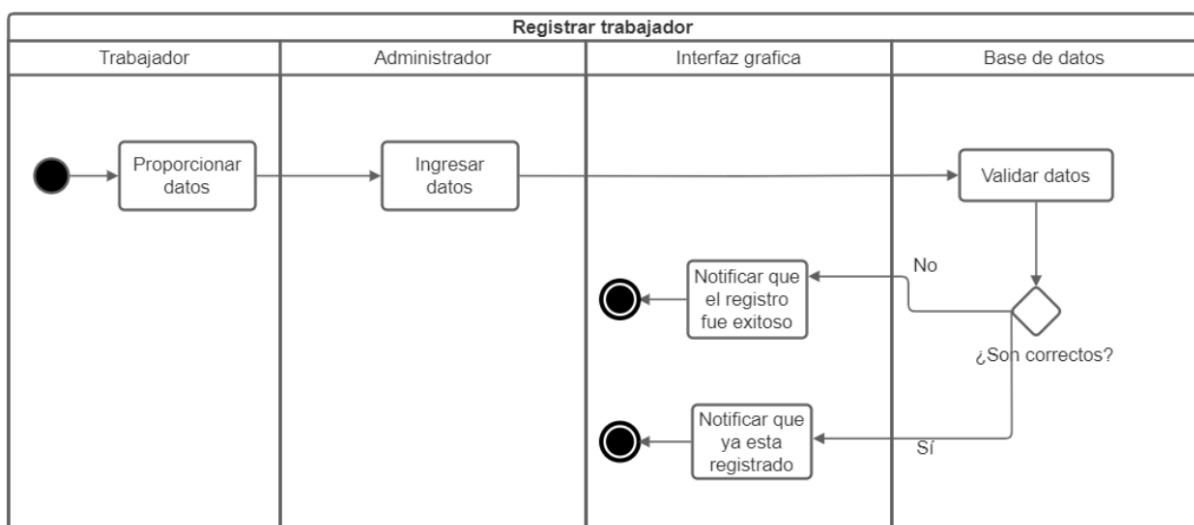


Figura 22. Diagrama de Actividades: Registrar Trabajador

En el diagrama de Consultar automóvil tenemos los carriles Trabajador, Interfaz gráfica y Base de datos en donde se realiza una validación de los datos en la base de datos con un nodo de decisión para corroborar que el automóvil a consultar se encuentre en la base de datos.

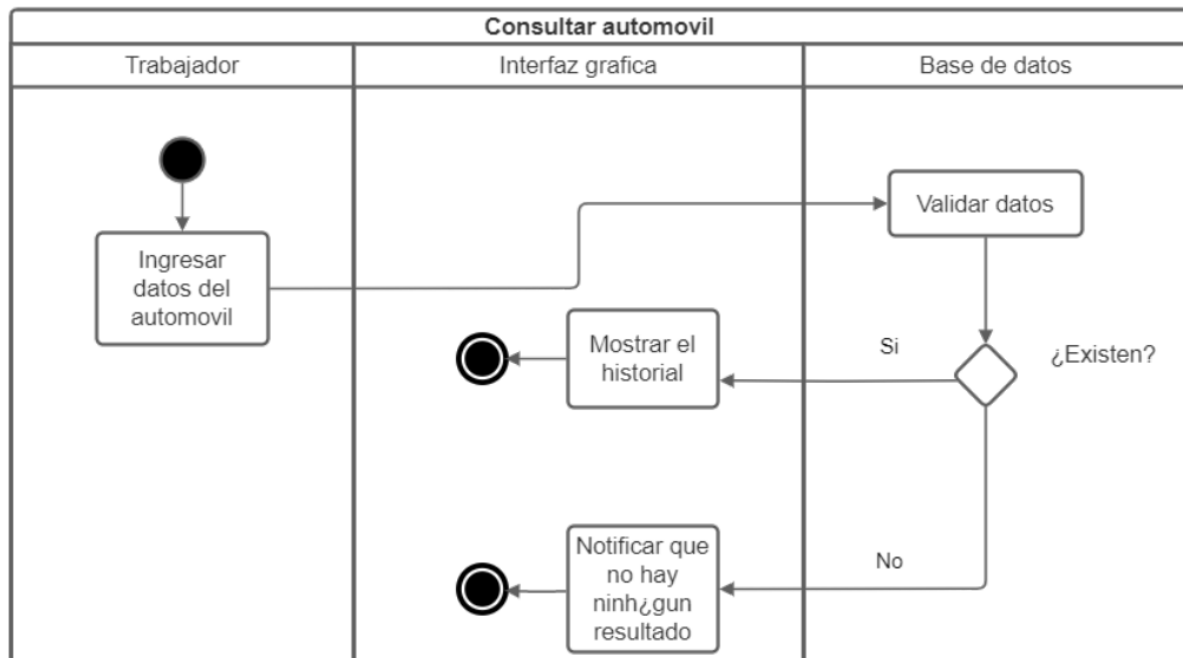


Figura 23. Diagrama de Actividades: Consultar automóvil

En el diagrama de Agendar cita tenemos los carriles Cliente, Trabajador, Interfaz gráfica y Base de datos en donde se realiza una validación de los datos en la base de datos con un nodo de decisión para corroborar que la fecha deseada está disponible

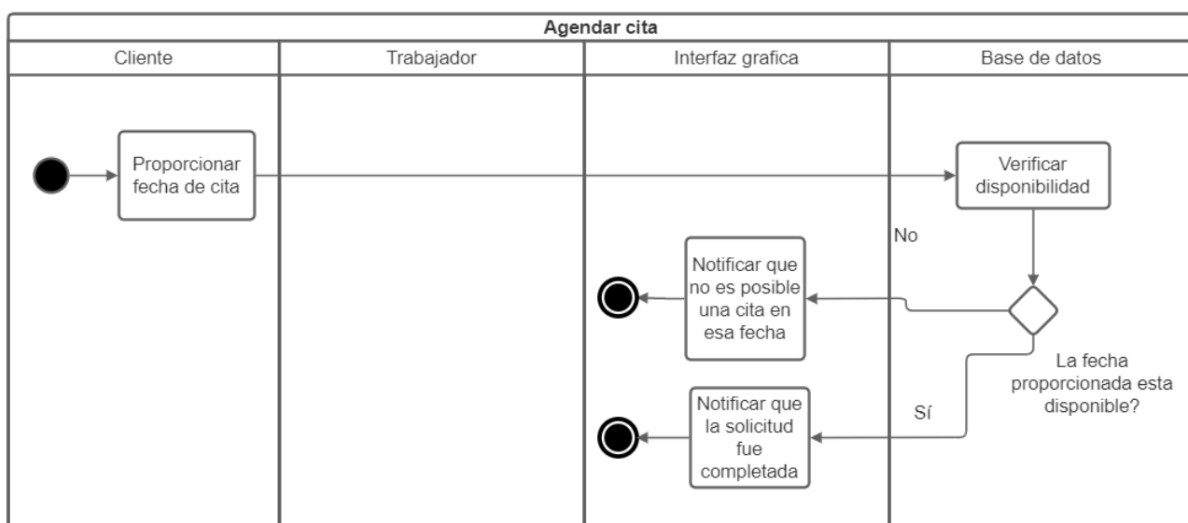


Figura 24. Diagrama de Actividades: Agendar cita

En el diagrama de Iniciar sesión tenemos los carriles trabajador, Interfaz gráfica y Base de datos en donde se realiza una validación de los datos en la base de datos, los cuales serian los campos de usuario y contraseña, si estos campos no son correctos el usuario puede pedir un cambio de contraseña para poder ingresar la sistema.

