

MULTIMOTORS (TFG)

MEMORIA Y DOCUMENTACIÓN: MULTIMOTORS

Desarrollado por Fabián Rodríguez y Sergio Hernández



Tutor: José Luis Meiriño Seoane

ÍNDICE

DEFINICIÓN DEL SISTEMA.....	4
Análisis de la utilidad/necesidad del sistema. Proyección a nivel empresarial.....	4
Identificación del alcance del sistema. (Objetivos, requisitos iniciales, algún diagrama inicial que represente el sistema...)	4
Diagrama de casos de uso de funcionalidades principales	5
Identificación del entorno tecnológico.....	5
Planificación de las fases del desarrollo del proyecto.	7
METODOLOGÍA UTILIZADA	7
Identificación de funcionalidades.....	7
Definición de versiones incrementales	8
Mejoras.....	8
Pruebas integrales.	8
Despliegue final.....	8
ANÁLISIS DEL SISTEMA.	9
Especificación de requisitos.	9
Especificación de estándares y normas de aplicación en el proyecto.	10
Diseño de interfaces de usuario.....	10
Arquitectura del sistema.	10
Seguridad de la información.....	11
Almacenamiento de datos.	11
DISEÑO DEL SISTEMA.	12
Diseño de interfaces de usuario.....	12
Diseño del modelo de datos.....	18
ASPECTOS RELEVANTES DE LA IMPLEMENTACIÓN.	19
Diseño y estructura de la base de datos.	19
Integración de APIs y servicios externos.	20
Amazon Web Services (AWS).....	20
API de Chat GPT.....	21
API de Stripe.	21
Otras APIs.	21
Funcionalidades destacadas y distintas.	22

PRUEBAS DE ACCESIBILIDAD Y USABILIDAD.....	38
Pruebas de Usabilidad.	38
Pruebas de Accesibilidad.	38
Otras pruebas y mejoras implementadas.	39
DESPLIEGUE DEL PROYECTO.....	40
PROBLEMAS ENCONTRADOS.	42
CONCLUSIONES.	44
Evaluación crítica de los resultados obtenidos.	44
Propuestas de mejoras.	44
Desarrollos futuros.	45
BIBLIOGRAFÍA.....	46

DEFINICIÓN DEL SISTEMA.

Nuestra aplicación es una aplicación basada en una aplicación de concesionario. Diseñada para que los usuarios tengan una experiencia realista, funcional y agradable; y los administradores tengan de una forma sencilla y eficaz la realización de cualquiera de sus actividades.

Análisis de la utilidad/necesidad del sistema. Proyección a nivel empresarial.

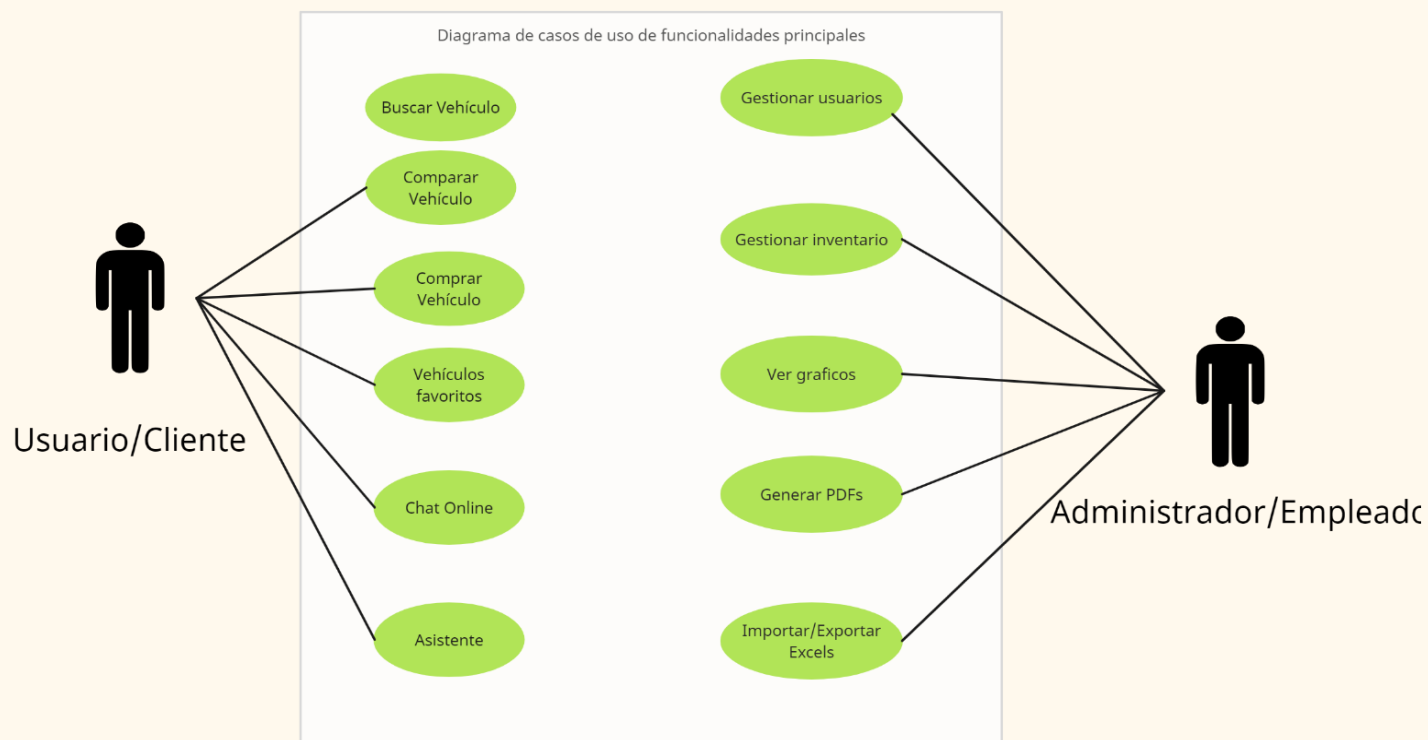
La aplicación de MultiMotors se ha desarrollado con el objetivo de ofrecer a los usuarios una plataforma de compra/renta de vehículos completa. Para ello se han desarrollado diversas funcionalidades como búsquedas avanzadas por filtros, comparativa de vehículos para facilitar la elección del mismo, información detallada sobre cada vehículo y facilidad a la hora de comprar o rentar uno.

Proporciona una interfaz totalmente funcional y sencilla. Además, contamos con varias opciones de pago reales, generación de contratos y facturas y correos electrónicos de verificación.

Identificación del alcance del sistema. (Objetivos, requisitos iniciales, algún diagrama inicial que represente el sistema...)

El alcance de nuestra aplicación sería abarcar todo lo relacionado con una buena gestión de un concesionario de vehículos tanto por parte del usuario como por parte de las cuentas administradoras. Los usuarios tienen acceso a una tabla con una gran cantidad de vehículos con todo tipo de información sobre ellos. Por otro lado, los administradores tienen la capacidad de gestionar el inventario, los perfiles de los usuarios, realizar todos los cambios que ellos quieran tanto al vehículo como a un usuario, acceder a gráficos con datos importantes o la generación de PDFs con estos mismos.

Diagrama de casos de uso de funcionalidades principales



Identificación del entorno tecnológico

En este apartado explicaremos de forma breve todo lo que hemos utilizado para el desarrollo de esta aplicación. Mencionando también sus principales ventajas y desventajas a nuestro parecer.

Lenguajes de programación utilizados: Java y PHP

Java se ha utilizado para la realización de todas las funcionalidades de la interfaz de la aplicación. En cuanto a sus ventajas mencionaremos que es el lenguaje en el que mejor nos manejamos hasta el momento ya que es el que más hemos practicado y en cuanto a su principal desventaja sería que no puedes ejecutar la aplicación si no tienes Java instalado en tu ordenador. Sin duda, esta es una de las principales desventajas de nuestra aplicación.

PHP se ha utilizado para la realización de la API Rest. En cuanto a su principal ventaja es el orden con el que se trabaja ya que te permite separar el cliente por un lado y el servidor por otro. En cuanto a una desventaja es que tienes que cambiar tu forma de pensar a la hora de relacionar la base de datos con el código ya que puede llegar a ser más complejo.

JSON se utiliza para la comunicación entre el cliente y el servidor.

IDE utilizados: NetBeans y Visual Studio Code

Netbeans fue nuestro IDE escogido para realizar la aplicación y todo lo relacionado a ella menos la API. Como ventaja es que te pone muchas facilidades a la hora de por ejemplo hacer un Commit and Push para github o también que tiene una configuración muy sencilla. Como principal desventaja es lo limitado que estas sobre todo en el tema del diseño, es mucho más complicado realizar un diseño bonito en NetBeans con Java Swing que en Visual Studio con WPF por ejemplo.

Visual Studio Code ha sido el IDE utilizado para realizar los códigos PHP, sin duda uno de los mejores entornos de desarrollo que casi nunca da problemas. No hemos tenido ningún tipo de problema ni hemos encontrado alguna desventaja que nos haya perjudicado a gran nivel.

Base de datos utilizada y Almacenamiento en la nube: MySQL y AWS

MySQL es uno de los sistemas de gestor de bases de datos más usados y conocidos del mundo. Nosotros lo elegimos porque es el que mejor conocemos y como la base de datos era una pieza fundamental de la aplicación preferimos ir a lo seguro.

En cuanto al Almacenamiento en la nube decidimos utilizar el Web Service de Amazon (AWS), era de los pocos que nos proporcionaba pruebas gratuitas y una gran variedad de opciones como por ejemplo su servicio s3 en el que puedes almacenar ahí tus archivos u otro servicio de gestión de APIs donde hemos podido subir nuestra API Rest. A pesar de lo complicado que fue la configuración de algunas opciones como la de subir la API o controlar los archivos, estamos muy contentos con su función.

En cuanto a las librerías, hemos usado una gran cantidad de librerías para poder conseguir muchas de las funcionalidades presentes en la aplicación, aquí mencionaremos las más destacadas.

Librerías utilizadas que destacar: Stripe, POI, lombok, jMail, iText7, jfreechart.

Stripe: API y servicio externo para la gestión de los pagos.

POI: Todo lo relacionado con los Excels.

Lombok: Automatizador de clases en Java.

jMail: Servicio para enviar correos electrónicos.

iText7: Para la creación de los PDFs.

jfreechart: Para la creación de los gráficos.

Para gestionar las librerías hemos utilizado Maven.

Planificación de las fases del desarrollo del proyecto.

METODOLOGÍA UTILIZADA

En el desarrollo de nuestra aplicación, se ha seguido una **metodología incremental**. Esta metodología permite una construcción del producto final a medida que se avanza en el proyecto,

es decir, en cada etapa se irá desarrollando y agregando una nueva funcionalidad importante, lo que permitirá ver los resultados de una forma más rápida que otro tipo de metodologías. A continuación, se detallan los pasos clave de la metodología incremental utilizada:

Identificación de funcionalidades

En esta etapa inicial, se identifican las funcionalidades claves e importantes de nuestra aplicación, haciéndonos una pequeña idea de las necesidades de los usuarios y administradores. Se priorizaron las funcionalidades que brindarán el mayor valor y utilidad en las primeras etapas del desarrollo.

Definición de versiones incrementales

Se dividió el proyecto en pequeñas “versiones”, donde cada versión se centrará en un grupo específico de funcionalidades mínimamente relacionadas. Por cada versión se establece un objetivo de alcance y una fecha de finalización para poder gestionar bien las mejoras y el tiempo restante para la entrega final.

Mejoras.

Tras terminar nuestra versión del proyecto, nos reunimos para juntarlas y realizar pequeñas mejoras y se corrigen pequeños errores entre los dos que hagan que esas nuevas funcionalidades funcionen lo mejor posible.

Pruebas integrales.

