

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE CIENCIAS Y SISTEMAS**  
**ANÁLISIS Y DISEÑO 1**  
**SEGUNDO SEMESTRE 2023**

**GRUPO 3**  
**PROYECTO 2**

<b>Nombre</b>	<b>Carnet</b>
Mario Cesar Moran Porras	202010793
Esdras Rodolfo Toc Hi	201807373
Fernando José Jr. Serrano Mejia	201701039
Gerson Rubén Quiroa del Cid	202000166
Jose Fernando Alvarez Morales	201900528
Elder Anibal Pum Rojas	201700761

# DESCRIPCIÓN DEL MODELO BRANCHING

## Gitflow

Es un modelo de branching que se utiliza comúnmente en el control de versiones con Git. Fue propuesto por Vincent Driessen y se centra en la gestión de flujos de trabajo de desarrollo de software. A continuación, te proporciono una descripción de cómo funciona el modelo de branching Gitflow:

### 1. Rama principal (master):

La rama master es la rama principal y estable del proyecto. Debe contener únicamente versiones estables y probadas del software. La liberación de una nueva versión se fusiona en esta rama.

### 2. Rama de desarrollo (develop):

La rama develop es donde se integran las nuevas características y se realiza la mayoría del desarrollo. Es una rama en constante evolución y puede contener características aún en desarrollo.

### 3. Ramas de características (feature branches):

Para cada nueva característica o mejora, se crea una rama de características separada desde develop. Esto permite a los desarrolladores trabajar en paralelo en diferentes características sin interferir entre sí.

### 4. Ramas de liberación (release branches):

Antes de lanzar una nueva versión del software, se crea una rama de liberación desde develop. En esta rama, se realizan las pruebas finales y se corrigen los errores. No se agregan nuevas características en esta etapa.

### 5. Ramas de corrección (hotfix branches):

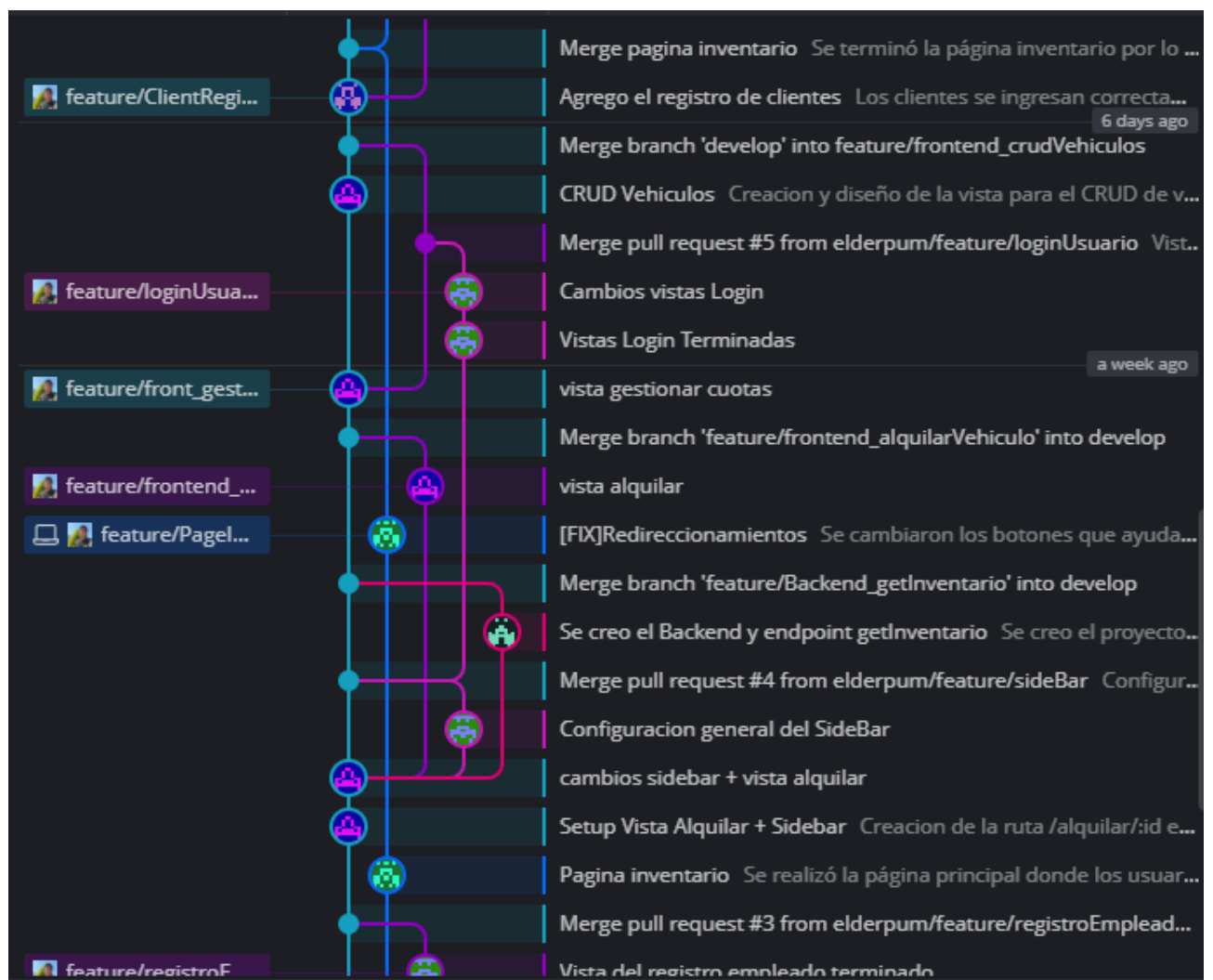
Si se detecta un error crítico en la rama master que debe corregirse de inmediato, se crea una rama de corrección desde master. Una vez corregido el error, se fusiona tanto en master como en develop.