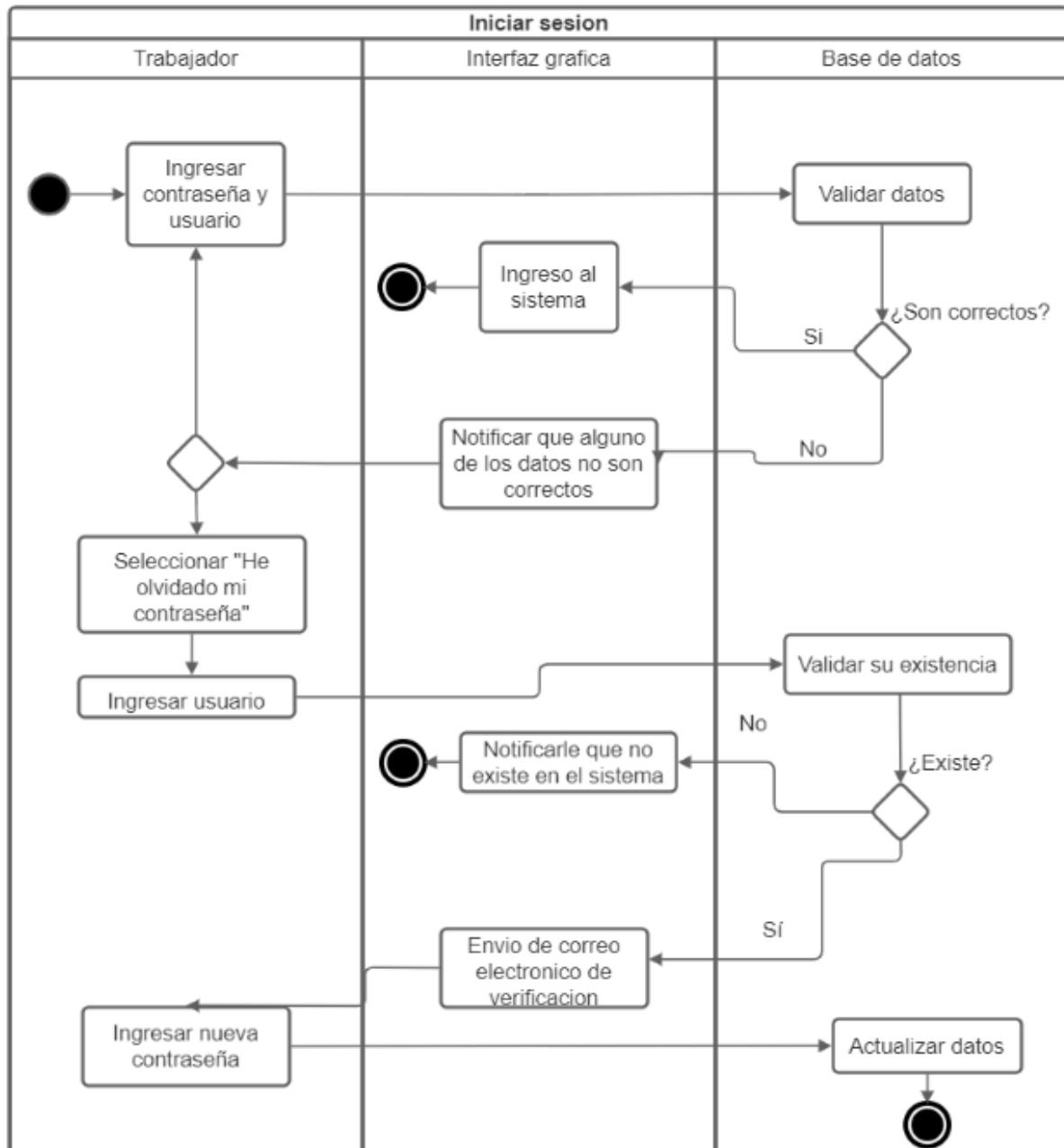


Figura 25. Diagrama de Actividades: Iniciar sesión

En el diagrama de Notificación de ajuste tenemos los carriles Cliente, Trabajador, Interfaz gráfica y Base de datos en donde se realiza una validación de los datos en la base de datos con un nodo de decisión para revisar si existe el auto en la base de datos y así poder enviar la notificación a su respectivo cliente

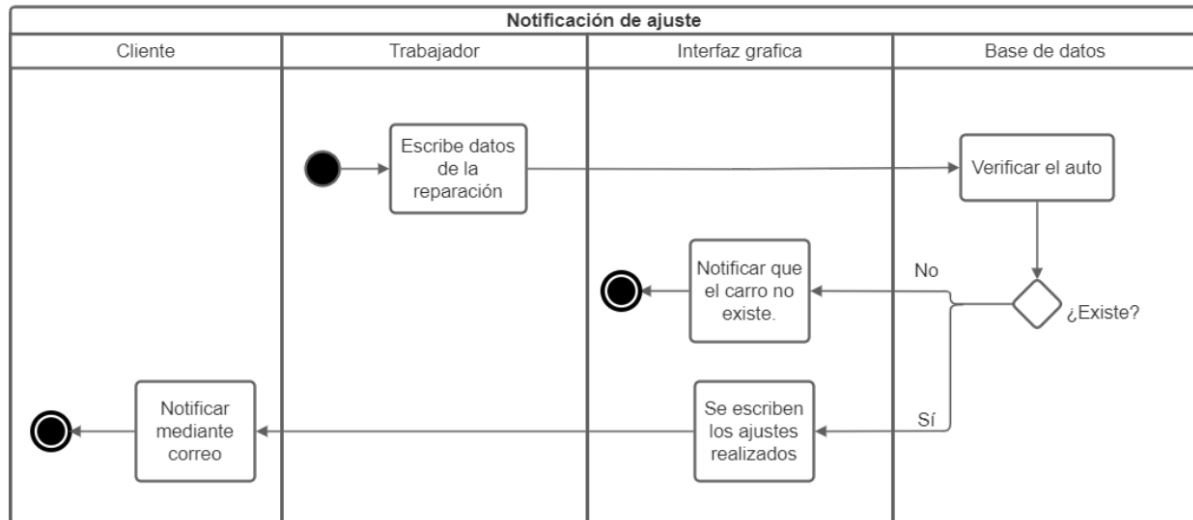


Figura 26. Diagrama de Actividades: Notificación de ajuste

En el diagrama de registrar cliente tenemos los carriles Trabajador, Cliente, Interfaz gráfica y Base de datos en donde se realiza una decisión del método de pago a utilizar y en base a eso seguir el flujo correspondiente para poder realizar la operación de pago.

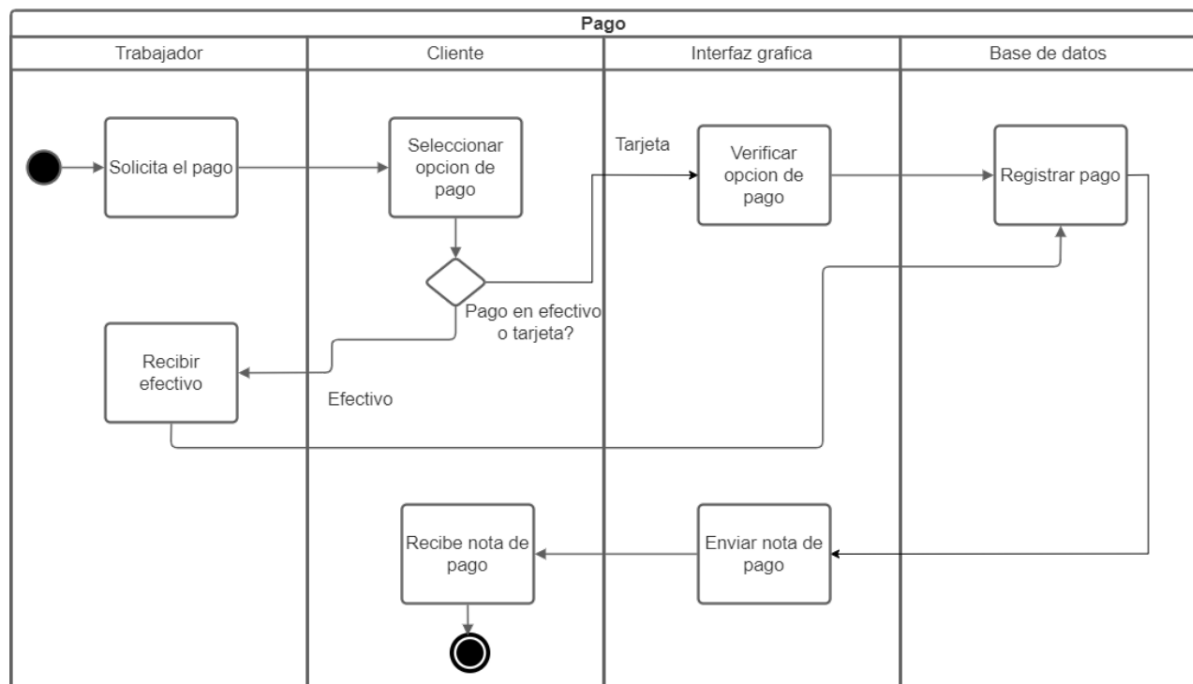


Figura 27. Diagrama de Actividades: Pago

CAPÍTULO 4. DISEÑO

Pantalla de inicio de sesión.

En la Figura 28 se puede observar el diseño de la pantalla de inicio de sesión del sistema, el cual cuenta con nuestro logo en la parte izquierda y en la parte derecha los contenedores para poner la información requerida para el inicio de sesión, en este caso para iniciar sesión se necesita un correo electrónico y contraseña, Si alguno de estos campos es incorrecto se muestra en pantalla un mensaje que dice “Verifique los datos que ingreso, alguno es incorrecto” como se muestra en la Figura 29. Las sesiones con las que se cuenta son de administrador y trabajador, dependiendo de cómo se haya registrado el correo que se coloque en el inicio de sesión aparecerá la interfaz de trabajador o administrador según sea el caso, y esto se logra con la base de datos, la cual nos indica si la sesión iniciada es de administrador o trabajador.