A medida que se completaron las versiones anteriores, se realizaron pruebas de todo tipo que abarque todas las funcionalidades implementadas hasta ese momento. Esto asegura la integridad y la compatibilidad entre las diferentes partes de la aplicación.

Despliegue final.

Una vez completadas todas las versiones anteriores se alcanzó un gran nivel de funcionalidad y calidad en la aplicación. Se varían realizando nuevas funcionalidades menores con relación al tiempo que nos queda. Estas funcionalidades seguirán otro tipo de metodología ya que las habría que hacer en un periodo muy corto de tiempo.

ANÁLISIS DEL SISTEMA.

Especificación de requisitos.

Este análisis se enfocará en identificar y resaltar los requisitos funcionales y no funcionales de la aplicación de MultiMotors. Los requisitos funcionales describirán las acciones y funcionalidades más destacadas e importantes con las que cuenta nuestra aplicación. Por parte del usuario podemos contar con la búsqueda de vehículos, la comparación de características básicas y detalladas, un apartado de tus vehículos comprados, una lista de favoritos, un chat online para conversar con los otros usuarios que se encuentran conectados y la generación de contratos y facturas al realizar una compra. También, cuenta con pequeñas funcionalidades, pero no por ello menos importantes como una función de Recordar la cuenta, una función de mantener la sesión iniciada o una función de has olvidado tu contraseña, todas ellas hacen de nuestra aplicación una aplicación sencilla e intuitiva para el usuario. Por otro lado, tenemos a los administradores, los cuales cuentan con todo tipo de gestión de usuarios e inventario (Agregar, editar, eliminar, importar y exportar Excels, descargar Excels preformateados) y una gran variedad de gráficos relacionados con los tipos de vehículos, las categorías, las marcas y las ventas, gráficos los cuales más tarde pueden ser descargados en un PDF generado por ti. Por otro lado, contamos con otras

funcionalidades importantes que hacen de nuestra aplicación a parte de sencilla e intuitiva, segura. Todas las contraseñas de cada usuario se guardan en la base de datos tras un método de encriptación a través de PHP, las contraseñas se llegan a encriptar hasta más de 5 veces antes de guardarse en la base de datos, todo esto gracias al método `password_hash`. También hay que recalcar que datos como el método de pago, número de tarjeta o ISBN no son nunca guardados en la base de datos. Simplemente se traspasan al servicio de pago.

Especificación de estándares y normas de aplicación en el proyecto.

En el desarrollo de MultiMotors, se han ido estableciendo estándares y normas con el objetivo de garantizar una aplicación con calidad y consistencia a la hora de trabajar con ella. Contamos con varios estándares y normas principales:

Diseño de interfaces de usuario.

Se ha seguido el diseño de interfaces de usuario tipo Java Swing para crear una experiencia fácil de usar y no muy saturada. A pesar de que creemos que el diseño es nuestra principal desventaja en el desarrollo de la aplicación, se ha conseguido lo que teníamos en mente y a pesar de las dificultades que nos hemos encontrado al realizarlo con Java Swing. Se ha prestado atención a aspectos como la disposición de componentes para que la pantalla no llegue a estar demasiado saturada, la consistencia en la apariencia visual a la hora de intentar mantener un mismo estilo o tema en cada pantalla, la navegación intuitiva ya que en la mayoría de pantallas tienes un botón de volver a pantalla principal o bien de cerrar la pantalla arriba a la derecha y la accesibilidad a otro tipo de pantallas habiendo también otro tipo de botones que dejan muy claro a que pantalla se dirigen.

Arquitectura del sistema.

Se ha aplicado y seguido el modelo de arquitectura de Modelo-Vista-Controlador (MVC). En cuanto a la capa de Modelo tenemos varias clases en las cuales tenemos un modelo predefinido

de todo lo que se encuentra o se genera en la aplicación, ya sea la dirección, un usuario, un vehículo, etc...

La vista cuenta con cada clase tipo Form, donde se realiza todo el diseño y las funcionalidades relaciones con el diseño, como puede ser la acción que se realiza al presionar un botón o al estar encima de un área.

Contamos con varios controladores donde gestionamos y controlamos todos los métodos y llamadas a nuestra API Rest generando listas de JSON que se envían a nuestro código PHP para llamar a la base de datos.

También tenemos otros tipos de clases que hemos llamado útiles que tienen métodos concretos que se usan a lo largo de la aplicación como el de enviar un correo electrónico, generar un pdf o realizar un pago.

Seguridad de la información.

Se han implementado medidas de seguridad para proteger la integridad, la confidencialidad y la disponibilidad de los datos del sistema. Se han seguido pautas en términos de autenticación (a la hora de enviar un correo a la hora de registrarte o loguearte), autorización (una confirmación de correo al realizar un pago), encriptación de datos (como la contraseña en la base de datos), protección contra ataques (para ello hemos subido nuestra base de datos y API Rest a un web service seguro como es AWS) y seguimiento de registros para la detección de errores comunes (a través de logs generados en nuestra API).

Almacenamiento de datos.

Se han utilizado estándares de diseño de bases de datos para garantizar una estructura lógica y eficiente en el almacenamiento de datos, llegando a relacionar hasta 3 tablas o colecciones seguidas.

DISEÑO DEL SISTEMA.

Como hemos mencionado anteriormente, nuestro diseño podría llegar a ser nuestro punto débil y nuestra principal desventaja durante el desarrollo de MultiMotors. Aun así, pensamos que hemos conseguido un diseño bonito y ordenado.

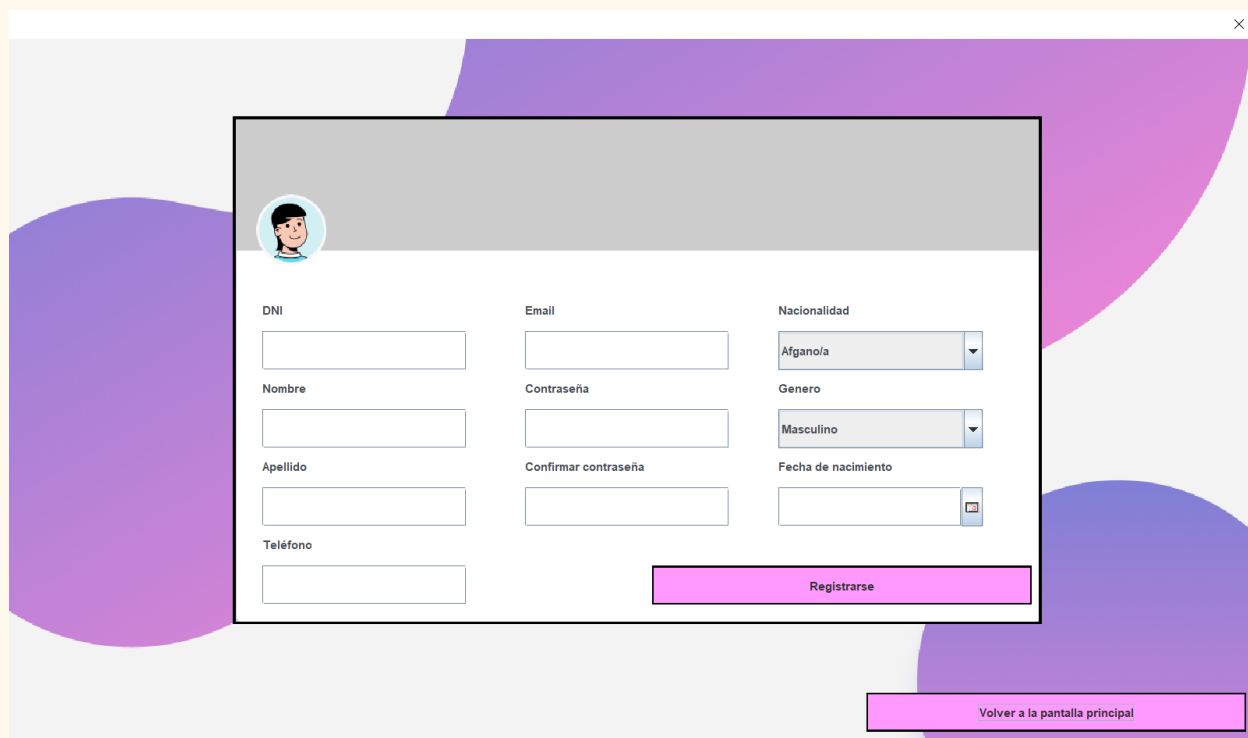
Diseño de interfaces de usuario.

Colores: Los colores que hemos elegido para la aplicación están de esa forma elegidos porque no queríamos colores apagados, queríamos que resaltasen un poco y que fuesen fáciles de recordar. ¿Cuántos concesionarios has visto tú con estos colores?, seguro que muy pocos. Creemos que el color rosa que usamos mayoritariamente hace también un buen contraste con el color negro y color gris, por lo tanto, es fácil distinguir entre el fondo y el texto.

Fuentes: Las fuentes que hemos elegido han sido mayoritariamente la de Segoe UI 12 Plain, esta fuente es la predeterminada que te ofrece Java Swing en NetBeans a la hora de escribir algo de texto en un diseño. Pensamos que es una fuente que da igual que sea predeterminada, es muy legible, las letras se distinguen perfectamente entre mayúsculas y minúsculas y no es un tipo de letra que haga daño a los ojos, visualmente es limpia y bonita. En cuanto a los colores de los textos la gran mayoría son negros o grises, como hemos mencionado anteriormente el rosa es un color que contrasta perfectamente con colores oscuros como el negro o el gris, por lo tanto, se distingue perfectamente entre el fondo y el texto.

Bocetos: Ahora vamos a mostrar un par de capturas de nuestras pantallas más destacadas y vamos a explicar porque están diseñadas de esa forma y porque creemos que es la mejor estructura.

Comenzamos con los formularios de registro e inicio de sesión.



A registration form interface displayed within a window. The form is titled with a user icon and contains several input fields and dropdown menus. The fields are arranged in a grid-like structure. At the bottom right of the form is a pink button labeled 'Registrarse'. Below the form, outside the window, is another pink button labeled 'Volver a la pantalla principal'.

Formulario de Registro		
DNI	Email	Nacionalidad
<input type="text"/>	<input type="text"/>	<input type="text" value="Afgano/a"/>
Nombre	Contraseña	Genero
<input type="text"/>	<input type="text"/>	<input type="text" value="Masculino"/>
Apellido	Confirmar contraseña	Fecha de nacimiento
<input type="text"/>	<input type="text"/>	<input type="text"/>
Teléfono		
<input type="text"/>		

[Registrarse](#)

[Volver a la pantalla principal](#)

Correo electrónico

sergiopozanco@gmail.com

Contraseña

.....

☒ Recuerdame ☒ Mantener sesión iniciada

Iniciar sesión

Registrate [¿Olvidaste la contraseña?](#)

Volver a la pantalla principal

Estos formularios están distribuidos de tal forma que el campo de texto a rellenar sea amplio y claro y justo encima del mismo esté escrito con lo que se tiene que rellenar. Son campos de textos grandes y creemos que la distribución de ellos es lo suficientemente clara y separada como para saber que campo rellenar para cada dato. Todo esto está acompañado de algunos botones que al ser rosas no pasan desapercibidos, el botón menos importante está apartado, en una esquina, en cambio el botón más importante está en el panel del centro junto al formulario. No son formularios muy saturados. Esta misma distribución de campos de textos amplios también la tendrán otros apartados como el de mi cuenta o el de has olvidado tu contraseña.


Ahora veremos la pantalla principal, la primera que se abre al abrir la aplicación. En esta aplicación vamos a ver sobre todo dos paneles, uno en el lado derecho y otro en el izquierdo. El del lado derecho refleja una tabla con los vehículos que están disponibles en ese momento y sus

datos más importantes. El lado izquierdo refleja un gran catálogo de filtros avanzados con los que podrás buscar tu coche favorito.