## Flujo de trabajo en Gitflow:

1. Se comienza en la rama develop, donde se desarrollan las nuevas características.
2. Cuando una característica está lista, se crea una rama de características y se trabaja en ella.
3. Una vez que la característica está completa, se fusiona de nuevo en la rama develop.

4. Cuando se está listo para lanzar una nueva versión, se crea una rama de liberación desde develop. Se realizan pruebas y se corrigen errores.
5. Una vez que la versión de liberación está lista, se fusiona en master y develop.
6. Si se detecta un error crítico en la rama master, se crea una rama de corrección desde master, se corrige y se fusiona de nuevo en master y develop.
7. El ciclo se repite para nuevas características y versiones.



Gitflow proporciona una estructura organizada para el desarrollo de software, facilitando la gestión de versiones y la colaboración en equipos de desarrollo. Sin embargo, es importante recordar que Gitflow es solo uno de varios modelos de branching disponibles en Git, y su elección debe depender de las necesidades específicas de tu proyecto y equipo.






# TECNOLOGÍAS UTILIZADAS

## 1. Tecnologías En El Backend

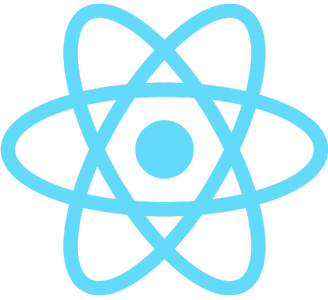


	<p><b>NodeJS</b></p> <p>Node.js es un entorno de ejecución de código JavaScript en el lado del servidor. Esto significa que permite ejecutar código JavaScript en el servidor web, en lugar de solo en el cliente.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"><li>• Node.js es muy rápido y escalable.</li><li>• Es fácil de aprender y usar.</li><li>• Es compatible con una amplia gama de tecnologías y frameworks.</li></ul>
	<p><b>Express</b></p> <p>Express es un framework web para Node.js que proporciona una API minimalista y fácil de usar. Esto significa que facilita la creación de aplicaciones web con Node.js.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"><li>• Express es muy ligero y eficiente.</li><li>• Es fácil de aprender y usar.</li><li>• Es compatible con una amplia gama de tecnologías y librerías.</li></ul>
	<p><b>MySQL</b></p> <p>Express es un framework web para Node.js que proporciona una API minimalista y fácil de usar. Esto significa que facilita la creación de aplicaciones web con Node.js.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"><li>• Express es muy ligero y eficiente.</li><li>• Es fácil de aprender y usar.</li><li>• Es compatible con una amplia gama de tecnologías y librerías.</li></ul>

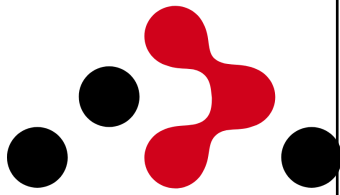
	<p><b>Morgan</b></p> <p>Morgan es una librería que proporciona un middleware para registrar las solicitudes HTTP. Esto significa que permite guardar información sobre las solicitudes HTTP, como la dirección IP del cliente, el método HTTP, la ruta y los encabezados.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"> <li>• Morgan se puede utilizar para realizar un seguimiento del tráfico web.</li> <li>• Morgan se puede utilizar para depurar errores.</li> <li>• Morgan se puede utilizar para generar informes de rendimiento.</li> </ul>
	<p><b>JSON Web Token</b></p> <p>JSON Web Token (JWT) es un estándar para la transmisión segura de información entre el cliente y el servidor. Los JWT se utilizan para transmitir datos de autenticación, como el nombre de usuario y la contraseña, entre el cliente y el servidor.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"> <li>• JWT se utilizan para autenticar a los usuarios.</li> <li>• JWT se utilizan para autorizar a los usuarios a realizar acciones.</li> <li>• JWT se utilizan para transmitir datos de sesión.</li> </ul>
	<p><b>CORS</b></p> <p>CORS (Cross-Origin Resource Sharing) es una medida de seguridad que limita las solicitudes HTTP entre dominios. Esto significa que solo se permite que las solicitudes HTTP se realicen entre dominios que han sido autorizados.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"> <li>• CORS se utiliza para evitar ataques de cross-site scripting (XSS).</li> <li>• CORS se utiliza para evitar ataques de cross-site request forgery (CSRF).</li> <li>• CORS se utiliza para proteger la privacidad de los usuarios.</li> </ul>

 <p>bcrypt in plain JavaScript</p>	<p><b>Bcrypt</b></p> <p>Bcrypt es una función hash de contraseña que se utiliza para proteger las contraseñas de los usuarios. Bcrypt genera un hash de la contraseña que es muy difícil de descifrar.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"> <li>• Bcrypt se utiliza para proteger las contraseñas de los usuarios de ataques de fuerza bruta.</li> <li>• Bcrypt se utiliza para cumplir con los requisitos de seguridad de las contraseñas.</li> <li>• Bcrypt se utiliza para proporcionar una capa adicional de seguridad a las aplicaciones web.</li> </ul>
---	--

 	<p><b>AWS SDK</b></p> <p>AWS SDK es un conjunto de herramientas que permite a los desarrolladores interactuar con los servicios de Amazon Web Services. Los servicios de AWS incluyen servicios de almacenamiento, bases de datos, computación en la nube, análisis de datos y mucho más.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"> <li>• AWS SDK se utiliza para crear aplicaciones que se ejecutan en la nube de AWS.</li> <li>• AWS SDK se utiliza para integrar aplicaciones existentes con los servicios de AWS.</li> <li>• AWS SDK se utiliza para automatizar tareas en los servicios de AWS.</li> </ul>
---	---