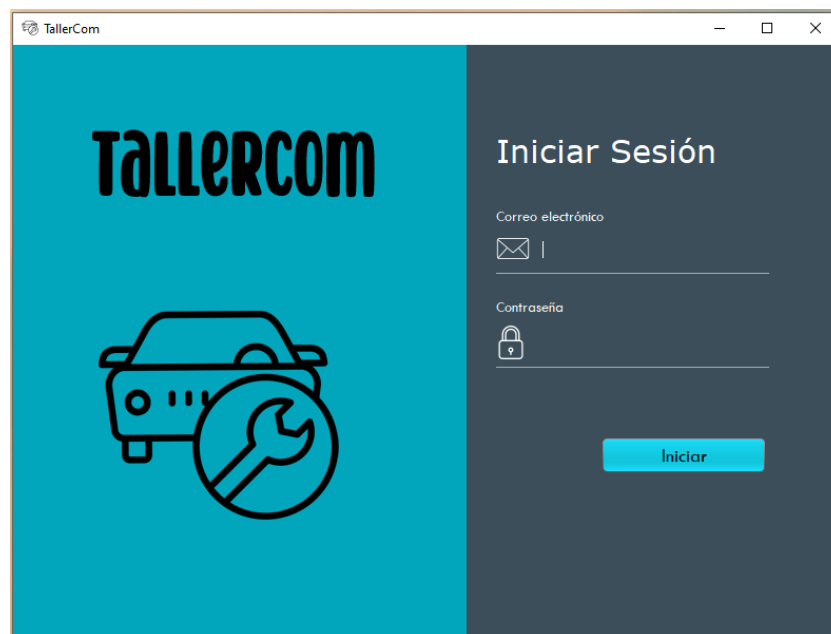


Figura 28. Inicio Sesión

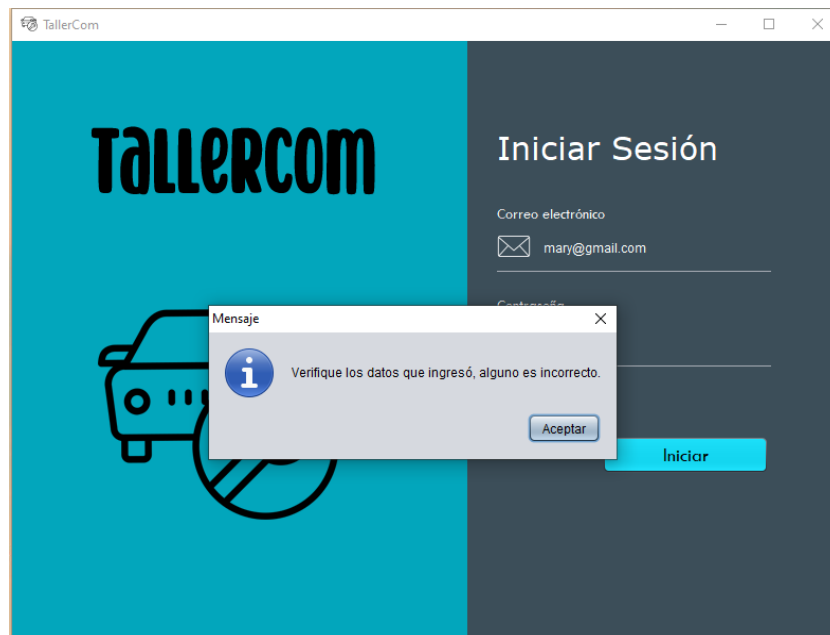


Figura 29. Inicio Sesión Error

Sesión de Administrador

Esta es la pantalla que muestra el sistema al ingresar con una sesión de tipo administrador (Figura 30). En la parte izquierda muestra un saludo y el nombre registrado de la persona que inició la sesión, en la izquierda se encuentran seis opciones con su respectivo símbolo, las cuales son: Agregar trabajador, Editar trabajador, consultar trabajador, editar trabajador, Cliente y Carro, al seleccionar alguna de estas seis opciones se nos mostrará la pantalla correspondiente a la opción seleccionada.

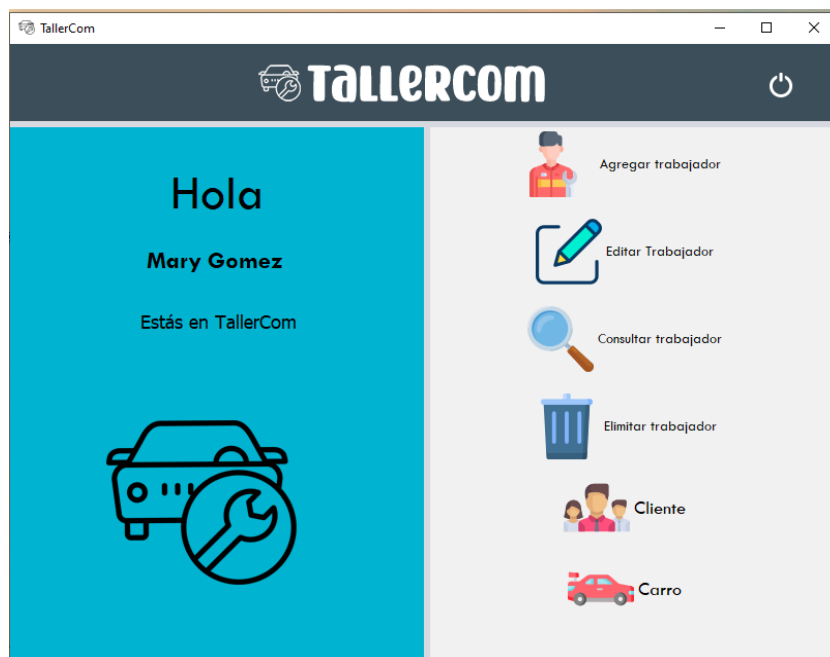


Figura 30. Inicio de la sesión Administrador

Al entrar en la opción “Agrega trabajador” del lado derecho aparecen los campos que se necesitan llenar para poder agregar a un nuevo trabajador, así como un botón de “Agregar” para que una vez que los campos hayan sido llenados de manera correcta se proceda a agregar al trabajador, si alguno de los datos es erróneo el sistema mostrará una imagen de una “x” indicando que sucedió un error al momento de ingresar los datos como se muestra en la Figura 31, y si los datos son correctos aparece la imagen de una paloma indicando que el registro fue completado correctamente como se muestra en la Figura 32

The screenshot shows the 'Agregar trabajador' (Add worker) form in the TallerCom application. The form is titled 'Agregar trabajador' and includes a sub-header 'Verifique que los datos ingresados sean correctos.' (Verify that the entered data is correct). The form fields are: Nombre (Name), Rango (Rank) with a dropdown menu set to 'Administrador', Teléfono (Phone), Correo (Email), and Contraseña (Password). A large red circle with a white 'X' is displayed on the left side of the form, indicating an error. A blue 'Agregar' (Add) button is located at the bottom right of the form. The background is light blue.

Figura 31. Agregar trabajador Error

The screenshot shows the 'Agregar trabajador' (Add worker) form in the TallerCom application. The form is titled 'Agregar trabajador' and includes a sub-header 'Se agregó al trabajador exitosamente.' (Worker added successfully). The form fields are: Nombre (Name) with the value 'Heidi Figueroa', Rango (Rank) with a dropdown menu set to 'Trabajador', Teléfono (Phone) with the value '5511334455', Correo (Email) with the value 'heidi@gmail.com', and Contraseña (Password) with the value '*****'. A large green circle with a white checkmark is displayed on the left side of the form, indicating success. A blue 'Agregar' (Add) button is located at the bottom right of the form. The background is light blue.

Figura 32. Agregar trabajador correcto

Al entrar en la opción “Consultar trabajador” en la parte derecha se muestra un campo para colocar el correo del trabajador del cual se desea realizar la consulta y un botón denominado “consulta” el cual se presiona después de haber llenado el campo “Correo”, si este dato es correcto en la parte izquierda se mostrarán los datos del trabajador (Figura 33) y si el correo fue escrito incorrectamente o no se cuenta con el en la base de datos aparece una imagen con una “X” en la parte izquierda (Figura 34).