Vehículo	Combustible	Kilometros	Transmision	Potencia	Precio
BMW 4 Series	Gasolina	60000	Manual	250	30000.0
Kia Rio	Gasolina	99000	Automática	400	22000.0
Chevrolet Cruze	Gasolina	70000	Automatica	160	12000.0
Hyundai Tucson	Diesel	1000	Manual	800	10000.0
Jeep Cherokee	Gasolina	80000	Manual	290	25000.0
Kawasaki Ninja	Gasolina	100	Manual	1000	10000.0
Kenton GTR	Gasolina	12	Manual	200	350000.0
Honda Civic	Diesel	260000	Automatica	160	14000.0
Kenton GTR	Gasolina	12	Manual	200	350000.0
Toyota Camry	Gasolina	50000	Automatica	200	15000.0
Ford Explorer	Gasolina	10	Automatica	300	40000.0
Volkswagen Golf	Diesel	80000	Automatica	150	10000.0


Este diseño se podría mejorar haciendo el panel de los filtros más estrecho y la tabla algo más grande. Aun así, que los filtros sean grandes y ocupen casi el 50% de la pantalla te ayudará a detectar que puedes filtrar para que no haya tanto caos de vehículos en la tabla.

Otros diseños destacados por parte del usuario serían las pantallas de los vehículos y el formulario de compra/renta de vehículo. Ambos son claros, no hay componentes en exceso y son fáciles para la comprensión del usuario. El formulario de compra seguiría el mismo estilo que todos los formularios, campos de texto grandes y separados con los demás.



Datos básicos		
Marca Kia	Categoría Sedan	Precio de Renting 22000.0 €
Modelo Rio	Matrícula 1234 ABC	Precio de compra 22000.0 €

Datos técnicos		
Capacidad del depósito 60 L	Kilómetros 99000 km	Año del modelo 2018
Tipo gasolina Gasolina	Tipo de transmisión Automática	Potencia 400 cv
Consumo medio 12 L / 100km	Peso 12000	Tracción Delantera




Protegido por garantía

Este vendedor ofrece una garantía de un año para este vehículo. Recuerda que, a la entrega del vehículo este deberá de estar tal y como figura en el contrato de compraventa

COMPARAR CON OTRO VEHICULO

COMPRAR O RENTAR



RENTAR

COMPRAR

Datos de dirección

Pais
España

Ciudad
Madrid

Población
Madrid

Código postal
28011

Nombre de la calle
Calle de la Huerta de Castañeda

Número
33

Bloque
3

Planta
3

Puerta
3

Datos de pago ☒ Tarjeta ☐ Transferencia

Número de la tarjeta

Fecha de caducidad
MM/YY

Titular de la tarjeta

Método de pago
Al contado

Emisor de la tarjeta
VISA

CVV

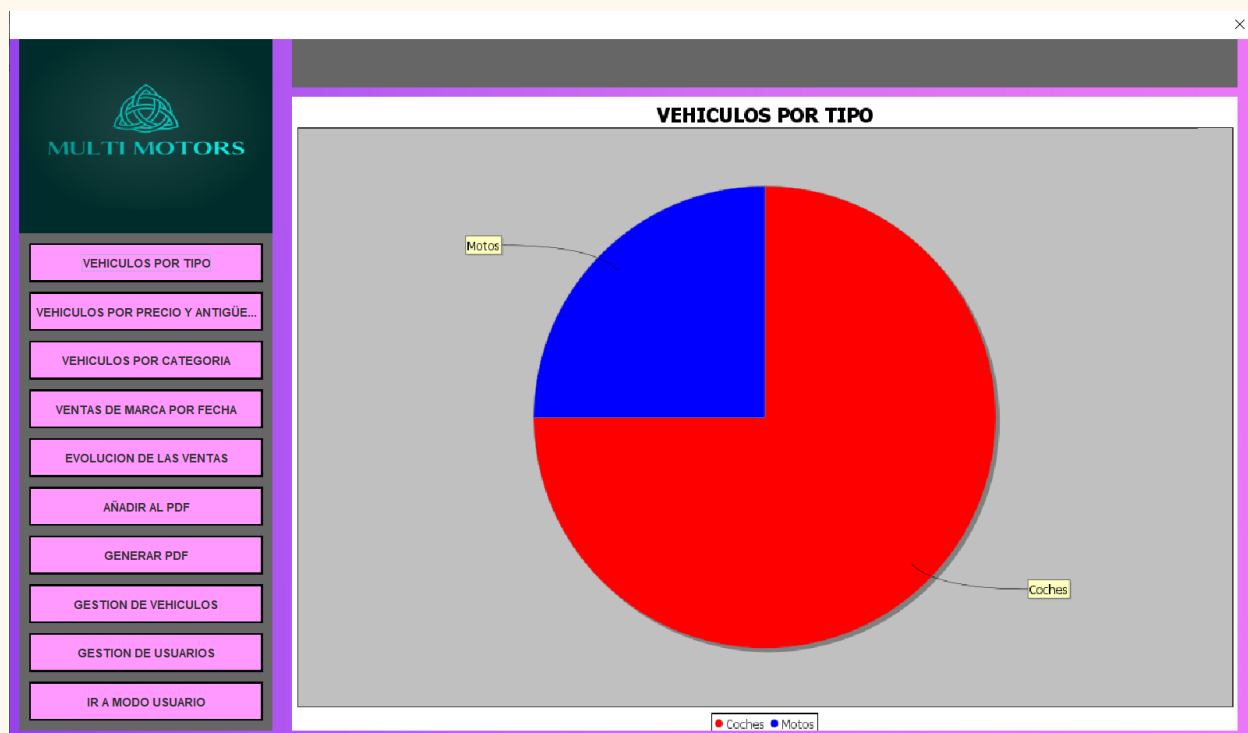
Número de cuotas
1

IMPRIMIR FACTURA

FINALIZAR COMPRA

En cuanto a los diseños del administrador no son muy complejos. El panel principal consta de una gráfica gigante que ocupa en torno al 70% de la pantalla la cual podrá ir variando, dependiendo que gráfica escojas, y un menú en el lateral izquierdo con todas las opciones de las gráficas y el acceso a los paneles de gestión de vehículos y usuarios.

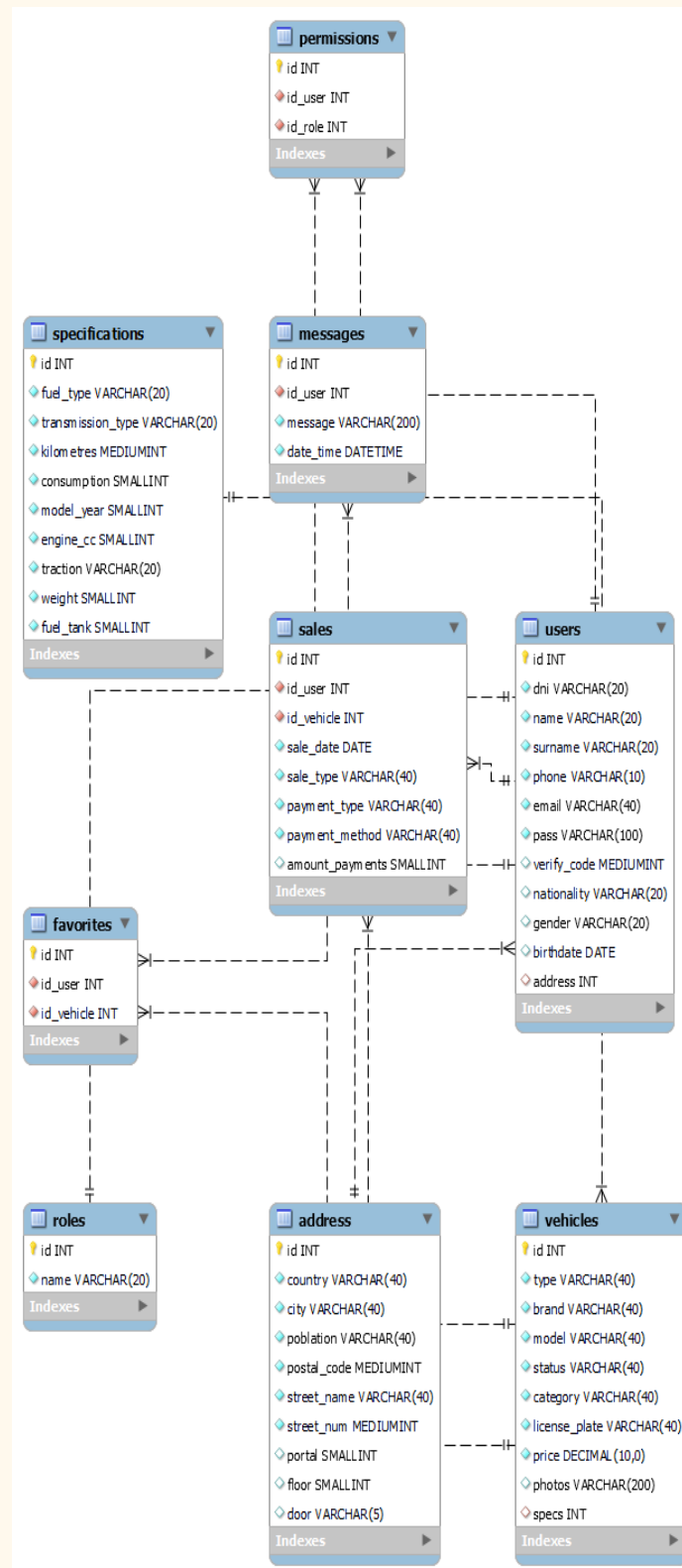
Los paneles de gestión tienen la misma estructura, a la derecha aparece una tabla con los vehículos o usuarios con todos sus datos y a la izquierda campos de texto que se irán rellenando a medida que selecciones una fila en la tabla con los datos que le correspondan. Debajo de los campos de texto aparecen 3 iconos que hacen referencia añadir, actualizar o eliminar. Debajo de estos vemos otros 3 botones que hacen referencia a funcionalidades distintas dependiendo si son usuarios o vehículos. En la parte inferior de la tabla veremos varios botones, todos ellos relacionados con las funciones que se pueden realizar mediante un Excel en nuestra aplicación.



Tipo	Marca	Modelo	Estado	Categoría	Matricula
Coche	BMW	4 Series	Disponible	Clasico	345 JKL
Coche	Kia	Rio	Disponible	Sedan	1234 ABC
Coche	Chevrolet	Cruze	Disponible	Sedan	789 MNO
Coche	Hyundai	Tucson	Disponible	Clasico	111 ABC
Coche	Jeep	Cherokee	Disponible	SUV	2234 STU
Moto	Kawasaki	Ninja	Disponible	Circuito	122 MBC
Coche	Kenton	GTR	Disponible	Sedan	1111AAAA
Coche	Honda	Civic	Disponible	Sedan	5567 XTW
Moto	Kenton	GTR	Disponible	Montaña	444 MBC
Coche	Toyota	Camry	Disponible	Sedan	123 XYZ
Coche	Ford	Explorer	Disponible	SUV	456 ABC
Coche	Volkswagen	Golf	Disponible	Sedan	789 DEF

Diseño del modelo de datos.

Tenemos una base de datos grande, aunque no muy compleja por la forma en la que hemos conseguido estructurarla. Más adelante explicamos textualmente la base de datos con todas sus relaciones para la gente que no sabe leer este tipo de diagramas, pero, a continuación, vamos a mostrar un diagrama de lo que sería nuestra base de datos con todas sus relaciones.



ASPECTOS RELEVANTES DE LA IMPLEMENTACIÓN.

En nuestra aplicación, destacan varios elementos y aspectos claves que han contribuido al desarrollo de MultiMotors para hacer de su funcionamiento un funcionamiento eficiente y efectivo. Algunos de los aspectos más relevantes son:

Diseño y estructura de la base de datos.