## 2. Tecnologías En El Frontend

	<p><b>React</b></p> <p>React es una librería Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"><li>• Consideramos que react facilita el proceso de escritura de componentes.</li><li>• Es fácil de aprender y de implementar.</li><li>• Tiene un gran número de librerías fáciles de usar.</li></ul>
	<p><b>Material UI</b></p> <p>Material UI es una biblioteca de componentes de interfaz de usuario para React, diseñada para ayudar a los desarrolladores a construir aplicaciones web modernas</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"><li>• Consideramos que Material UI es una librería con una amplia gama de componentes y utilidades lo que la hacen dinámica y adaptable a nuestras necesidades.</li></ul>
	<p><b>styled-components</b></p> <p>Styled Components es una biblioteca de javascript frecuentemente usada en React con la que puedes crear componentes de estilización personalizados, te permite escribir código CSS dentro de una función Javascript, en lugar de escribir CSS en un archivo separado.</p> <p>Motivos de uso:</p> <ul style="list-style-type: none"><li>• La utilizamos ya que nos brinda la facilidad de insertar componentes con css sin necesidad de utilizar archivos css lo cual nos da agilidad al momento de trabajar.</li></ul>



### **react-router-dom**

React Router es una librería que se utiliza bastante con React y que nos facilita el proceso de definir las rutas de navegación dentro de nuestra aplicación.


Motivos de uso:

- La utilizamos ya que nos brinda la facilidad de utilizar rutas para componentes y navegar entre ellas con facilidad.



## MOCKUPS

- **Login Cliente:**



### LOGIN

Nombre de usuario \*

Contraseña \*

☐ código de acceso


[INGRESAR](#) [REGISTRARSE](#)

[Enviar código de acceso](#)

[Registro empleado](#)

[Soy administrador](#)

- **Login Administrador:**



### LOGIN ADMINISTRADOR

usuario admin \*

Contraseña \*


[INGRESAR](#) [REGRESAR](#)

- Login Empleado:




## LOGIN EMPLEADO

usuario \*



Contraseña \*

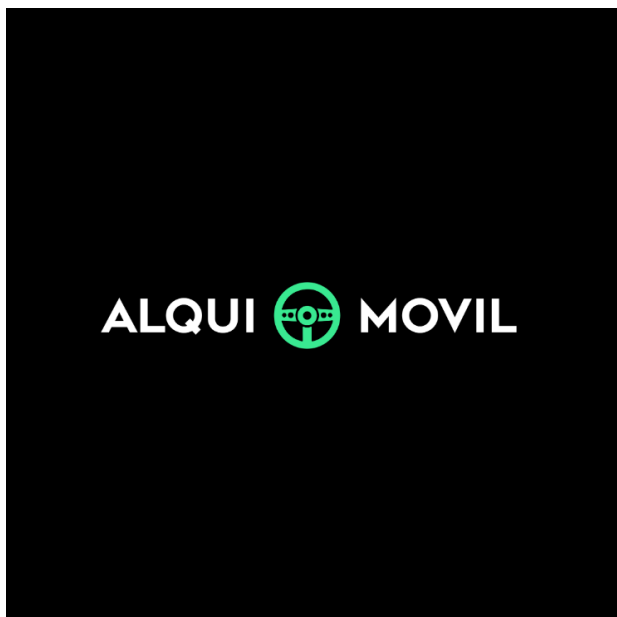


☐ código de acceso

**INGRESAR**

[Enviar código de acceso](#)

- Registro Empleado:



## REGISTRO EMPLEADO

Nombre \*

Apellido \*

Correo electrónico \*

Número de teléfono \*


Dirección \*

Usuario \*


Contraseña \*

**REGISTRARSE** **ATRAS**

- **Registro Cliente:**




### Hazte cliente ahora

Nombre *	Apellido *
Correo electrónico *	
Número de licencia *	Número de teléfono *
Dirección *	
Usuario *	Contraseña *
Fecha de nacimiento 	

REGISTRARSE

ATRAS

- **Enviar código de acceso:**



Correo electronico \*

@

ENVIAR CÓDIGO

ATRAS

- Solicitar Vehiculos (Alquiler Vehiculos):

LOGO

Inicio

Historial

Solicitudes de Renta

Cerrar Sesión

◀

Alquilar Vehiculo

CLK500 - Mercedes Benz

FOTO

Transmission: [Transmission]

Modelo: [Modelo]

Combustible: [Combustible]

Estado: [Estado]

Categoria: [Categoria]

Asientos: [Asientos]

Cuota por día: [Cuota]

Cuota:

**Q 100.00**

Fecha Inicio

Fecha Final

Alquilar

- Gestionar Cuotas:

LOGO

Inicio

Historial

Solicitudes de Renta

Cerrar Sesión

◀

Gestionar Cuota

CLK500 - Mercedes Benz

FOTO

Transmission: [Transmission]

Modelo: [Modelo]

Combustible: [Combustible]

Estado: [Estado]

Categoria: [Categoria]

Asientos: [Asientos]