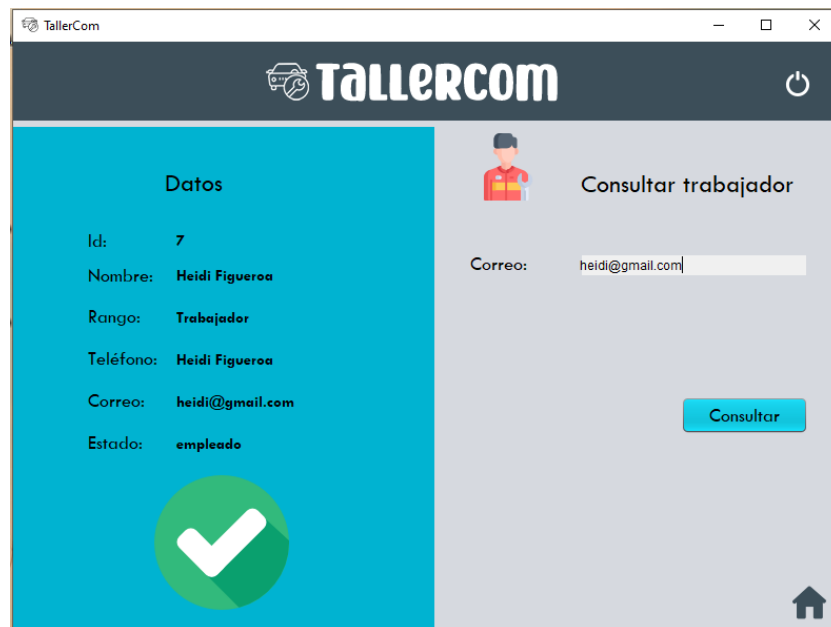


Figura 33.Consultar trabajador correcto

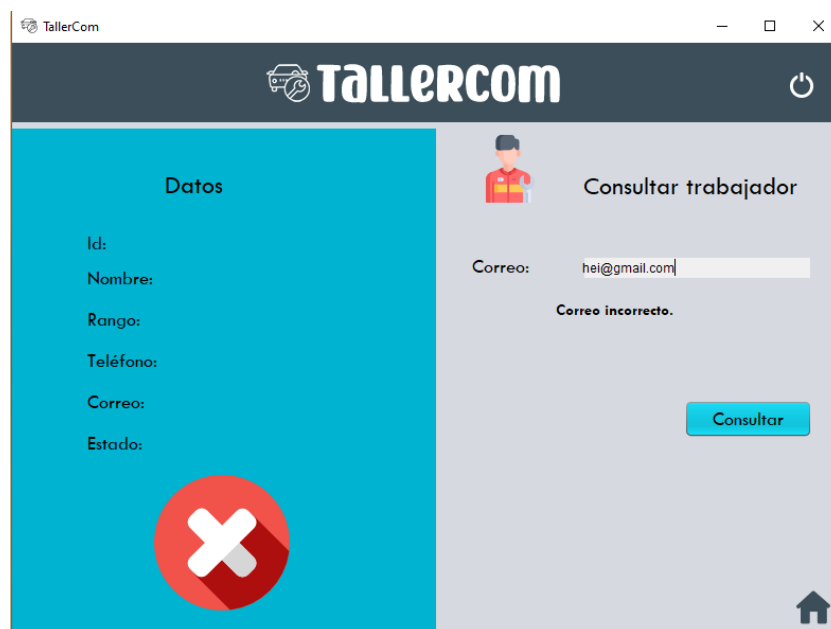


Figura 34.Consultar trabajador incorrecto

Al entrar en la opción “Editar trabajador” en la parte derecha nos aparece el correo del trabajador al que se le desea editar los datos y los campos de edición disponibles que son: Nombre, cargo, teléfono, correo y contraseña. también se muestra un botón denominado “Editar”, si el correo colocado no está guardado en la base de datos este mostrará un error (Figura 35) representado con una imagen colocada en la parte izquierda y si el correo es correcto (Figura 36) aparece el texto “Se editó _____ exitosamente” en donde el espacio varía dependiendo a qué sección se haya editado.

The screenshot shows the 'Editar trabajador' interface in the TallerCom application. On the left, a blue panel displays a red circle with a white 'X' and the text 'Verifique que los datos ingresados sean correctos.' On the right, a light gray panel contains a form. At the top, it asks '¿Qué desea editar?' with a dropdown menu set to 'Rango'. Below this, the 'Correo:' field is populated with 'hei@gmail.com'. A section titled 'Datos a editar' with a pencil icon contains fields for 'Nombre:', 'Rango:' (set to 'Administrador'), 'Teléfono:', 'Correo:', and 'Contraseña:'. A blue 'Editar' button is at the bottom of the form. A home icon is visible in the bottom right corner.

Figura 35. Editar trabajador correcto

This screenshot shows the same 'Editar trabajador' interface as Figure 35, but in a success state. The left blue panel now features a green circle with a white checkmark and the text 'Se editó el carro exitosamente.' The right gray panel and its form elements remain identical to the previous figure.

Figura 36. Editar trabajador incorrecto

La opción eliminar en la sesión de administrador nos permite eliminar trabajadores, clientes o autos. en la pantalla se muestra del lado derecho un campo de elegir una opción, que puede ser cualquiera de las tres antes mencionadas y posteriormente hay un cuadro de texto en el que se puede ingresar el correo, placa o alias dependiendo de la opción a borrar seleccionada, y por último, cuenta con el botón “Eliminar” en el que si no hay ningún error marca con una imagen de una palomita que se eliminó exitosamente la opción seleccionada (Figura 37) y si hubo algún error metiendo los datos aparecerá una imagen de una “X” en la parte izquierda de la pantalla (Figura 38).

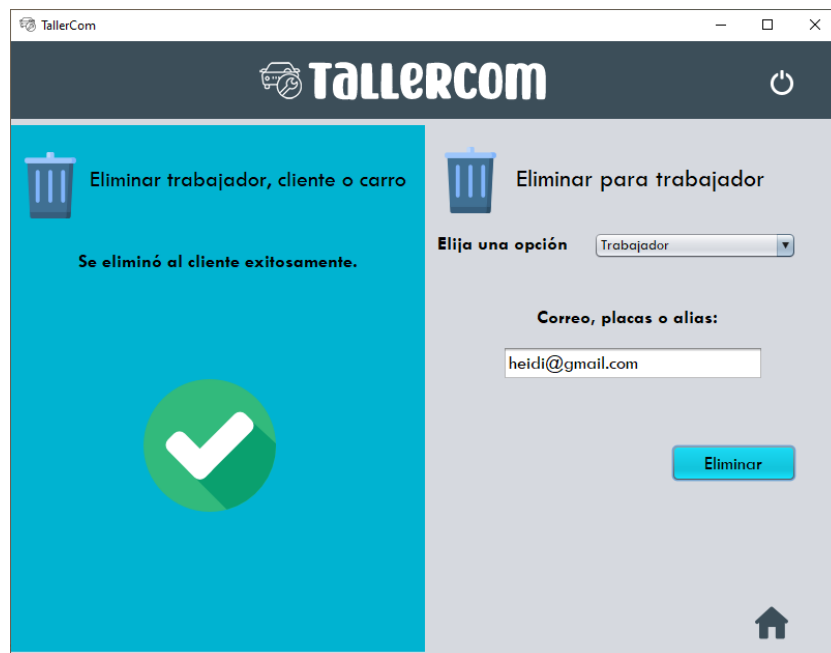


Figura 37. Eliminar trabajador correcto

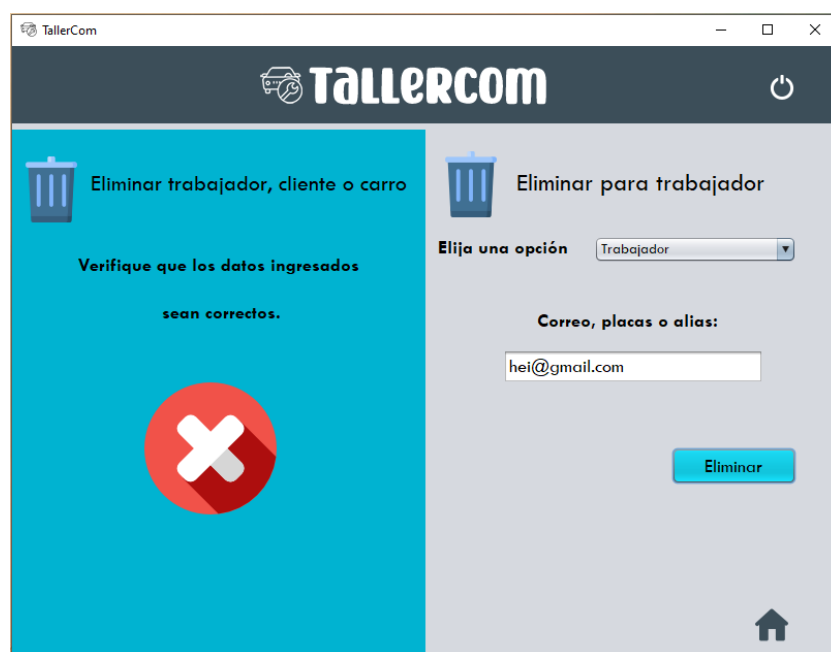


Figura 38. Eliminar trabajador incorrecto

Sesión de Trabajador

Esta es la pantalla que muestra el sistema al ingresar con una sesión de tipo Trabajador (Figura 39), como se puede observar es muy similar a la pantalla de la sesión del administrador pero esta pantalla no tiene la opción “Agregar trabajador”, solo tiene las opciones Agregar cliente, Consultar cliente, editar cliente, eliminar cliente y carro