La base de datos de MultiMotors utiliza MySQL como tecnología de base de datos y consta de 9 tablas relacionadas entre sí para almacenar, gestionar y obtener los datos.

Anteriormente ya hemos enseñado un diagrama de la base de datos. Ahora lo vamos a explicar de manera escrita por si queda alguna duda.

Tabla "usuarios": Almacena información sobre los usuarios de la aplicación, incluyendo datos como el ID, DNI, nombre, apellidos, teléfono, correo, contraseña, código, nacionalidad, género, fecha de nacimiento y dirección. La relación con la tabla "dirección" se establece mediante el campo "dirección" en esta tabla.

Tabla "dirección": Contiene detalles sobre las direcciones de los usuarios, como país, ciudad, población, código postal, nombre de la calle, número, portal, planta y puerta. La relación con la tabla "usuarios" se establece mediante el campo "id" en esta tabla y el campo "dirección" en la tabla "usuarios".

Tabla "roles": Almacena los roles de los usuarios en la aplicación, como administrador, usuario normal, etc. Los roles se identifican mediante un ID y un nombre.

Tabla "permisos": Establece los permisos asociados a los roles de los usuarios. Se relaciona con las tablas "usuarios" y "roles" a través de los campos "id_usuario" e "id_rol" respectivamente.

Tabla "vehículos": Almacena información sobre los vehículos disponibles, incluyendo detalles como el ID, tipo, marca, modelo, estado, categoría, matrícula, precio, fotos y especificaciones. La relación con la tabla "especificaciones" se establece mediante el campo "especificaciones" en esta tabla.

Tabla "especificaciones": Contiene los detalles específicos de cada vehículo, como tipo de combustible, transmisión, kilómetros, consumo, año, potencia, tracción, peso y capacidad.

La relación con la tabla "vehículos" se establece mediante el campo "id" en esta tabla y el campo "especificaciones" en la tabla "vehículos".

Tabla "ventas": Registra las ventas realizadas, incluyendo el ID de usuario, ID de vehículo, fecha, tipo de venta, pago, método de pago y cantidad de pagos.

Tabla "mensajes": Permite la gestión de un chat en la aplicación, donde se almacenan los mensajes enviados por los usuarios, junto con su ID de usuario y fecha.

Tabla "favoritos": Guarda los vehículos añadidos a favoritos por los usuarios. Se relaciona con las tablas "usuarios" y "vehículos" a través de los campos "id_usuario" e "id vehículo" respectivamente.

Integración de APIs y servicios externos.

En el desarrollo de nuestra aplicación se ha llevado a cabo la integración de varias APIs y el uso de varios servicios externos para mejorar la funcionalidad y la facilidad de poder ejecutar la aplicación casi en cualquier momento. Los principales aspectos de estas integraciones han sido:

Amazon Web Services (AWS).

Se ha utilizado AWS como servicio externo para alojar mucha parte de gran importancia de nuestra aplicación como la base de datos, la API Rest, las imágenes de los vehículos y ciertos archivos más. AWS proporciona una infraestructura confiable que garantiza el almacenamiento seguro de los datos y la disponibilidad del sistema en todo momento. Pese a la dificultad para llegar a realizar las configuraciones para subir la API o los archivos, AWS cuenta con un gran catálogo de servicios y muchas ayudas y documentaciones para facilitarte este proceso.

API de Chat GPT.

Se ha integrado la API de Chat GPT para proporcionar un asistente virtual en la aplicación. Esta API permite interactuar con un modelo de lenguaje de Chat GPT antiguo, lo que permite ofrecer respuestas automáticas y personalizadas a las consultas de los usuarios. Esto, lo hemos querido adaptar bajo el punto de vista de un asistente virtual al que tu puedas preguntarle, por ejemplo, ¿cuál es el mejor coche por un precio inferior a 10.000€?, el asistente virtual te responderá y ayudará a mejorar la experiencia del usuario al brindar información y guiarlo en la búsqueda de vehículos.

API de Stripe.

Se ha realizado la integración de tres claves de Stripe, aunque solo se use una. Una es la clave publicable, la cual puedes compartir, otra es la clave privada de prueba y otra es la clave secreta real, esta última es la que tenemos implementada y sirve para permitir pagos reales con tarjetas de crédito en la aplicación. Stripe es una plataforma de pagos en línea que ofrece una forma segura y confiable de procesar transacciones financieras. Al integrar estas claves y esta API, los usuarios pueden realizar compras de vehículos de manera segura, facilitando el proceso de pago.

Otras APIs.

Además de las APIs mencionadas anteriormente, se ha hecho uso de una API Rest adicional. Esta API ha sido desarrollada con el fin de relacionar la base de datos con la aplicación y mantener separados la parte del cliente con la parte del servidor.

En conclusión, la integración de estas APIs y servicios externos ha permitido mejorar a un gran nivel la funcionalidad de la aplicación ofreciendo un asistente virtual, opciones de pago reales y seguras y ayudándonos de un web service confiable para el almacenamiento de datos. Esto contribuye a tener una experiencia realista y completa en la aplicación.

Funcionalidades destacadas y distintas.


Nuestra aplicación se destaca por ofrecer una amplia gama de funcionalidades que no se suelen ver en aplicaciones de concesionarios o en TFGs. Ahora vamos a nombrar las funcionalidades más destacadas o distintas de cada pantalla haciendo un breve tour por nuestra aplicación.

Comenzamos con la pantalla principal: En esta pantalla aparecen en el lado izquierdo una gran cantidad de filtros de todo tipo y en el lado derecho una tabla con todos los vehículos. Os sorprendería la cantidad de aplicaciones de concesionarios que no tienen un gran catálogo de filtros avanzados, con nuestros filtros puedes encontrar el vehículo que buscas ya sea por el nombre normal o incluso intercalando con el tipo de transmisión, tracción, categoría del vehículo, caballos de potencia...

The interface features a sidebar with filters on the left and a table of results on the right. The filters include fields for Marca, Modelo, Precio Max., Kilometros Max., Combustible (Gasolina, Diesel, Otro), Año del modelo (1950-2023), Tipo, Categoría, Tipo de transmisión (Automatica, Manual), Potencia del motor, and a checkbox for Disponible. A 'FILTRAR' button is at the top right of the filter section, and a 'LIMPIAR CAMPOS' button is at the bottom right. The results table has columns for Vehículo, Combustible, Kilometros, Transmisión, Potencia, and Precio. The 'Kenton GTR' row is highlighted in blue. A 'MI CUENTA' button is located at the bottom right of the interface.

Vehículo	Combustible	Kilometros	Transmisión	Potencia	Precio
BMW 4 Series	Gasolina	60000	Manual	250	30000.0
Kia Rio	Gasolina	99000	Automática	400	22000.0
Chevrolet Cruze	Gasolina	70000	Automatica	160	12000.0
Hyundai Tucson	Diesel	1000	Manual	800	10000.0
Jeep Cherokee	Gasolina	80000	Manual	290	25000.0
Kawasaki Ninja	Gasolina	100	Manual	1000	10000.0
Kenton GTR	Gasolina	12	Manual	200	350000.0
Honda Civic	Diesel	260000	Automatica	160	14000.0
Toyota Camry	Gasolina	50000	Automatica	200	15000.0
Ford Explorer	Gasolina	10	Automatica	300	40000.0
Volkswagen Golf	Diesel	80000	Automatica	150	10000.0

Al pulsar en un vehículo nos encontramos una pantalla donde nos encontramos fotos de vehículos y datos sobre el vehículo, esto no es nada nuevo, en todo concesionario hay datos sobre los vehículos que venden, lo que no tienen muchos es un comparador, donde puedes escoger otro vehículo del mismo tipo (Coche o Moto), y ver todos sus detalles comparados en paralelo y coloreados según cual sea mejor o peor. A todo esto, se le añade un gráfico de radar por si te gusta más verlo por gráfico, aunque de esta forma no aparecen todos los datos, solo los más importantes.




Datos básicos


Marca	Categoría	Precio de Renting
Kia	Sedan	22000.0 €
Modelo	Matrícula	Precio de compra
Rio	1234 ABC	22000.0 €


Datos técnicos

Capacidad del depósito	Kilómetros	Año del modelo
60 L	99000 km	2018
Tipo gasolina	Tipo de transmisión	Potencia
Gasolina	Automática	400 cv
Consumo medio	Peso	Tracción
12 L / 100km	12000	Delantera


 **Protegido por garantía**

Este vendedor ofrece una garantía de un año para este vehículo. Recuerda que, a la entrega del vehículo este deberá de estar tal y como figura en el contrato de compraventa

[COMPARAR CON OTRO VEHICULO](#)
[COMPRAR O RENTAR](#)




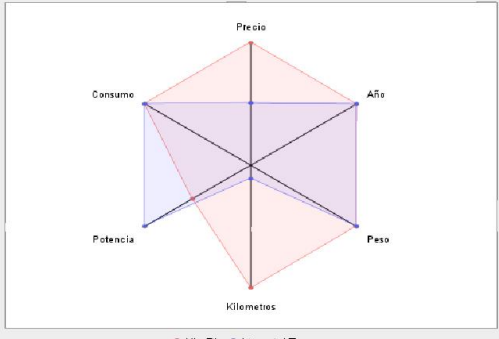
Kia
Rio



Hyundai
Tucson

22000.0 €	PRECIO	10000.0 €
Gasolina	COMBUSTIBLE	Diesel
12 L / 100km		12 L / 100km
60 L		60 L
Automática	TRANSMISION	Manual
99000 km	KILOMETROS	1000 km
2018	AÑO	2023
400 cv	POTENCIA	800 cv
12000	PESO	12000
Delantera	TRACCION	Delantera

COMPARATIVA DE CARACTERISTICAS



● Kia Rio ● Hyundai Tucson

Si te ha convencido el coche siempre podrás agregarlo a favoritos dando al corazón que aparece a la derecha del todo de la pantalla del vehículo, y si te ha gustado aún más podrás comprarlo. Ambas opciones se verán reflejadas en el apartado de mi cuenta, un apartado al que podrás acceder desde la pantalla principal. Aquí podrás gestionar tu cuenta, modificar tus datos, ver tus coches favoritos o ver tus coches comprados. También tendrás acceso a dos chats, uno online donde podrás conversar con los demás usuarios conectados y otro que ejerce de asistente virtual.

The screenshot shows a user profile page for 'Sergio'. The page has a purple header with a close button (X) in the top right corner. Below the header, there is a profile picture of a woman and the text 'Bienvenido Sergio'. The main content area is divided into three sections: 'Información personal', 'Otra información personal', and 'Información de login'. Each section has a pencil icon for editing. The 'Información personal' section contains fields for 'Nombre' (Sergio), 'Apellido' (Hernandez), and 'Teléfono' (610454973). The 'Otra información personal' section contains fields for 'Nacionalidad' (Española), 'Genero' (Masculino), and 'Fecha de nacimiento' (16 oct 2003). The 'Información de login' section contains fields for 'Correo electrónico' (sergiopozanco@gmail.com), 'Contraseña actual', and 'Nueva contraseña'. At the bottom of the main content area, there are two buttons: 'CERRAR SESION' and 'Actualizar'. To the right of the main content area, there is a vertical sidebar with four buttons: 'Chat GTP', 'Chat General', 'Mis Vehiculos', and 'Mis Favoritos'. At the bottom right of the page, there is a button labeled 'Volver a la pantalla principal'.