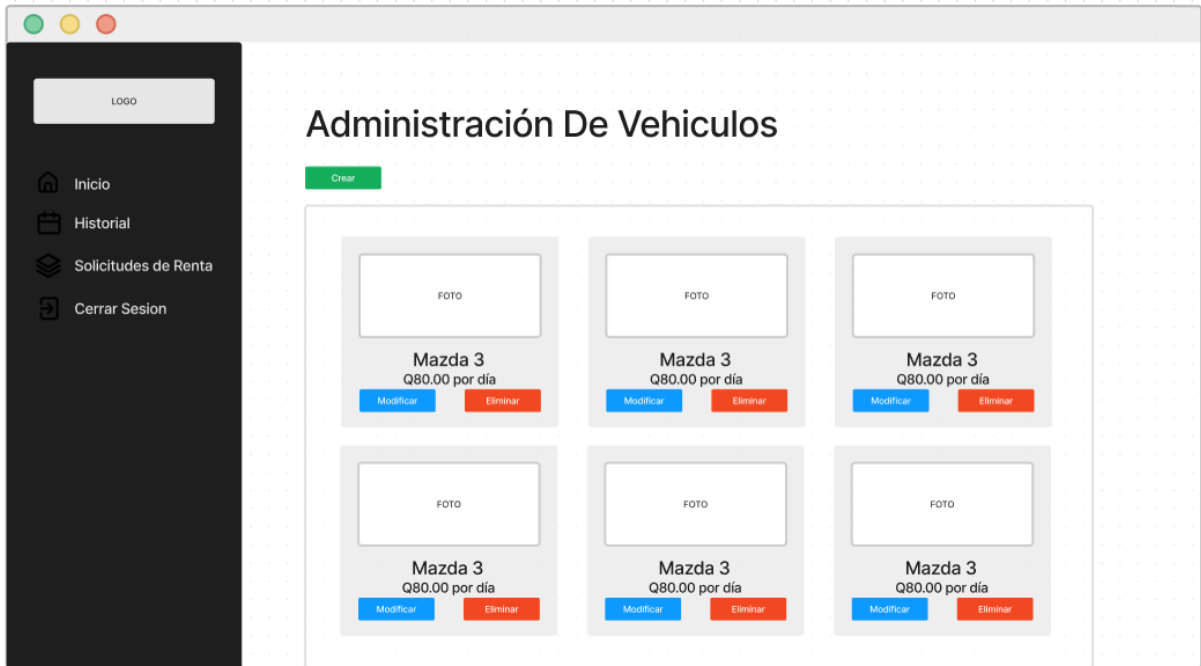
Cuota por día: [Cuota]

Cuota:

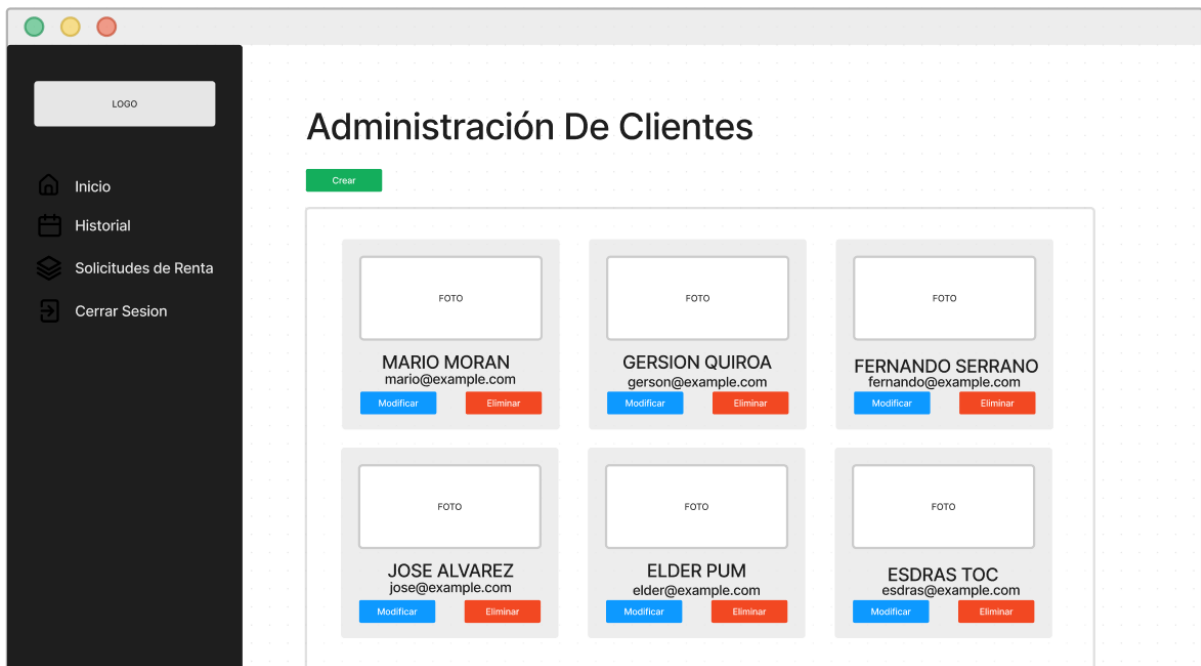
**Q 100.00**

Modificar

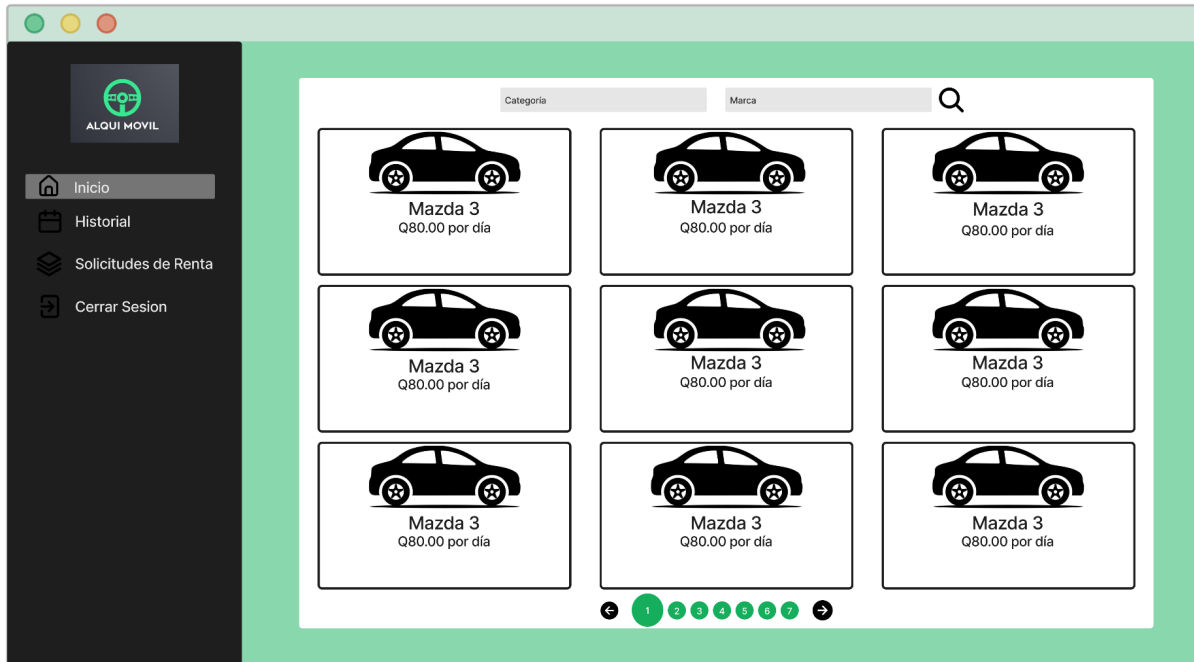
- **Administrar Vehículos (CRUD):**



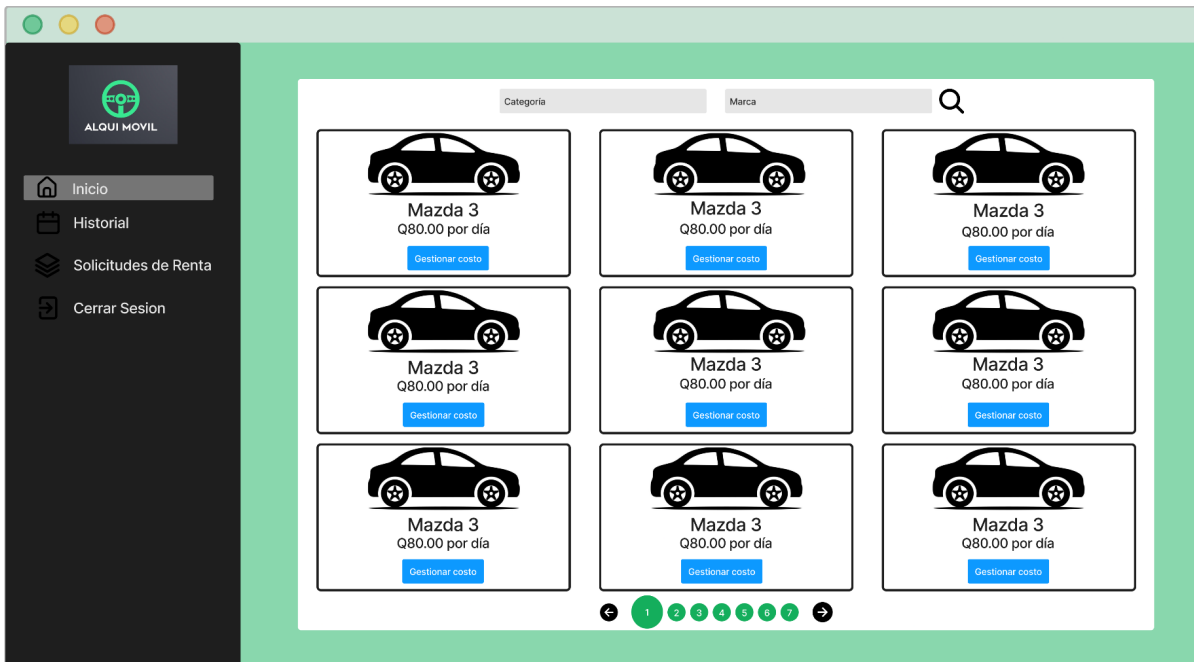
- **Administrar Clientes (CRUD):**



- **Inventario de Vehículos (Inicio de cliente):**



- **Inventario de Vehículos (Inicio de empleado o administrador):**



- **Pago alquiler vehiculo:**

LOGO

Inicio

Historial

Solicitudes de Renta

Cerrar Sesión

## ← Pago Alquiler

CLK500 - Mercedes Benz

Nombre de la Tarjeta Número de tarjeta

Fecha Vencimiento CVC

Pagar

Cuota: **Q 100.00**

- **Historial de alquileres de vehículos:**

## ← Historial de Alquiler

CLK500 - Mercedes Benz	Cuota: <b>Q 500.00</b>
2020 - Honda Civic	Cuota: <b>Q 200.00</b>
2023 - Ford Mustang	Cuota: <b>Q 5000.00</b>
2020 - BMW 3 SERIE	Cuota: <b>Q 2200.00</b>

- Devolver Vehículo:

LOGO

Inicio

Historial

Solicitudes de Renta

Cerrar Sesión

## ◀ Devolver Vehículo

CLK500 - Mercedes Benz

FOTO

Transmision: [Transmision]  
Modelo: [Modelo]  
Combustible: [Combustible]  
Estado: [Estado]  
Categoría: [Categoría]  
Asientos: [Asientos]  
Cuota por día: [Cuota]

Fecha Limite  
XX/XX/XXXX  
Devolver

- Método de pago:

LOGO

Inicio

Historial

Solicitudes de Renta

Cerrar Sesión

## Método de Pago

Nombre del titular:

Número de tarjeta:

Fecha de vencimiento:

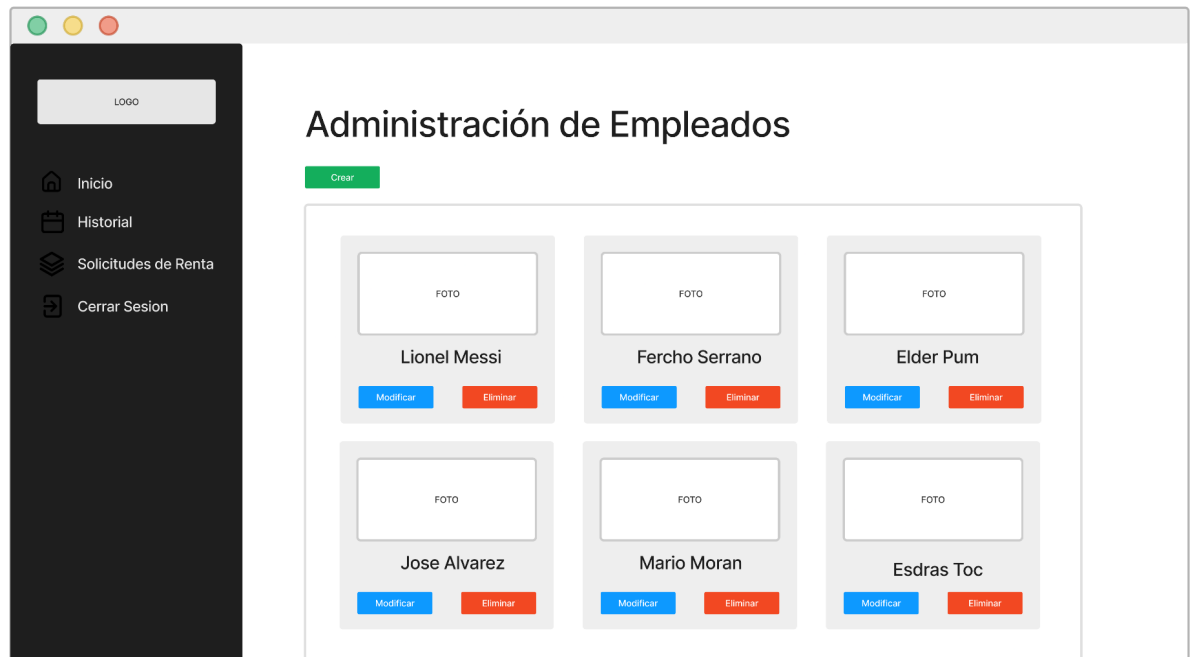
CVC (Código de seguridad):