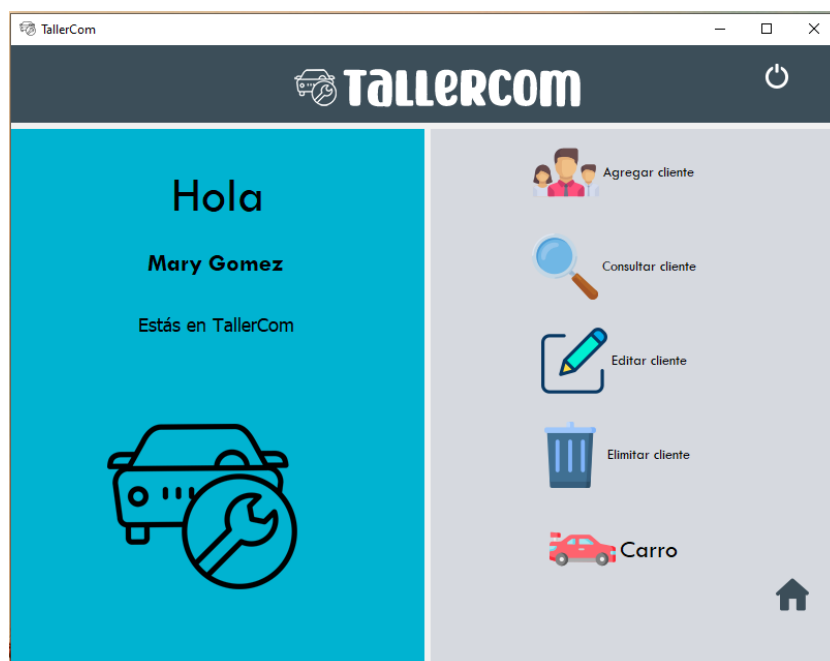


Figura 39. Inicio de la sesion Trabajador

La pantalla Agregar cliente le permite al trabajador registrar al cliente, guardando sus datos en la base de datos, en la parte derecha aparecen los campos de texto nombre, alias, teléfono, correo y número de carros, los cuales serán proporcionados por el cliente. También cuenta con el botón “Agregar”, si los campos están rellenos correctamente se agregara el cliente exitosamente como se muestra en la Figura 40, de lo contrario no se guardará ningún dato del cliente y aparecerá un error en la pantalla en el lado derecho de esta

Figura 40. Pantalla para agregar cliente

Al entrar en la opción “Consultar cliente” en la parte derecha se muestra un campo para colocar el alias del cliente del cual se desea realizar la consulta y un botón denominado “Consultar” el cual se presiona después de haber llenado el campo “Alias”, si este dato es correcto en la parte izquierda se mostrarán los datos del cliente (Figura 41) y si el correo fue escrito incorrectamente o no se cuenta con el en la base de datos aparece una imagen con una “X” en la parte izquierda y no se mostrará ningún dato

Figura 41. Pantalla para consultar cliente

Al entrar en la opción “Editar cliente” en la parte derecha nos aparece el campo “Cliente ”en el que se escribirá el cliente al que se le desea editar los datos y los campos de edición disponibles que son: Nombre, Alias, teléfono, correo y numero de carros, también se muestra un botón denominado “Editar”, si el cliente colocado no está registrado en la base de datos este mostrará un error representado con una imagen colocada en la parte izquierda y si el correo es correcto (Figura 42) aparece el texto “Se editó _____ exitosamente” en donde el espacio varía dependiendo a qué sección se haya editado



Figura 42. Pantalla para editar cliente

La opción eliminar en la sesión del trabajador nos permite eliminar clientes o autos. En la pantalla se muestra del lado derecho un campo de elegir una opción, que puede ser cualquiera de las dos opciones antes mencionadas y posteriormente hay un cuadro de texto en el que se puede ingresar el correo, placa o alias dependiendo de la opción a borrar seleccionada, y por último, cuenta con el botón “Eliminar” en el que si no hay ningún error marca con una imagen de una palomita que se eliminó exitosamente la opción seleccionada (Figura 43) y si hubo algún error metiendo los datos aparecerá una imagen de una “X” en la parte izquierda de la pantalla.

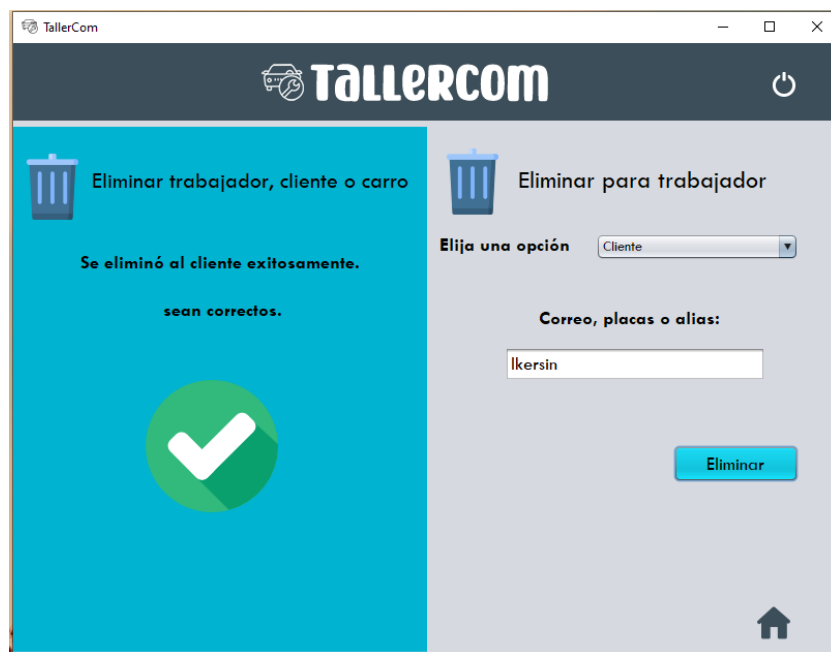


Figura 43. Pantalla para eliminar cliente o carro

La opción Carro en la sesión de administrador y trabajador nos permite realizar las mismas acciones, las cuales son: agregar carro, consultar carro, editar carro y ajuste. estas opciones aparecen en la parte derecha de la pantalla con sus respectivos iconos como se muestra en la Figura 44.

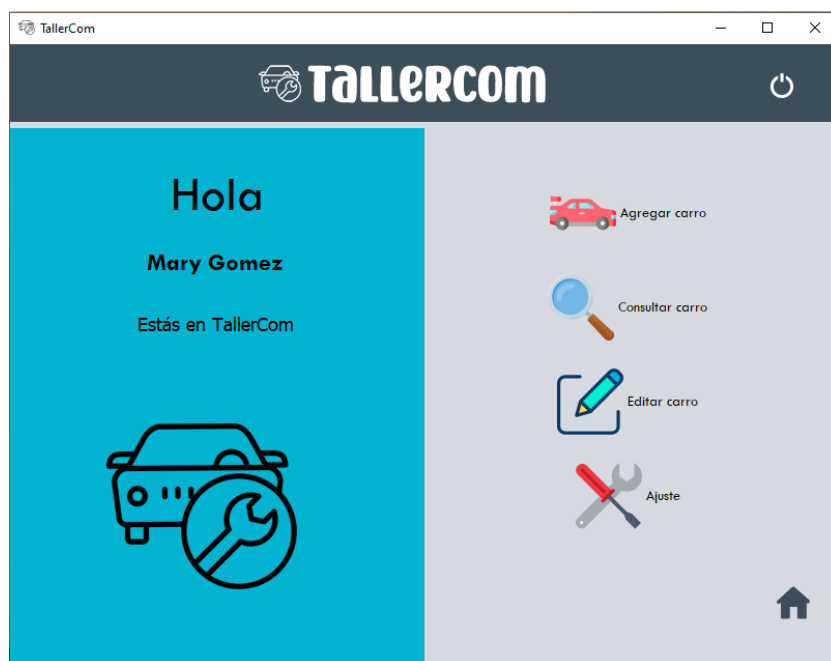


Figura 44. Pantalla para mostrar opciones de carro

La pantalla para agregar carro cuenta con los campos de texto propietario, placas, alias, modelo, y una lista de opciones para elegir el mes de verificación, también cuenta con el botón “Agregar” en el que si uno de los campos es incorrecto o el propietario no está registrado en la base de datos marcara un error en la parte derecha de la pantalla y si todos los campos fueron rellenos correctamente se agregara el carro exitosamente(Figura 45).