Bienvenido Sergio

Información personal

Nombre: Sergio, Apellido: Hernandez, Teléfono: 610454973

Otra información personal

Nacionalidad: Española, Genero: Masculino, Fecha de nacimiento: 16 oct 2003

Información de login


Correo electrónico: sergiopozanco@gmail.com, Contraseña actual, Nueva contraseña

CERRAR SESION, Actualizar

Chat GTP, Chat General, Mis Vehiculos, Mis Favoritos

Volver a la pantalla principal

Vehículo	Combustible	Kilometros	Transmisión	Potencia	Precio
Kia Rio	Gasolina	99000	Automática	400	22000.0



Datos técnicos

Capacidad del depósito
60 L

Tipo gasolina
Gasolina

Consumo medio
12 L / 100km

Kilómetros
99000 km

Tipo de transmisión
Automática

Peso
12000

Año del modelo
2018

Potencia
400 cv

Tracción
Delantera

Datos básicos

Marca
Kia

Modelo
Rio

Categoría
Sedan

Matrícula
1234 ABC

Estado
Disponible

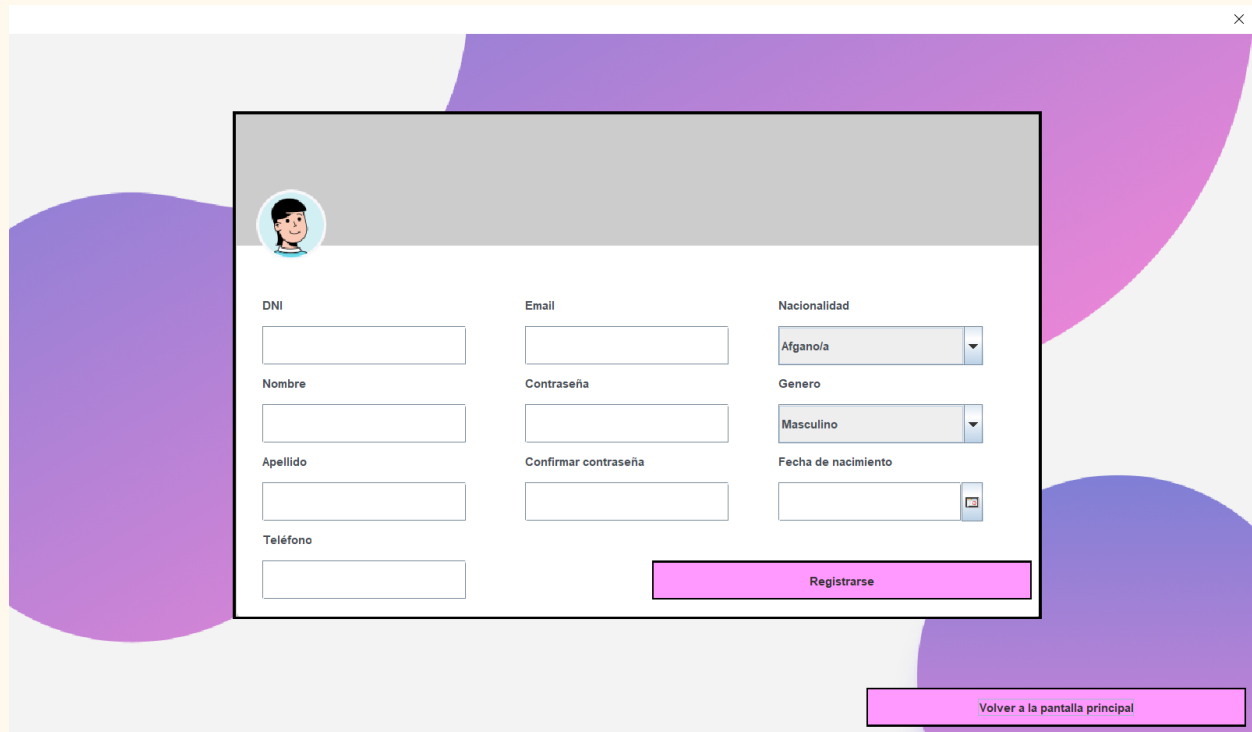
Precio
22000.0 €

Comprar

Quitar de favoritos

No te sale el apartado de Mi cuenta, eso quiere decir que no has iniciado sesión, en su defecto tendrás un botón el cual pone iniciar sesión y al presionarlo irás a la pantalla de iniciar sesión donde tienes una amplia gama de opciones. Aparte del inicio de sesión normal, tendrás acceso a permitir que se recuerden tus credenciales o incluso que se mantenga la sesión iniciada, por si eres un poco perezoso y no quieres estar iniciando sesión todo el rato.

Si estás introduciendo usuarios, correos, contraseñas y no te para de dar error, hay 2 opciones, o que te has olvidado tu contraseña o que no tienes cuenta. Tranquilo, tenemos solución para ambos problemas. En caso de que se te haya olvidado tu contraseña hay una opción de Has olvidado tu contraseña, simplemente sigue los pasos y podrás volver a tener tu cuenta disponible, procura que esta vez no se te olvide tu contraseña. En caso de que no tengas cuenta, verás una opción de registrarte, la cual te llevará a un formulario con el cual podrás empezar a ser miembro de MultiMotors.



A registration form overlay is displayed on a background with large, abstract purple and pink shapes. The form is enclosed in a white box with a thin black border and a close button (X) in the top right corner. At the top left of the form is a circular profile picture placeholder showing a cartoon character. The form fields are organized into three columns:

DNI	Email	Nacionalidad
<input type="text"/>	<input type="text"/>	<input type="text" value="Afgano/a"/>
Nombre	Contraseña	Genero
<input type="text"/>	<input type="text"/>	<input type="text" value="Masculino"/>
Apellido	Confirmar contraseña	Fecha de nacimiento
<input type="text"/>	<input type="text"/>	<input type="text"/>
Teléfono		
<input type="text"/>		

Below the form fields is a large pink button labeled "Registrarse". At the bottom right of the entire overlay area is a smaller pink button labeled "Volver a la pantalla principal".

Si ya has podido iniciar sesión y decides comprar un vehículo, simplemente tendrás que dar a comprar/rentar en la pantalla del vehículo y te llevará a un simple y breve formulario el cual será de rellenar dirección y método de pago.

RENTAR **COMPRAR**

Datos de dirección

País **Ciudad** **Población** **Código postal**
Espana Madrid Madrid 28011

Nombre de la calle **Número** **Bloque** **Planta** **Puerta**
Calle de la Huerta de Castañeda 33 3 3 3

Datos de pago ☒ **Tarjeta** ☐ **Transferencia**

Número de la tarjeta **Fecha de caducidad** **Titular de la tarjeta** **Método de pago**
MM/YY Al contado

Emisor de la tarjeta **CVV** **Número de cuotas**
VISA 1

IMPRIMIR FACTURA **FINALIZAR COMPRA**

En caso de haberlo rellenado con éxito, si has realizado una compra te llegará la factura a tu correo electrónico y en caso de que haya sido un renting te llegará un contrato a tu correo electrónico. Ambas opciones estarán siempre controladas a través de un código de verificación.

Factura

Nº Factura AB123345

Fecha Factura 05-06-2023

Información de la compra

Método de pago

Tarjeta de credito

Correo electrónico

fabianrodriguez27@gmail.com

Información de envío

Nombre completo

Fabián Rodríguez

Dirección de envío

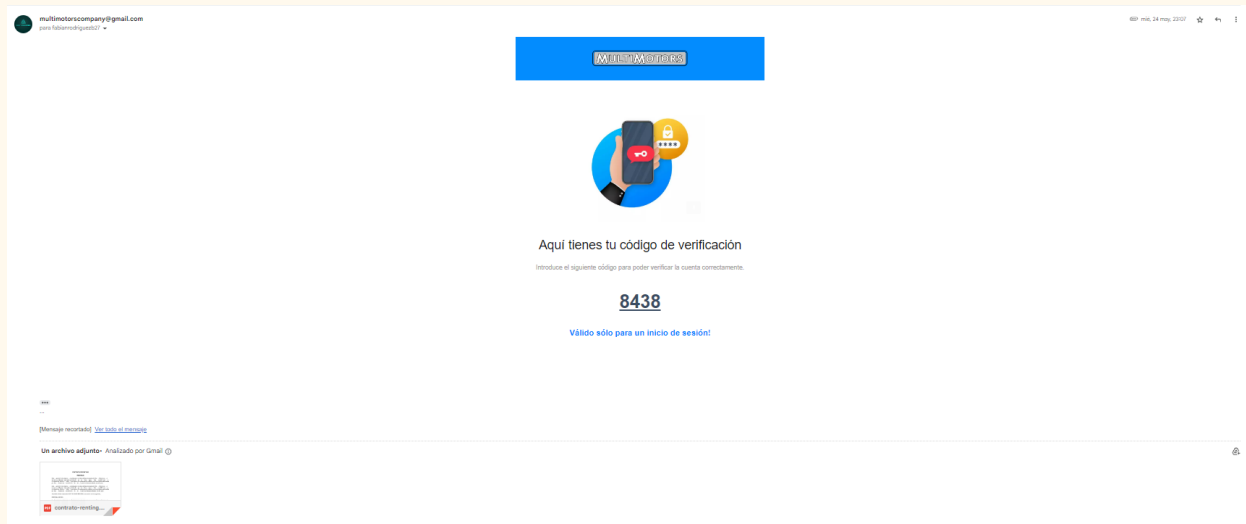
Calle Hermanamiento

11 4A

Alcorcon, 28922

Producto

Vehículo	Número de cuotas	Precio
Lamborghini Gallardo	1	350000.0
Total		350000.0



CONTRATO DE RENTING

REUNIDOS

Don ..., gerente de la empresa ..., constituida en escritura pública otorgada ante Don ..., Notario de ..., e inscrita en el Registro Mercantil con fecha de ... de ... de ..., libro ..., tomo ..., folio ..., con CIF núm. ..., y con domicilio social en ..., calle ..., con poderes de representación conforme a escritura pública autorizada por Don ..., Notario de ... con fecha de ... de ... de ..., a partir de ahora arrendador, de una parte;

Don ..., gerente de la empresa ..., constituida en escritura pública otorgada ante Don ..., Notario de ..., e inscrita en el Registro Mercantil con fecha de ... de ... de ..., libro ..., tomo ..., folio ..., con CIF núm. ..., y con domicilio social en ..., calle ..., con poderes de representación conforme a escritura pública autorizada por Don ..., Notario de ... con fecha de ... de ... de ..., a partir de ahora arrendatario, de otra parte;

Acuerdan celebrar el presente CONTRATO DE RENTING, de acuerdo con las siguientes,