VISA MasterCard PayPal

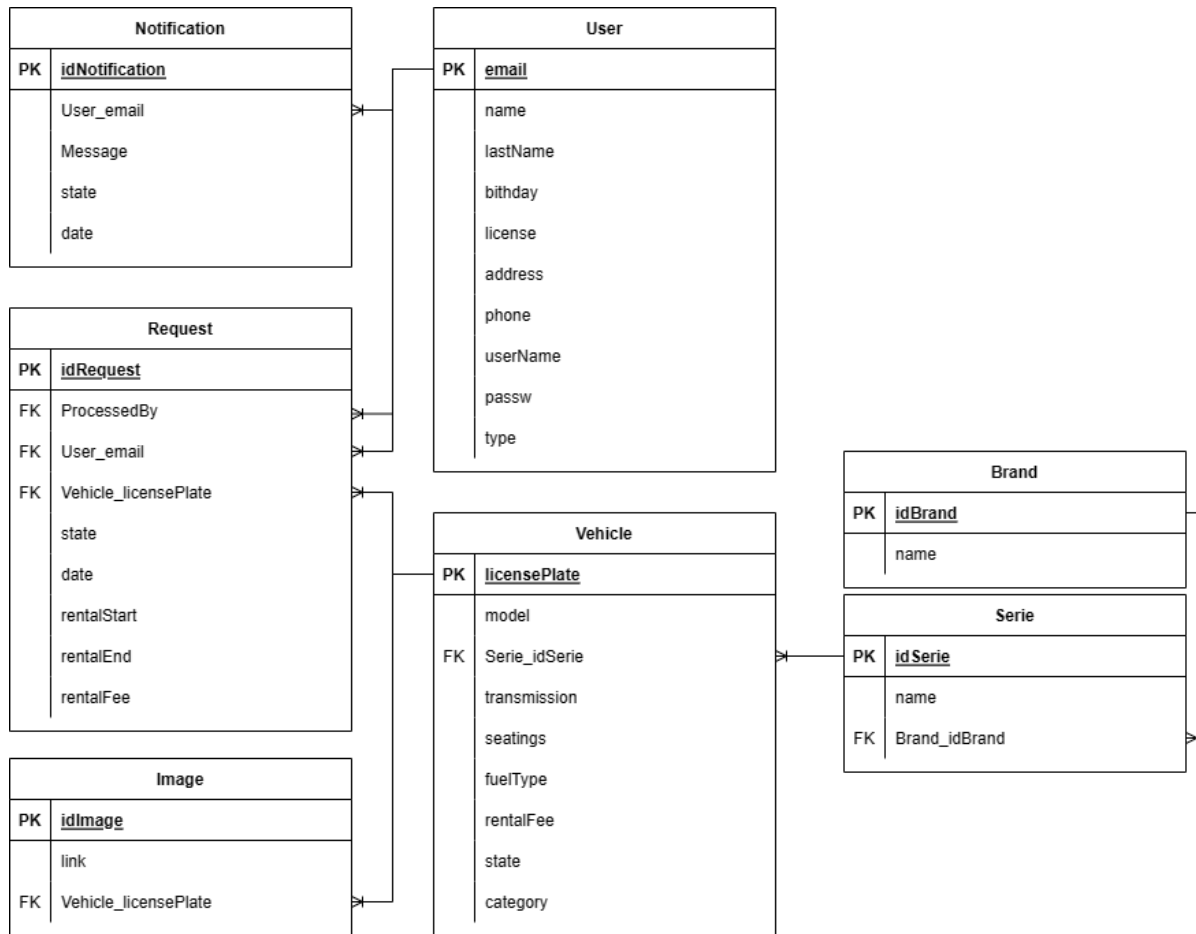
Cancelar Pagar



- **CRUD Empleados:**



# ENTIDAD RELACIÓN



## GLOSARIO

1. *Paginación*: La paginación es una secuencia de páginas que están conectadas y tienen un contenido similar. Es importante recordar que incluso cuando el contenido de una sección de una página web se divide en páginas diferentes, lo definiremos como paginación.
2. *Filtro*: es un tipo de búsqueda que permite restringir los resultados según criterios concretos, como el tipo de contenido o el periodo.
3. *CRUD*: CRUD es el acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar). Este concepto se utiliza para describir las cuatro operaciones básicas que pueden realizarse en la mayoría de las bases de datos y sistemas de gestión de información.
4. *Usuario*: Un usuario es aquella persona que utiliza un producto o servicio de forma habitual, beneficiándose de algún modo de dicha utilización, sin entrar a valorar la marca, el precio o las características técnicas de lo que utiliza.
5. *Encriptar*: proceso de transformar información legible o datos en un formato ilegible y seguro mediante el uso de algoritmos matemáticos y claves.
6. *Plataforma*: Software o aplicación web que permite gestionar el alquiler de vehículos.
7. *Endpoint*: se refiere a un punto final o URL específica que se utiliza para acceder a los servicios de una aplicación o API (Interfaz de Programación de Aplicaciones). Es básicamente un punto de conexión donde las aplicaciones pueden comunicarse entre sí.
8. *Navigate*: Es un hook proporcionado por la librería React Router para la gestión de la navegación en aplicaciones React. Este hook devuelve la función `navigate`, que se utiliza para cambiar programáticamente la ubicación de la aplicación.
9. *Tarjeta*: El instrumento de pago que el cliente utiliza para realizar el pago, que puede ser una tarjeta de débito o una tarjeta de crédito.
10. *CVC*: El Código de Verificación de Tarjeta, un número de seguridad de tres dígitos en la parte posterior de la tarjeta.
11. *Simulado*: Un proceso que replica una transacción de pago sin realizar un cargo real en la tarjeta del cliente.
12. *Bucket*: Se utiliza para referirse a un contenedor de datos o información, en sistemas de almacenamiento en la nube como Amazon, un bucket es un contenedor principal donde los datos se almacenan y organizan.
13. *Token*: Es una estructura de datos en formato JSON que se utiliza para transmitir información entre partes de manera segura.
14. *JSON*: Es un formato de intercambio de datos ligero y de fácil lectura, se utiliza para representar datos estructurados que son legibles tanto para humanos como para máquinas.
15. *API*: Es un conjunto de reglas y definiciones que permite que diferentes aplicaciones se comuniquen entre sí. Es un intermediario que permite que dos programas de software se hablen uno al otro.
16. *Frontend*: Se refiere a la parte de un sistema informático que interactúa directamente con los usuarios. Es la interfaz de usuario y todo lo que los usuarios ven y con lo que interactúan en una aplicación o sitio web.
17. *Aplicación Web*: Es un tipo de software que los usuarios pueden acceder y utilizar a través de un navegador web en lugar de tener que instalarlo en sus dispositivos.

## CASOS DE USO

- **Registro**

<b>Caso de Uso</b>	Registrarse
<b>Actores</b>	Empleado, Cliente
<b>Propósito</b>	Permitir el registro de nuevos clientes y empleados
<b>Tipo</b>	Principal
<b>Descripción:</b> Permite el registro de clientes y empleados dentro de la plataforma, obteniendo, de esta forma, acceso a las funcionalidades correspondientes para cada uno de los tipos de usuario.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"><li>1. El usuario ingresa al formulario de registro correspondiente (para empleado o cliente)</li><li>2. El usuario agrega cada uno de los datos solicitados en el formulario.</li><li>3. El usuario presiona el botón registrarse</li><li>4. Se muestra un mensaje de registro exitoso</li></ol>	
<b>Curso alternos:</b> <ol style="list-style-type: none"><li>4. Se muestra un mensaje de error por falta de formato correcto en alguno de los campos.</li><li>4. Se muestra un mensaje de error por la falta de alguno de los datos.</li><li>4. Se muestra un mensaje de error porque el usuario o correo electrónico ya se han utilizado con anterioridad.</li></ol>	

- **Inicio de sesión:**

<b>Caso de Uso</b>	Iniciar sesión
<b>Actores</b>	Administrador, Empleado, Cliente
<b>Propósito</b>	Permitir a los usuarios iniciar sesión en el sistema.
<b>Tipo</b>	Principal
<b>Descripción:</b> Permite a los usuarios de un sistema autenticarse y obtener acceso a las funciones del sistema correspondientes a su rol. El usuario ingresa su email y contraseña en el formulario de inicio de sesión. El sistema verifica el email y la contraseña en la base de datos. Si el email y la contraseña son correctos, el sistema autentica al usuario y muestra la interfaz de usuario correspondiente al rol del usuario..	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"><li>5. El usuario ingresa su email y la contraseña en el formulario de inicio de sesión.</li><li>6. El sistema verifica el email y la contraseña en la base de datos.</li><li>7. Si el nombre de usuario y contraseña son correctos, el sistema autentica al usuario.</li></ol>	

8. El sistema muestra la interfaz de usuario correspondiente al rol del usuario.

**Curso alternos:**

1. El email o contraseña son incorrectos: El sistema muestra un mensaje de error.
2. El usuario no tiene una cuenta: El sistema le permite al usuario crear una cuenta.