Figura 45. Pantalla para agregar carro

La pantalla consultar auto cuenta con un campo de texto denominado “Alias”, en el que se tiene que ingresar el alias del cliente del auto que quiere ser consultado, y un botón “Consulta”, en el que si el campo alias está registrado en la base de datos se muestra en pantalla en formato de tabla toda la información relacionada con los autos del cliente del que se especificó su alias y se mostrará la imagen de una palomita indicando que la búsqueda fue exitosa (Figura 46) de lo contrario se mostrará la imagen de una “X”

Figura 46. Pantalla para consultar carro

La pantalla de editar carro cuenta con la opción desplegable de “¿Que se desea editar?” y los datos a editar pueden ser las placas, el color y el mes de verificación, para poder identificar que carro sera al que se le editaran los datos se escribe la Placa de este en el campo “Placa”, si la placa colocada en el campo es correcta y se encuentra en la base de datos entonces se editará exitosamente el campo que se haya seleccionado (Figura 47), de lo contrario aparecerá una imagen de error en la parte izquierda de la pantalla



Figura 47. Pantalla para editar carro

Pantalla de Ajuste

Al seleccionar la opción de ajuste se despliega otra pantalla en la que se muestran las opciones Agregar ajuste, Consultar ajuste, Nota y Cita. Están situados en la parte derecha de la pantalla con sus respectivos iconos



Figura 48. Pantalla para mostrar opciones de ajuste

Al seleccionar la opción de Agregar ajuste se muestra la pantalla de la Figura 49, en la que del lado derecho tenemos los campos de texto placas, descripción, pendientes, realizó y costo, también tiene una lista de opciones de tipos de ajustes para ser realizados y el botón “Agregar” en el que si las placas del auto son correctas y se encuentran registradas en la base de datos se agregará el ajuste exitosamente, de lo contrario se marcará un error.

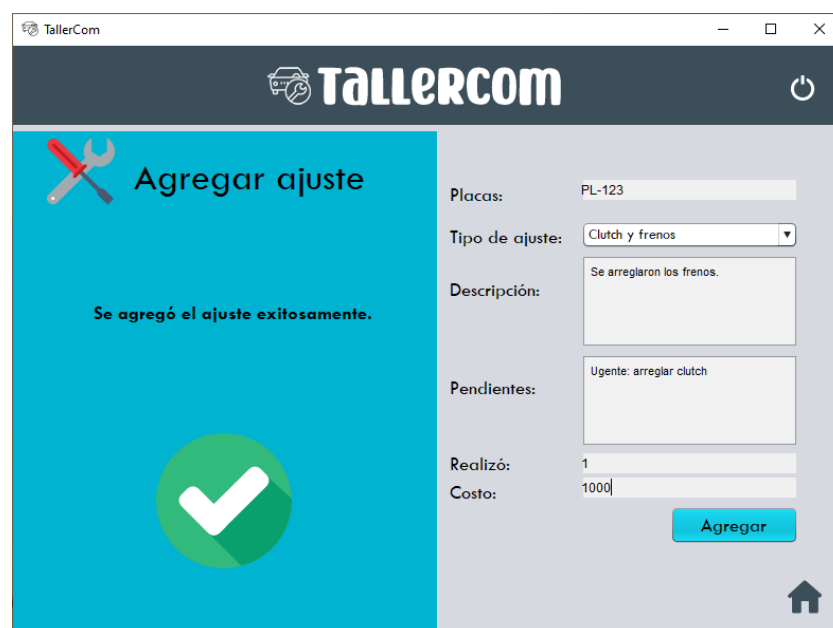


Figura 49. Pantalla para agregar ajustes

La opción Consultar ajuste nos permite visualizar los ajustes registrados en un automóvil, para esto necesitamos escribir las placas del carro en cuestión para poder visualizar sus ajustes, esta pantalla también cuenta con el botón “Consultar”, en el que una vez llenado el campo placas nos desplegara en la tabla que se muestra en la Figura 50, solo si este campo es llenado correctamente y la placa esté registrada en la base de datos de lo contrario se marcará un error.

The screenshot shows the 'Consultar ajuste' interface. At the top, there's a header with the 'Tallercom' logo and a power icon. Below the header, there's a search bar labeled 'Placas:' with the text 'PL-123' entered. To the right of the search bar is a blue button labeled 'Consultar'. Below the search bar is a table with the following data:

ID	Fecha	Tipo de ajuste	Descripción	Pendientes	Auto	Trabajador	Costo
80	2021-06-20	Clutch y fren...	Se arreglaro...	Ugente: arre...	PL-123	mary@gmail...	1000.0

At the bottom center of the screen is a large green circular icon with a white checkmark. In the bottom right corner, there is a small house icon representing a home button.

Figura 50. Pantalla para consultar ajuste

En la pantalla Nota se realizan las notas por ajustes realizados y refacciones utilizadas (Figura 51). En la parte derecha de la pantalla tenemos los campos para realizar la nota con los ajuste, este contiene los campos de texto Cliente Descripción y costo y la lista de opciones desplegable que permite seleccionar el tipo de ajuste que se realizará y finalmente se encuentra el botón “Agregar Ajuste” y agrega los datos registrados previamente a la nota.

En la parte izquierda de la pantalla se encuentran los campos para poder añadir una refracción a la nota, estos campos son: número de piezas, nombre y el costo. También cuenta con el botón Agregar refracción en el que después de completar los campos se añaden los datos registrados a la nota .

En la parte inferior de la derecha se encuentra el botón “Generar PDF” y este lo que hace es generarnos la nota en formato PDF con los ajustes y refracciones registradas previamente. Aun lado de este botón se encuentra otro botón denominado “Enviar PDF” el cual nos permite enviar el PDF, de la nota que se generó con anterioridad, vía correo electrónico registrado en la base de datos del cliente al que se le hizo la nota(Figura 52).

Figura 51. Pantalla paar realizar nota

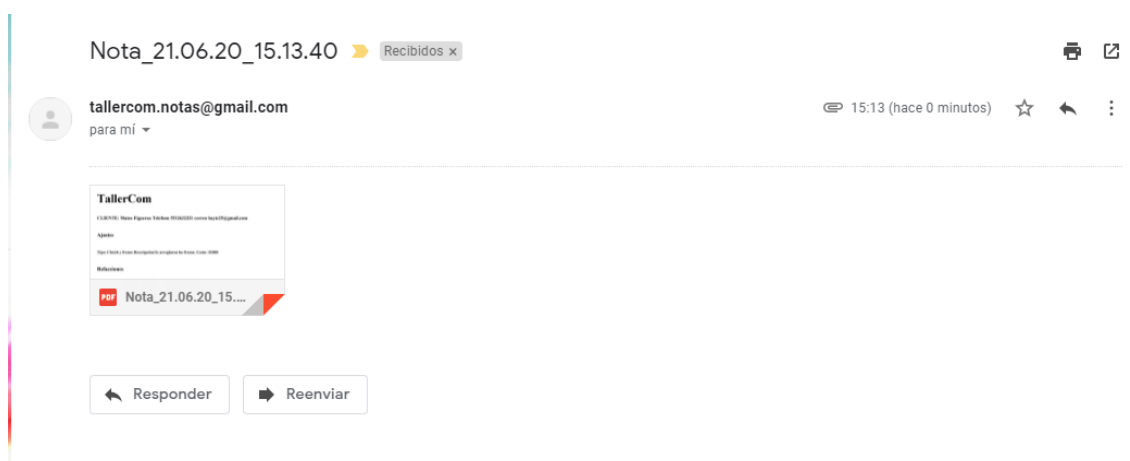


Figura 52. Correo entregado de la Nota formato pdf

En la pantalla Agendar cita como su nombre lo indica, se agendan las citas que el cliente solicite o sean necesarias para verificación del automóvil. EN la parte derecha de la pantalla tenemos el campo de texto cliente en el que se pone el nombre del cliente al que se le agendará la cita, después tenemos un calendario que se despliega para poder seleccionar la fecha de la cita, los numero que salen en color rojo en el calendario marca dias que ya cono cuentan con horarios para citas, después se tiene una lista desplegable que muestra las opciones disponibles para escoger la hora de la cita. Después de haber proporcionado los datos anteriores se agenda la cita presionando el botón “Agendar cita” en el que si los campos se llenaron de manera correcta nos agenda la cita (Figura 53) y si no son correctos se marca un error. En esta pantalla también es posible enviar la cita vía correo electrónico al cliente (Figura 54), solo se tiene que presionar el botón “Enviar cita ” una vez agendada la cita.