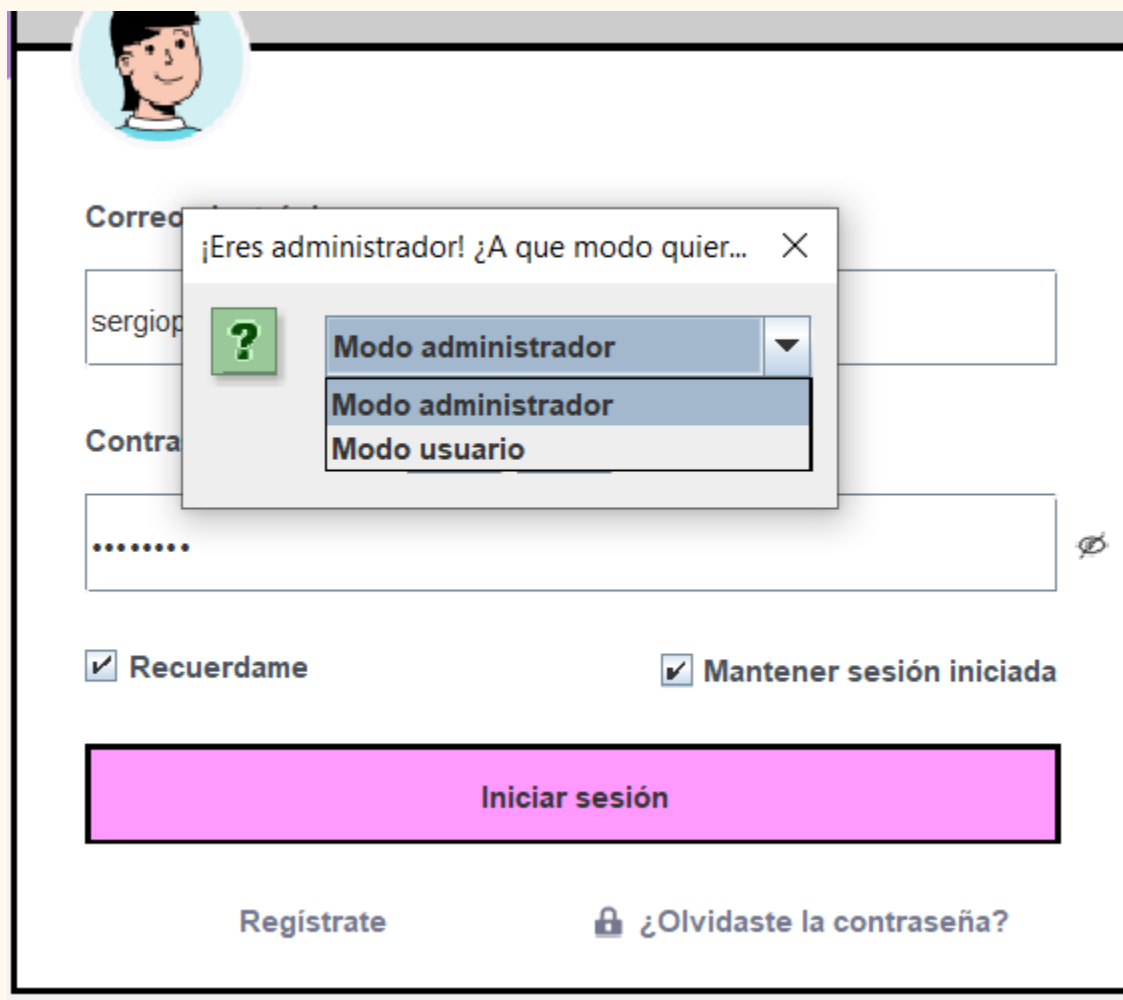
ESTIPULACIONES

- I. La empresa ... entrega en arriendo a la empresa ... la maquinaria de su propiedad especificada en el anexo I del presente contrato para (especificar la finalidad), sin que la misma pueda ser utilizada para fines distintos sin autorización expresa de la empresa arrendadora.
- II. La empresa arrendataria inspecciona la maquinaria confirmando que se encuentra en perfecto estado para el uso al que va a ser destinada.
- III. El presente contrato comienza el día de hoy, finalizando en fecha de ... de ... de
- IV. El precio del arrendamiento es de ... euros que se fracciona en ... mensualidades, abonando la empresa ... a la empresa arrendadora la cantidad de ... mensuales, en los cinco primeros días cada mes, ingresando dicha suma en la cuenta que la empresa ... tiene abierta en la Entidad Bancaria ..., sucursal ..., número de cuenta
- V. Por el presente contrato la empresa ... entrega a la empresa arrendadora la cantidad de ... euros en garantía del cumplimiento de las obligaciones. Dicha suma se reintegrará cuando la empresa arrendataria devuelva la maquinaria en buen estado, habiendo cumplido su obligación del pago de las rentas.
- VI. La empresa arrendadora se obliga al mantenimiento de la maquinaria, seguros e impuestos. Cuando la reparación requiera de un mínimo de ... días, la empresa arrendadora se obliga a reemplazar la máquina por otra de similares características temporalmente hasta que finalice la reparación, o definitivamente.
- VII. La empresa arrendadora se obliga a la evicción y saneamiento por vicios ocultos.
- VIII. Si la empresa arrendataria incumple su obligación de pago de las rentas, la empresa arrendadora podrá exigir las mantenimientos la vigencia del contrato o podrá pedir la resolución del mismo con la restitución de la maquinaria y exigiendo el pago de las cantidades de las rentas vencidas.
- IX. Para resolver cualquier cuestión derivada del presente contrato las partes se someten expresamente a los Tribunales de ..., con renuncia del fuero propio.

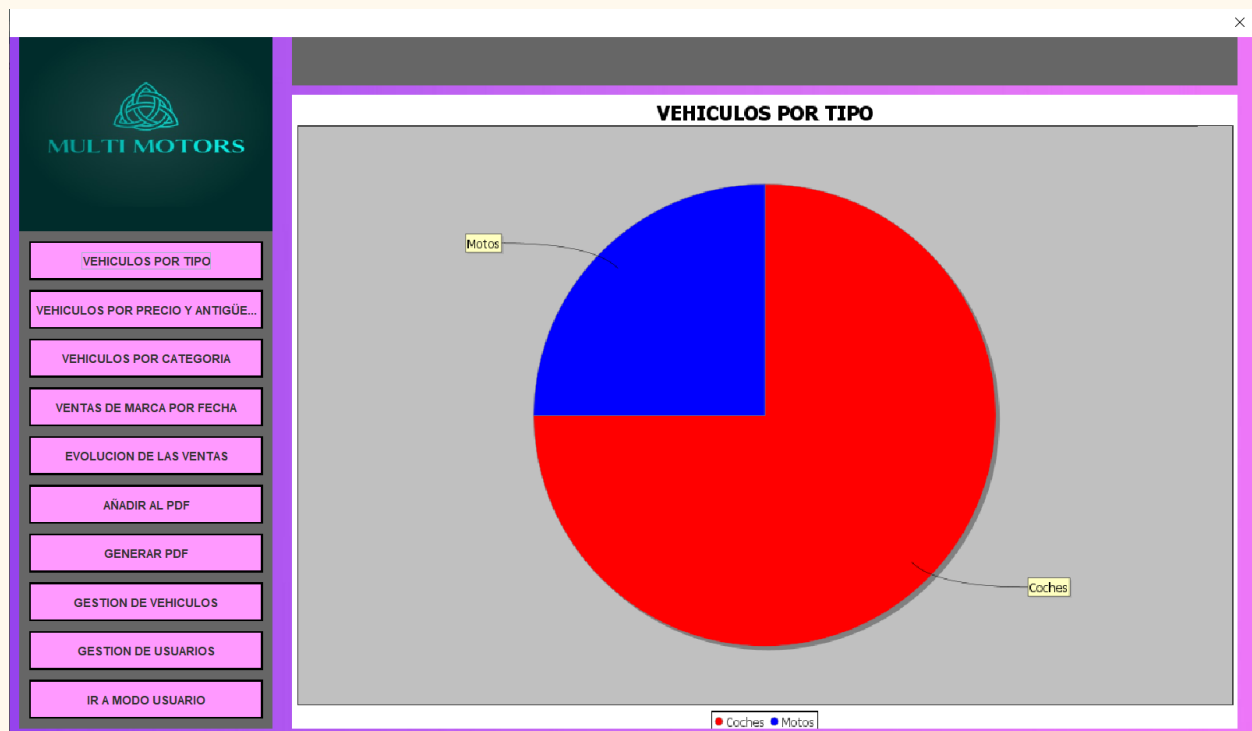
Conformes, firman el presente contrato de renting, por duplicado y a un solo efecto, Don ..., como gerente de la empresa ..., y Don ..., como gerente de la empresa

En ... a ... de ... de

Muy bien, has comprado un coche, para nada de esto sería posible sin los administradores. Los administradores tienen una importante labor en la aplicación.



Si has accedido al modo administrador es porque tu cuenta tiene algún tipo de permiso de administrador, ya sea ser un administrador completo o ser un simple empleado que solo puede ver gráficos. Este es el panel del administrador, por el cual puedes acceder a diversos gestores como el de inventario o el de usuarios, puedes volver al modo usuario o puedes ver hasta 5 gráficos distintos según vayas escogiendo en la parte superior de la barra del menú de la izquierda.



Si pulsas en gestionar vehículos, se te abrirá una pantalla con una tabla con todos los vehículos del concesionario y a la izquierda muchos campos de texto con varias opciones. Si pulsas un vehículo en la tabla, todos sus datos se rellenarán en los campos de texto de la izquierda pudiendo así actualizar alguno de ellos o borrar el vehículo entero. También tendrás opción de insertar un vehículo simplemente rellenando esos campos de texto y dando al botón de insertar. Debajo de la tabla vemos varias opciones, todas relacionadas con Excel. Nuestro gestor te permite desde Exportar a formato Excel la tabla con todos los vehículos, como descargar un excel preformateado que luego puedes rellenar, importarlo y añadir todos esos vehículos. Estas opciones sobre todo se usarán para cuando el concesionario necesita añadir 10 o 20 coches y no quieres hacerlo de 1 en uno porque tardarás mucho, de esta forma simplemente te tendrías que preparar un Excel, importarlo y darle a insertar. Eso sí, recuerda que sea con el formato establecido.

El gestor de usuarios es similar por no decir idéntico, pero en vez de gestionar vehículos, gestionas los datos de cada usuario, incluyendo sus permisos.

Tipo de vehículo

Marca

Modelo

Categoría

Matrícula

Precio

Tipo de combustible

Tipo de transmisión

Kilómetros

Consumo

Año del modelo

1950

Potencia

Tracción

Peso

Capacidad del tanque

+

↺

🗑️

FILTRAR

INSERTAR IMAGENES

LIMPIAR CAMPOS

Tipo	Marca	Modelo	Estado	Categoría	Matrícula
Coche	BMW	4 Series	Disponible	Clasico	345 JKL
Coche	Kia	Rio	Disponible	Sedan	1234 ABC
Coche	Chevrolet	Cruze	Disponible	Sedan	768 MNO
Coche	Hyundai	Tucson	Disponible	Clasico	111 ABC
Coche	Jeep	Cherokee	Disponible	SUV	2234 STU
Moto	Kawasaki	Ninja	Disponible	Circuito	122 MBC
Coche	Kenton	GTR	Disponible	Sedan	1111AAAA
Coche	Honda	Civic	Disponible	Sedan	5567 XTW
Moto	Kenton	GTR	Disponible	Montaña	444 MBC
Coche	Toyota	Camry	Disponible	Sedan	123 XYZ
Coche	Ford	Explorer	Disponible	SUV	456 ABC
Coche	Volkswagen	Golf	Disponible	Sedan	789 DEF

IMPORTAR UN EXCEL

INSERTAR VEHICULOS DE EXCEL

FORMATO DE EXCEL A IMPORTAR

EXPORTAR A EXCEL

DNI

Nombre

Apellido

Telefono

Email

Genero

Nacionalidad

Fecha de nacimiento

Pais

Ciudad

Poblacion

Codigo postal

Nombre de la calle

Numero

Bloque

Planta

Puerta

+

↺

🗑️

FILTRAR

GESTIONAR PERMISOS

LIMPIAR CAMPOS

DNI	Nombre	Apellido	Teléfono	Email	Genero
09847303Z	Fabián	Rodríguezz	635406489	fabianrodriguez27@gmail.com	Femenino
09847303S	Fabián	Rodríguezz	635406489	fabianrodriguez@gmail.com	Femenino
09847303H	Fabian	Rodriguez	632406489	fabia222@gmail.com	Masculino
70834599X	Juan	Carlos	628981267	juancpereira@gmail.com	Otro
09847303P	Periko	Gallardo	632406489	periko@gmail.com	Femenino
11111111C	David	Pirez	632406489	david@gmail.com	Masculino
09847303A	Rodolfo	Bogado	632406489	rodol@gmail.com	Masculino
70834592C	Sergio	Hernandez	610454973	sergiopozanco@gmail.com	Masculino

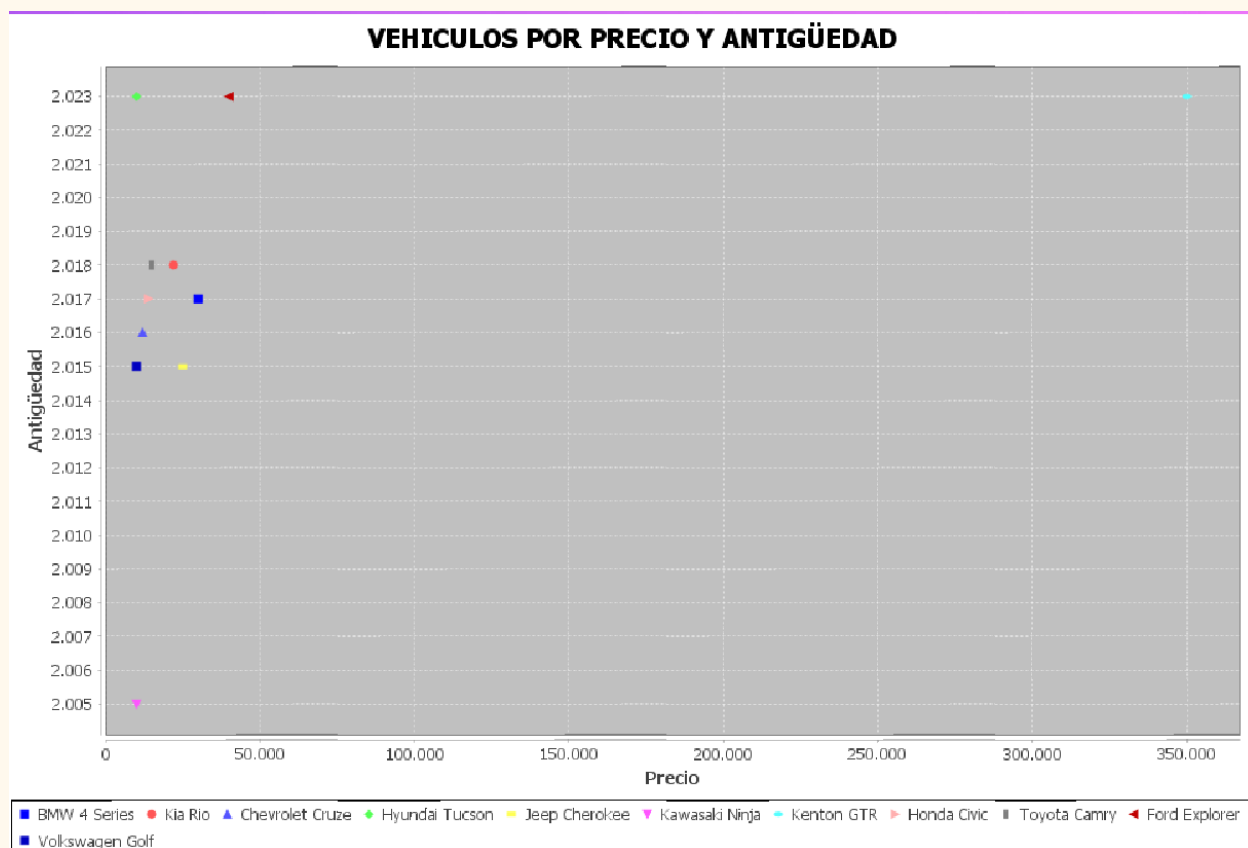
IMPORTAR UN EXCEL

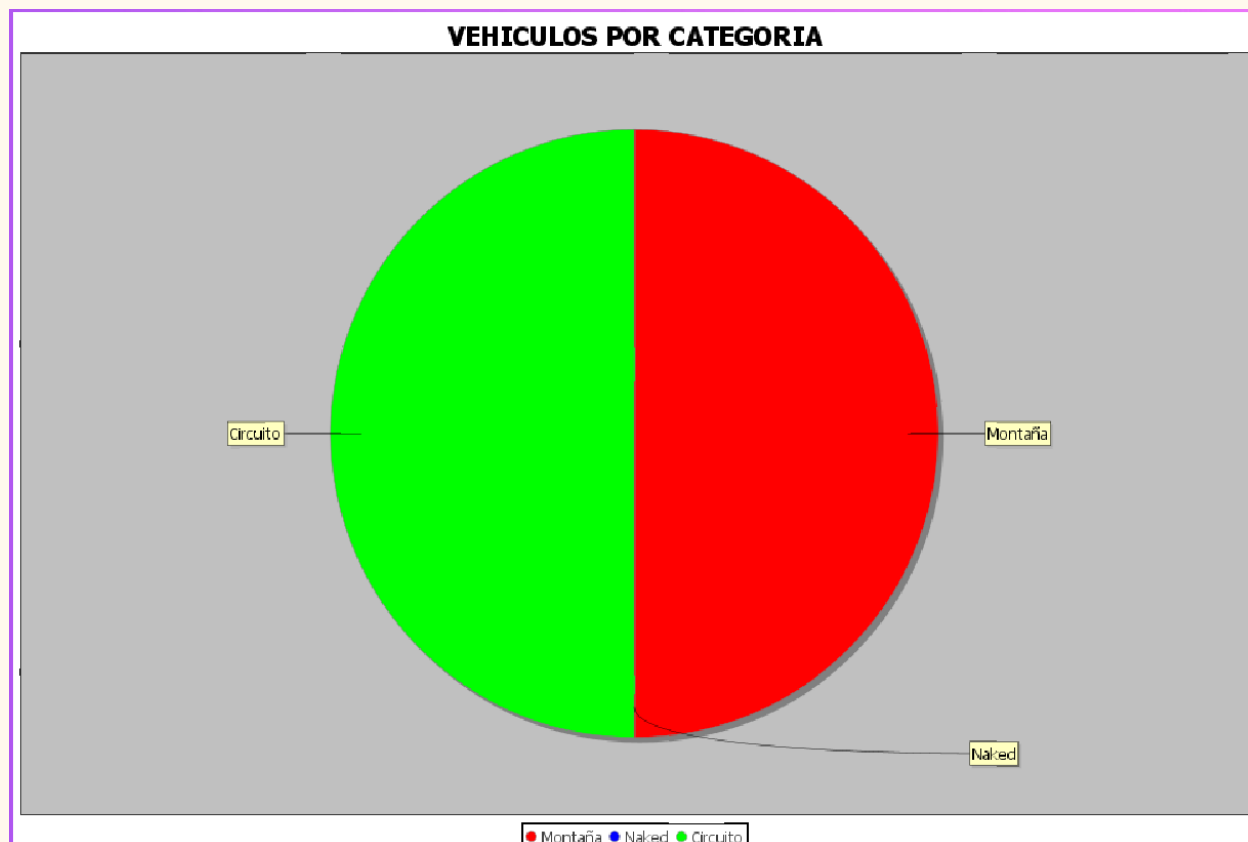
INSERTAR USUARIOS DE EXCEL

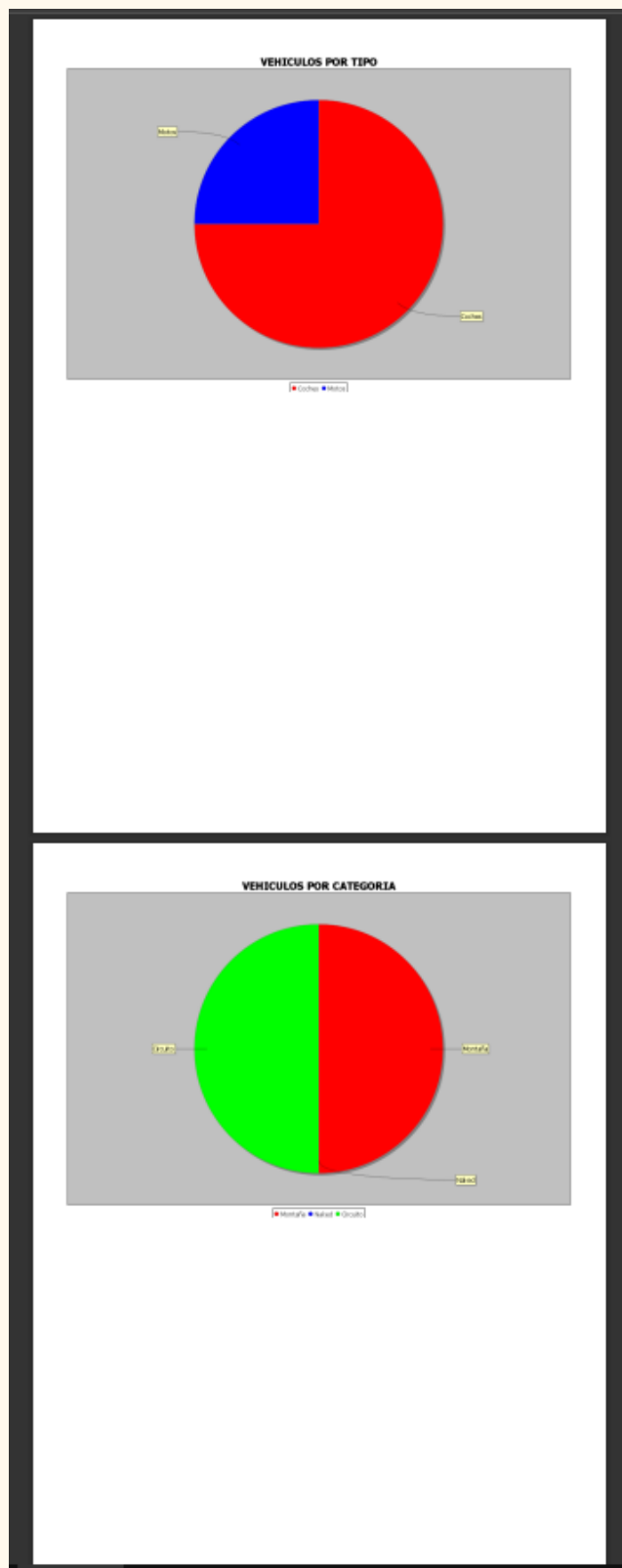
FORMATO DE EXCEL A IMPORTAR

EXPORTAR A EXCEL

Sin embargo, si eres un simple empleado como hemos mencionado antes y sólo puedes ver gráficos, a lo mejor es porque necesitas sacar el gráfico o crearlo en Excel, nosotros te lo facilitamos y simplemente tendrías que ir añadiendo los gráficos que tú quieras dando al botón de añadir PDF y luego podrás descargar este PDF, de esta forma tendrás un PDF con todos los gráficos que has ido añadiendo.







PRUEBAS DE ACCESIBILIDAD Y USABILIDAD.

Para este apartado, se han llevado a cabo una serie de evaluaciones para asegurar que nuestra aplicación cumple con unos buenos estándares de usabilidad y accesibilidad. Cabe destacar que no hemos tenido el tiempo suficiente para realizar todas las pruebas que queríamos, sobre todo de accesibilidad, pero aun así se han logrado cubrir muchas.

Pruebas de Usabilidad.

- Se han realizado pruebas con usuarios reales para comprobar si realmente la aplicación era de fácil uso.
- Realizando las últimas pruebas en la aplicación, se identificaron varias áreas de mejora como por ejemplo los diálogos de mensajes de error o la disposición de algunos componentes.
- Se realizaron ajustes en el diseño y en el código por problemas de duplicación de pantallas

Pruebas de Accesibilidad.

- Se han intentado seguir algunas pautas establecidas por la Web Content Accessibility Guidelines “WCAG”.
- Se han realizado pruebas de accesibilidad utilizando muchas evaluaciones manuales.
- Se han intentado corregir problemas como el uso inadecuado de algunos colores, la estructura de los componentes y el tamaño de la fuente del texto.

En cuanto a la accesibilidad, podemos destacar que nuestra aplicación es:

- Perceptible: Utiliza contraste de color suficiente entre el fondo y el texto principal para una legibilidad sencilla.

- Operable: Proporcionamos el tiempo que quieran a los usuarios para que lean y utilicen el contenido. La aplicación la manejan ellos.
- Compresible: Utilizamos un lenguaje claro y sencillo para que los usuarios tengan una fácil comprensión y organizamos el contenido de una manera lógica y coherente.