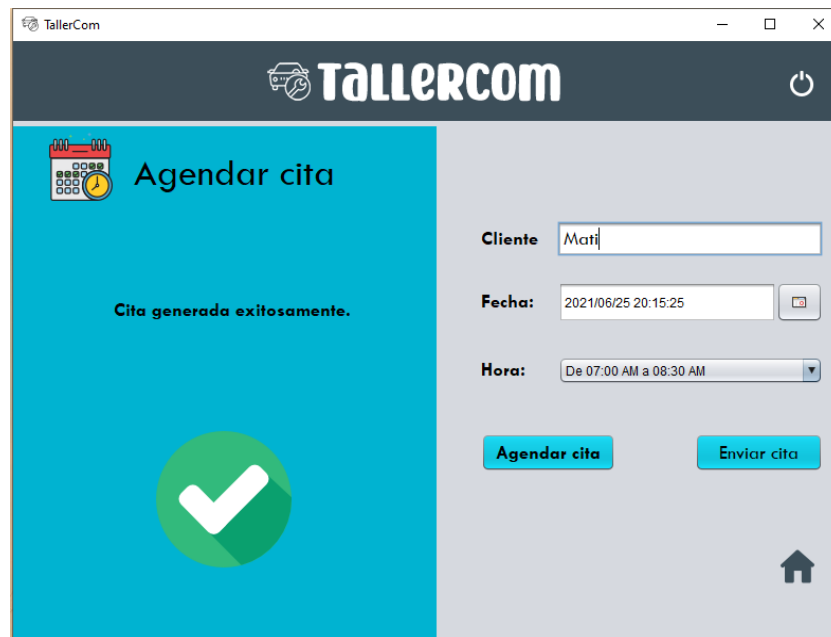


Figura 53. Pantalla de Agendar cita

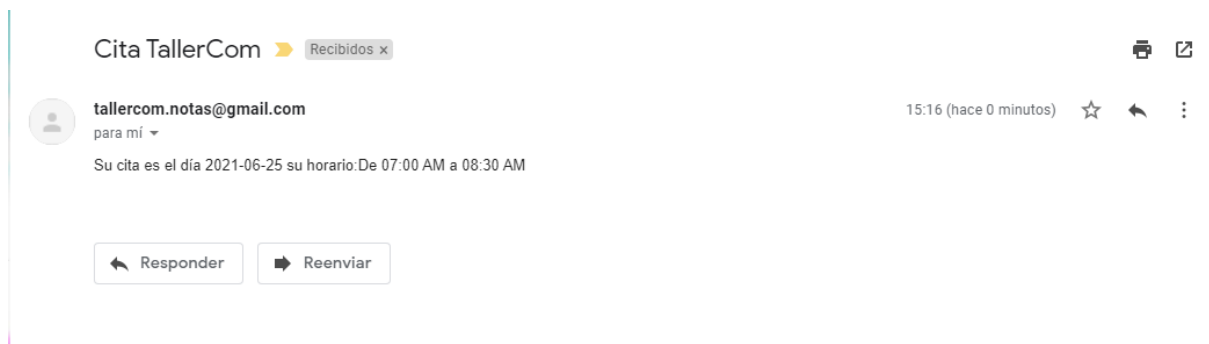


Figura 54. Correo entregado con la cita

CAPÍTULO 5. CODIFICACIÓN Y PRUEBAS

Para la conexión a la base de datos es necesario crear una variable tipo Connection e instanciarla con DriverManager.getConnection("Ubicación de la base de datos","Usuario","Contraseña"). Cabe resaltar que es de primordial importancia colocar try-catch para los posibles errores, en la conexión.

```
package Vistas;

/**
 *
 * @author lilia
 */
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/**
 * @author itz_1
 */
public class DBConexion {
    static {
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            //JOptionPane.showMessageDialog(null, "Error en el driver" + e);
            System.out.println(e.toString());
        }
    }

    public Connection getConnetion(){
        Connection con = null;
        try {
            con = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/ti
                "root","");
            //JOptionPane.showMessageDialog(null, "CONEXION EXITOSA");
        } catch (SQLException e) {
            //JOptionPane.showMessageDialog(null, "Error en conexion" + e);
            System.out.println(e.toString());
        }
        return con;
    }
}
```

Figura 55. Código de la conexión con la base de datos

Para la inserción, eliminación o actualización en la base de datos es necesaria la creación de dos variables, una de tipo Statement; la cual sirve para procesar una sentencia SQL y otra tipo ResultSet que con ayuda de la variable de tipo Statement ejecuta la sentencia SQL. Para recorrer la tabla seleccionada en la secuencia hasta que ya no haya más datos se debe utilizar el ciclo anidado while, que se seguirá repitiendo hasta que la variable ResultSet.next() sea falsa.

```

Class.forName("com.mysql.jdbc.Driver");
try {
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost/TallerCom","root","");
    Statement stm=con.createStatement();
    ResultSet rs=stm.executeQuery("select * from trabajador");
    while(rs.next()){
        correo=TxtCorreo.getText();
        correodb=rs.getString("correo");
        if(correodb.equals(correo)){
            LblError.setText("Ese correo ya existe.");
            break;
        }
    }
    if(CboXRango.getSelectedItem().equals("Administrador"))
        rango="admin";
    else
        rango="trab";
    if(correodb.equals(correo)==false){
        x = stm.executeUpdate("insert into trabajador (rango,nombre,telefono,correo,contrasena,estado) values ('"+rango+"'");
    }
    if(x==1){
        x=0;
        LblError.setText("Se agregó al trabajador exitosamente.");
        LblError2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Img/Correcto.png")));
    }else{
        LblError.setText("Verifique que los datos ingresados");
        LblError1.setText(" sean correctos.");
        LblError2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Img/Incorrecto.png")));
    }
}
System.out.println("x="+x);
TxtNombre.setText("");
TxtTel.setText("");
TxtCorreo.setText("");
TxtContrasena.setText("");

```

Figura 56. Código de agregar trabajador

Generar los PDFs, fue posible gracias a las librerías:

- org.apache.pdfbox.pdmodel.PDDocument
- org.apache.pdfbox.pdmodel.common.PDRectangle
- import org.apache.pdfbox.pdmodel.PDPage
- import org.apache.pdfbox.pdmodel.PDPageContentStream
- import org.apache.pdfbox.pdmodel.font.PDType1Font

Dichas librerías permiten crear un documento en formato PDF, crear una hoja dentro del formato pdf y agregar contenido al mismo.


```

PDDocument documento=new PDDocument();
try{
    PDPage pagina=new PDPage(PDRectangle.A4);
    documento.addPage(pagina);
    PDPageContentStream contenido=new PDPageContentStream(documento,pagina);
    contenido.beginText();
    contenido.setFont(PDType1Font.TIMES_BOLD,36);
    contenido.newLineAtOffset(20,pagina.getMediaBox().getHeight()-(52));
    contenido.showText("TallerCom");
    contenido.endText();
    contenido.beginText();
    contenido.setFont(PDType1Font.TIMES_BOLD,14);
    contenido.newLineAtOffset(20,pagina.getMediaBox().getHeight()-(52)*2);
    contenido.showText("CLIENTE: "+nombre+" Telefono "+telefono+" correo "+correodb);
    contenido.endText();
}

```

Figura 57. Código para la creación de PDFs

Finalmente, para enviar las notificaciones mediante correo electrónico se creó una conexión con los servidores de gmail, además de una variable `MimeMessage.addRecipient(Message.RecipientType.TO, new InternetAddress (destinatario))` y esa misma variable para agregar el asunto `.setSubject(asunto)`. Para enviar el archivo se creó una variable `MultiPart` donde se agregó el PDF, además de su ubicación y finalmente un variable tipo `Transport` para indicarle el correo desde donde saldrá el "mail" y su contraseña, y el correo destinatario. Todo este código está dentro de un try-catch.

```

Properties propiedad = new Properties();
propiedad.setProperty("mail.smtp.host", "smtp.gmail.com");
propiedad.setProperty("mail.smtp.starttls.enable", "true");
propiedad.setProperty("mail.smtp.port", "25");
propiedad.setProperty("mail.smtp.auth", "true");

Session session= Session.getDefaultInstance(propiedad);
String correoEnvia="tallercom.notas@gmail.com",contrasena="TallerCom20:21",destinatario=correodb,asunto ="!

MimeMessage mail=new MimeMessage(session);
BodyPart adjunto = new MimeBodyPart();
MimeMultipart multiParte = new MimeMultipart();

try {
    mail.setFrom(new InternetAddress (correoEnvia));
    mail.addRecipient(Message.RecipientType.TO, new InternetAddress (destinatario));
    mail.setSubject(asunto);
    adjunto.setDataHandler(new DataHandler(new FileDataSource("C:/PDFnotas/"+archivo+"")));
    adjunto.setFileName(""+archivo+"");
    multiParte.addBodyPart(adjunto);
    mail.setContent(multiParte);
    Transport transporte= session.getTransport("smtp");
    transporte.connect(correoEnvia,contrasena);
    transporte.sendMessage(mail, mail.getRecipients(Message.RecipientType.TO));
    transporte.close();
}

```

Figura 57. Código para enviar correo electrónico mediante gmail

CAPÍTULO 6. PRUEBAS

Plan de pruebas Plan001_TallerCom

Objetivo

Establecer y detallar las condiciones de las pruebas que se llevarán a cabo para validar un sistema con todas las funciones básicas definidas en los requerimientos de usuario.