Otras pruebas y mejoras implementadas.

- Se intentó optimizar el tiempo de carga de la aplicación mediante un código bien estructurado, un código limpio y un buen servicio externo de almacenamiento de base de datos para mejorar la experiencia del usuario.
- Se intentaron hacer más intuitivos y fáciles de usar los formularios y sus campos de entrada.

DESPLIEGUE DEL PROYECTO.

El despliegue de nuestro proyecto va a tratar de todos los procesos que hemos tenido que ir realizando para llevar nuestra aplicación de un entorno de desarrollo a un entorno de producción para facilitarles todo lo posible la accesibilidad a los usuarios finales. Nosotros lo hemos dividido en los siguientes apartados:

Elección del entorno de despliegue para el desarrollador: Al principio del desarrollo tuvimos que determinar el entorno que iba a ser el encargado de desplegar nuestra aplicación. Nosotros elegimos Amazon Web Services (AWS), un proveedor de servicios en la nube en el que finalmente acabamos subiendo todos nuestros archivos que necesitaban algún tipo de aplicación externa para funcionar correctamente como podía ser la API Rest.

Configuración de la base de datos y de la API Rest: Nos debemos asegurar de que la configuración de la base de datos y de la API Rest es la correcta. En el caso de la base de datos que esté permitido la creación, modificación y lectura en cualquier momento y en el caso de la API que esté correctamente configurada y conectada a la base de datos.

Configuración del entorno de despliegue: Tuvimos que establecer una compleja configuración tanto para el inicio de Amazon Web Services, para almacenar la base de datos, para almacenar la API Rest y para configuración Amazon S3, un servicio de AWS que te permite almacenar archivos.

Pruebas de pre-despliegue: Es importante realizar pruebas para verificar que funcione correctamente en el entorno de desarrollo. Esto implica probar la funcionalidad y el rendimiento.

Pruebas de post-despliegue: Una vez que se ha llevado a cabo correctamente el apartado anterior, es el momento de transformar el código en una aplicación ejecutable para Windows, nosotros hemos usado INNO SETUP para generar un instalador a través del .JAR. Importante decir que

para generar el .JAR correctamente deberás entrar en el CMD y dirigirte a la ruta donde se encuentre el proyecto de NetBeans usando `cd ruta`. Tras estar en la ruta tendrás que ejecutar ese comando: `mvn clean compile assembly:single`. Esto sirve para generar correctamente el .jar de nuestro archivo Maven con todas sus dependencias correctamente.

Mantenimiento: Una vez ya están todos los apartados anteriores completados y funcionando perfectamente damos paso al mantenimiento de la aplicación. Toda aplicación necesita recibir tareas de mantenimiento periódicas e ir recibiendo actualizaciones o mejoras de rendimiento.

Los usuarios finales terminarán descargando un instalador para ejecutar la instalación en su ordenador y no tendrán que realizar ningún tipo de tarea compleja para que la aplicación funcione correctamente. Ellos simplemente se limitarán al uso de la aplicación a través de la interfaz que hemos diseñado, nosotros, los desarrolladores, somos los encargados de realizar la configuración y administración de todos los archivos de AWS para garantizar que la aplicación esté disponible y funcione correctamente en todo momento para los usuarios.

PROBLEMAS ENCONTRADOS.

Recuérdame y Mantener sesión iniciada: Uno de los desafíos fue implementar una funcionalidad que permitiera a los usuarios recordar las credenciales de su inicio de sesión para acceder rápidamente a la aplicación o incluso que se mantuviese la sesión iniciada. Se implementó un sistema por el cual se accede a los ficheros de preferencias del ordenador, localizados en el Editor del Registro, y se crea una carpeta donde se guardan tus credenciales y una variable de mantener sesión iniciada, dependiendo de cómo se encuentre en esa carpeta se hará la función de recuérdame y de mantener sesión iniciada.

Gráfico de radar: Para proporcionar una representación visual de las características y atributos de los vehículos en comparación con otro vehículo, se implementó un gráfico de radar. Este gráfico permite a los usuarios visualizar las diferentes características de los vehículos en un solo vistazo, facilitando la comparación y la toma de decisiones. Para lograrlo fue complicado ya que la librería no disponía directamente del método para crear este gráfico, pero investigando nos dimos cuenta de que esta librería (jfreechart) tenía un github donde tenía algunos gráficos subidos de los que no tenían un método directo, entonces pudimos hacernos una idea y modificarlos un poco para hacernos con el gráfico que nosotros queríamos.

Configuración de AWS: La utilización de Amazon Web Services (AWS) como servicio externo planteó el desafío de configurar correctamente los recursos necesarios, ya que puede existir la opción de que, si configuras algo que no deberías, te puede llegar a cobrar ya que una de las desventajas de AWS es que tienes que tener un método de pago vinculado. La configuración de los servicios como la API Rest, el almacenamiento de la base de datos y el almacenamiento de imágenes de vehículos y de los archivos fue también un desafío ya que eran configuraciones complejas. Se siguió una configuración detallada de los servicios de AWS, asegurando la correcta integración y funcionamiento de los componentes externos.

Diseño de las interfaces: El diseño de las interfaces de usuario fue un total desafío ya que con Java Swing no tienes total libertad para realizar un buen diseño. Se adoptaron buenas prácticas de diseño de interfaces, y se cogieron plantillas e ideas de otros proyectos que nos sirvieran como guía o ayuda para realizar los diseños.

API Rest: La implementación de una API Rest fue fundamental para la comunicación entre la aplicación y los servicios externos. Pero costó mucho trabajo que estuviese al punto en el que se encuentra al final del desarrollo ya que nos encontrábamos continuamente con errores a los que se le daba solución y surge otro error distinto.

Ambos Chats: Ambos chats fueron complejos, tanto el Chat online para la comunidad como el Chat GPT para el asistente virtual. Sobre todo, nos costó llegar a la idea de cómo hacerlo ya que teníamos muchas cosas pensadas, pero todas eran algo completas. Finalmente, para el Chat online decidimos hacerlo con la base de datos y de esa forma se nos facilitó mucho el trabajo, pero nos costó llegar a esa finalidad. En cuanto al Chat GPT tuvimos que generar una clase de PHP para la gestión de mensajes recibidos y respuestas y tuvimos que sacar la API privada de Chat GPT. También tuvimos un pequeño problema y es que nosotros hemos estado trabajando con un repositorio público de github y si subíamos la API privada al github público se nos rompía y teníamos que volver a generar otra.

Generar factura: En cuanto a la generación de la factura, lo difícil no fue enviarla por correo, lo difícil fue crear el boceto de la factura. Utilizamos una famosa librería que se llama iText7 y tuvimos que realizar el diseño de la factura desde cero. El diseño de la factura está todo dividido en tablas y columnas.

CONCLUSIONES.

Evaluación crítica de los resultados obtenidos.

Durante el desarrollo de MultiMotors se ha ido llevando una evaluación crítica para llegar a los resultados que queríamos desde un principio y creo que los hemos superado con creces. En general, la aplicación ha cumplido con los objetivos establecidos en el anteproyecto y ha demostrado ser totalmente funcional e incluso realista. Sin embargo, no todo es perfecto, también hemos identificado diversos aspectos que se podrían llegar a mejorar para así perfeccionar la experiencia del usuario.

En términos de usabilidad la aplicación funciona prácticamente a la perfección. No obstante, existe la necesidad de mejorar el diseño de las interfaces del usuario para así poder brindar una experiencia visual más atractiva y sobre todo más moderna. También se ha llegado a observar en alguna ocasión que la aplicación puede presentar algo de lentitud para la realización de alguna funcionalidad como la de enviar un correo electrónico al usuario.

Propuestas de mejoras.

En base a la evaluación crítica realizada anteriormente, proponemos una serie de mejoras que podremos realizar próximamente para la aplicación de MultiMotors:

Rediseñar las interfaces: Queremos realizar un rediseño de las interfaces de usuario y asimismo un rediseño de la aplicación completa ya que para conseguir este rediseño lo más atractivo y moderno posible lo haríamos con WPF Y XAML. Esto daría lugar a desarrollar la aplicación también en Visual Studio con C# utilizando .Net 6.

Optimización del rendimiento: Queremos realizar ajustes y optimizaciones en el rendimiento de la aplicación, especialmente en la carga de datos, llamadas a la base de datos o ejecución de

métodos externos. Esto podría lograrse mediante el uso de técnicas de caché y tendríamos que estudiarlas para ejecutarlas correctamente.

Desarrollos futuros.

Además de las propuestas de mejoras mencionadas anteriormente, existen algunas mejores más que podrían enriquecer aún más nuestra aplicación:

Personalización de la aplicación: Esta opción la barajamos al final del desarrollo de la aplicación, que cada usuario pudiese elegir un tema y tener la aplicación “personalizada” a su manera. Finalmente, no dio tiempo a hacerlo y sería algo que se podría añadir en una futura actualización.

Aplicación multiplataforma: También nos gustaría desarrollar la aplicación en diversos dispositivos, no solo en el escritorio de Windows. Esto se iría realizando en un gran periodo de tiempo. Los dispositivos que podrían tener acceso a MultiMotors en un futuro podrían ser: Android (realizado a través de Android Studio y Kotlin), iOS (realizado a través de XCode y Swift) e incluso poder desarrollarla para una página web con HTML.

Integración de inteligencia artificial: Integración de técnicas de inteligencia artificial para mejorar la precisión de las recomendaciones de vehículos, la detección de fraudes y la personalización de la experiencia del usuario.

Integración con servicios de terceros: Pese a ya tener integración con varios servicios de terceros, está la posibilidad de seguir explorando cual podría encajar bien con MultiMotors.

En conclusión, nuestra aplicación, Multimotors, ha demostrado éxito en cuanto a funcionalidad se refiere. Sin embargo, como cualquier aplicación del mundo, existen grandes oportunidades de mejora que con algo más de tiempo y dedicación se podría llegar a realizar. Las propuestas de

mejora y los desarrollos de futuro mencionados anteriormente ayudarían a potenciar el nivel de la aplicación y otorgar una gran experiencia al usuario.

BIBLIOGRAFÍA.

En cuanto a la bibliografía, obviamente lo que más hemos utilizado han sido videotutoriales de YouTube, desde realizar la configuración de AWS casi al principio del proyecto como realizar lo del servicio de pago de Stripe casi al final del proyecto, pero también utilizándolo para realizar funcionalidades cornetas, ejemplos de diseños, etc...

Aquí algunos enlaces:

Stripe: [1. What is stripe payment?](#)

[Live Coding #1: Stripe payment integration - By @MarcoCodes](#)

Recuérdame en Login: [Login Form With Remember Me In Java](#)

Transformar un .dar a .exe: [Generar instalador y exe de un .JAR | Java + Apache NetBeans 12.2](#)

Diseño: [Colocar imágenes en JFrame Java](#)  [|| JPanel y JLabel || 2022](#) 

AWS: [Configurando mi primera PHP webapp sobre Elastic Beanstalk en AWS](#)

[TUTORIAL AWS - Crear una base de datos MySQL en RDS](#)

[🔗 ASÍ DE FÁCIL PUEDES CARGAR, DESCARGAR Y LISTAR ARCHIVOS DE AWS S3 USANDO SPRING BOOT Y JAVA](#) 

También hemos utilizado stackoverflow: <https://stackoverflow.com/>
Stackoverflow es una plataforma de preguntas y respuestas relacionadas con programación y desarrollo de software. Sirve sobre todo para la resolución de problemas comunes y de solución rápida.

Hemos utilizado algunos repositorios de github, el más destacado ha sido el repositorio de la librería jfreechart, el cual te permite generar muchos otros tipos de gráficos que no existen todavía en la librería.

jFree: <https://github.com/jfree>