Estrategia

Durante el desarrollo del proyecto se llevarán a cabo diferentes tipos de pruebas las cuales explicaremos a continuación:

//Se llevarán a cabo distintas pruebas funcionales, para garantizar que las //características y funcionalidades se comporten según lo ya diseñado.

El desarrollo de las pruebas funcionales, se llevarán a cabo en dos fases:

- Pruebas unitarias: en esta fase se procederán a hacer pruebas en cada módulo para tener un control de estas y de los errores en una pequeña escala, esto se realizará en cada módulo para verificar que estos cumplan con su debida función sin ningún error.
- Pruebas de integración: en esta fase se llevará a cabo la integración de los distintos módulos y las pruebas de estos, de esta forma se validará la integración de diferentes módulos juntos e identificar errores y problemas relacionados con ellos.

Se llevarán a cabo las pruebas mínimas no funcionales relacionadas con el performance testing, las cuales se encargan de determinar la velocidad, estabilidad y escalabilidad de la aplicación, estas pruebas serán llevadas a cabo de la integración de todos los módulos, para comprobar en un ambiente real el flujo de datos y el almacenamiento de estos.

Se llevarán a cabo pruebas de regresión cuando se realice cualquier cambio en el sistema para poder observar que todo sigue funcionando después de ese cambio.

Alcance

El sistema deberá ser capaz de cumplir con los requisitos establecidos por el usuario, teniendo un rendimiento adecuado en el ambiente que será instalado. Además de cubrir con las características de seguridad especificadas por el cliente.

Propósito

Encontrar errores y defectos que puedan existir en el uso del sistema a fin de corregirlos, además de cotejar que los validadores de datos funcionen y limiten el ingreso de información, para que no se puedan ingresar datos que no están permitidos.

Entorno de configuración de pruebas

- NetBeans
- Servidor MySQL database versión 8.1.
- Equipos de los desarrolladores
 - Intel Core 7-4790K CPU a 4GHz, 8GB de RAM, Windows 10.
 - Ryzen 5 2600 CPU a 3.4GHz, 16 GB de RAM, Windows 10.
 - Intel Core i3-2330M CPU a 2.2GHz, 4GB de RAM, Windows 10.

- Intel Core i3-3220 CPU a 3.30GHz, 6 GB de RAM, Windows 10.

Características a ser probadas

Característica	Descripción	Módulo
Verificar el inicio de sesión de los trabajadores.	Realizar una consulta en la base de datos con los datos ingresados, para poder asegurar el inicio de una sesión.	1
Examinar la realización del registro de clientes y sus vehículos, además de los trabajadores del taller	El usuario que utilice la aplicación podrá dar de alta a clientes nuevos, vehículos y trabajadores, realizando el llenado de un formulario con los datos requeridos para cada objeto y esto debe verse reflejado en la base de datos.	2
Comprobar las consultas de los trabajadores, los clientes y sus vehículos.	Se podrá comprobar que los datos obtenidos de las distintas consultas del sistema sean correctos.	3
Probar el registro de las reparaciones hechas a cada vehículo que un cliente tenga.	Verificar que los registros sean insertados en la base de una forma correcta.	4
Cotejar que las notas lleguen al cliente que corresponde.	Hacer la comprobación del envío de correos a los clientes para no tener un destinatario erróneo.	5

Tabla 1: Características a ser probadas.

Criterios de aprobación y fallo

Errores graves: información crítica presentada erróneamente, como datos relacionados con los autos o nombres del cliente, información mal registrada en la base de datos, caídas del programa, incumplimiento de objetivos en funciones importantes.

Errores medios (comunes): errores en presentación de datos, incumplimiento de objetivos en funciones secundarias, como la falta de impresión de un dato.

Errores leves: errores en la presentación de datos secundarios, comportamientos correctos pero diferentes en situaciones similares, dificultades de operación, etc.

Criterio de aprobación: el proyecto será aprobado con un 100% de las pruebas ejecutadas y se tomarán en cuenta todas aquellas pruebas que tengan el 90% de aceptación. Esto quiere decir que el 90% de las pruebas deben ser exitosas y sin errores. Para el 10% restante pueden existir errores medios o bajos, pero no graves.

Criterios de suspensión y reanudación

Cuando se lleva a cabo la ejecución de una prueba y esta nos arroja un resultado no deseado, se suspenderán las pruebas y se procederá a solucionar el error en el sistema, una vez encontrado y solucionado se procederá a reanudar estas pruebas para poder verificar el buen funcionamiento del sistema.

Test-ware (Herramientas de prueba)

- JUnit. JUnit es una herramienta de prueba de unidad de código abierto para el lenguaje de programación Java.

CAPÍTULO 7. CONCLUSIONES

Se logró elaborar el prototipo funcional de la aplicación propuesta “Tallercom” con su debido análisis y diseño que esto conlleva, esta aplicación cumple con los objetivos planteados al inicio del proyecto, con lo que es posible mencionar que la aplicación permite llevar el control de los clientes y sus vehículos, además de llevar un registro de las reparaciones realizadas en el control, de esta forma permite una fácil gestión del taller, esto se logró gracias a todo el diseño previo, el cual fue desarrollado a lo largo de la materia “Análisis y diseño orientados a objetos”, cubriendo las normativas dadas para el desarrollo de los distintos diagramas que se realizaron en el desarrollo. Podemos mencionar que el proyecto es totalmente viable para su implementación en talleres locales.

REFERENCIAS

- ❖ L, A. R. (2003, 22 septiembre). Metodologías para generación de Sistemas Orientados a Objetos. Cursos Clavijero. Recuperado de https://cursos.clavijero.edu.mx/cursos/178_pds/modulo1/recursos/T32.pdf
- ❖ Gómez, J. (2011, 26 mayo). Manuales de usuario y técnico. Slideshare. Recuperado de <https://es.slideshare.net/Dolphinus/manuales-de-usuario-y-tecnico>
- ❖ Ittgweb. (2016, mayo). Descomposición modular. Ittgweb. Recuperado de [https://ittgweb.wordpress.com/2016/05/29/descomposicion-modular/#:~:text=Descomposici%C3%B3n%20Modular%20o%20Modularizaci%C3%B3n%20es,cohesi%C3%B3n%20\(con%20significado%20propio\).&text=El%20dise%C3%B1o%20modular%20propone%20dividir,diferenciadas%20y%20definir%20sus%20interfaces](https://ittgweb.wordpress.com/2016/05/29/descomposicion-modular/#:~:text=Descomposici%C3%B3n%20Modular%20o%20Modularizaci%C3%B3n%20es,cohesi%C3%B3n%20(con%20significado%20propio).&text=El%20dise%C3%B1o%20modular%20propone%20dividir,diferenciadas%20y%20definir%20sus%20interfaces)
- ❖ Chavez Gona, V. M., & Olivares Rojas, J. C. (2002, 12 marzo). METODOLOGÍA OMT. itmorelia. Recuperado de [http://dsc.itmorelia.edu.mx/~jcolivares/documents/omt#:~:text=La%20metodolog%C3%ADa%20OMT%20\(Object%20Modeling,que%20existen%20en%20la%20actualida](http://dsc.itmorelia.edu.mx/~jcolivares/documents/omt#:~:text=La%20metodolog%C3%ADa%20OMT%20(Object%20Modeling,que%20existen%20en%20la%20actualida)
- ❖ 401, R. (2018, 7 noviembre). Apps para gestionar el trabajo y la organización del taller. Recuperado de <https://blog.reparacion-vehiculos.es/apps-gestionar-trabajo-taller>

- ❖ EAE, (2019, 25 diciembre). Cronograma de actividades: qué es y cómo hacerlo. Recuperado de <https://retos-operaciones-logistica.eae.es/cronograma-de-actividades-que-es-y-como-hacerlo/>

ANEXOS

Cronograma de actividades

Actividad	Abril	Mayo	Junio
Recolectar datos	Semana 2-3		
Modelo de objetos	Semana 3		
Modelo dinamico	Semana 4		
Modelo Funcional	Semana 4		
Diseño del sistema		Semana 1	
Diseño de los objetos		Semana 2	
Implementacion		Semana 3-4	

Figura Anexos 1. Cronograma de actividades