<b>Caso de Uso</b>	Iniciar sesión como administrador
<b>Actores</b>	Administrador
<b>Tipo</b>	Principal
<b>Descripción:</b> Permite al administrador iniciar sesión en el sistema y acceder a las funciones de administración.	

<b>Caso de Uso</b>	Iniciar sesión como empleado
<b>Actores</b>	Empleado
<b>Tipo</b>	Principal
<b>Descripción:</b> Permite al empleado iniciar sesión en el sistema y acceder a las funciones de empleado.	

<b>Caso de Uso</b>	Iniciar sesión como cliente
<b>Actores</b>	Cliente
<b>Tipo</b>	Principal
<b>Descripción:</b> Permite al cliente iniciar sesión en el sistema y acceder a las funciones de cliente.	

- **Crear Vehículos**

<b>Caso de Uso</b>	Creación de Vehículos
<b>Actores</b>	Administrador
<b>Propósito</b>	Agregar un nuevo vehículo al sistema
<b>Tipo</b>	Primario
<b>Descripción:</b> El administrador crea un nuevo registro de vehículo en el sistema. Se proporcionan los detalles del vehículo en el formulario. El sistema verifica la validez de los datos ingresados.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"> <li>1. El administrador inicia sesión en el sistema de administración de vehículos</li> <li>2. El administrador selecciona la opción "Crear Nuevo Vehículo"</li> <li>3. El administrador ingresa toda la información necesaria</li> <li>4. El sistema verifica la validez de los datos ingresados</li> <li>5. El sistema registra el vehículo en la base de datos</li> </ol>	

- **Actualizar Vehículo**

<b>Caso de Uso</b>	Actualizar Vehículo
<b>Actores</b>	Administrador
<b>Propósito</b>	Modificar la información de un vehículo existente en el sistema.
<b>Tipo</b>	Primario
<b>Descripción:</b> El administrador modifica los detalles de un vehículo existente mediante la placa del vehículo. El sistema verifica la validez de los datos actualizados.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"> <li>1. El administrador inicia sesión en el sistema de administración de vehículos</li> <li>2. El administrador selecciona el vehículo que desea actualizar</li> <li>3. El administrador modifica los detalles del vehículo</li> <li>4. El sistema verifica la validez de los datos actualizados</li> <li>5. El sistema guarda los cambios en la base de datos</li> </ol>	

- **Eliminar Vehículo**

<b>Caso de Uso</b>	Eliminar Vehículo
<b>Actores</b>	Administrador
<b>Propósito</b>	Eliminar un vehículo del sistema
<b>Tipo</b>	Primario
<b>Descripción:</b> El administrador elimina un vehículo del sistema. El sistema confirma la	

eliminación y elimina los datos del vehículo de la base de datos.

**Curso Normal de los eventos:**

1. El administrador inicia sesión en el sistema de administración de vehículos
2. El administrador selecciona la opción "Eliminar Vehículo"
3. El administrador elige el vehículo que desea eliminar
4. El sistema muestra una confirmación de eliminación
5. El administrador confirma la eliminación
6. El sistema elimina los datos del vehículo de la base de datos

• **Listado de Vehículos**

<b>Caso de Uso</b>	Listado de Vehículos
<b>Actores</b>	Administrador
<b>Propósito</b>	Ver la lista de todos los vehículos registrados en el sistema
<b>Tipo</b>	Primario
<b>Descripción:</b> El administrador accede a una lista que muestra todos los vehículos registrados en el sistema. Puede ver detalles como el número de placa, la imagen, etc.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"><li>1. El administrador inicia sesión en el sistema de administración de vehículos</li><li>2. El administrador selecciona la opción "Mostrar Listado de Vehículos"</li><li>3. El sistema muestra una lista de todos los vehículos registrados en el sistema, con detalles como el número de placa, cantidad de asientos, imagen, etc.</li></ol>	

• **Devolución Vehículos:**

<b>Caso de Uso</b>	Devolución de vehículo
<b>Actores</b>	Cliente
<b>Propósito</b>	Devolver un vehículo alquilado
<b>Tipo</b>	Primario
<b>Descripción:</b> El cliente devuelve un vehículo alquilado al finalizar el período de alquiler. El sistema verifica el estado del vehículo y calcula el importe a pagar. El cliente paga el alquiler y el sistema actualiza el estado del vehículo a disponible.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"><li>6. El cliente llega al lugar de devolución del vehículo.</li><li>7. El cliente verifica el estado del vehículo y confirma que está en buenas condiciones.</li><li>8. El cliente inicia sesión en la plataforma de alquiler de vehículos.</li><li>9. El cliente selecciona el vehículo que desea devolver.</li><li>10. El cliente proporciona los datos de su tarjeta de crédito o débito.</li></ol>	

11. El sistema verifica la validez de los datos de la tarjeta.
12. El sistema calcula el importe a pagar.
13. El sistema solicita al cliente que confirme el pago.
14. El sistema procesa el pago y actualiza el estado del vehículo a disponible.

**Curso alternos:**

- El vehículo está en malas condiciones: El cliente informa al sistema del estado del vehículo. El sistema genera un informe de daños y el cliente debe pagar una tarifa adicional.
- El cliente no tiene una tarjeta de crédito o débito: El cliente debe pagar en efectivo. El sistema imprime un recibo de pago.

<b>Caso de Uso</b>	Verificación del estado del vehículo
<b>Actores</b>	Cliente
<b>Propósito</b>	Verificar que el vehículo esté en buenas condiciones antes de devolverlo.
<b>Tipo</b>	Secundario
<b>Descripción:</b> El cliente inspecciona el exterior, interior y estado de los neumáticos, aceite y combustible del vehículo para verificar que esté en buenas condiciones. Si el vehículo está en malas condiciones, el cliente informa al sistema y debe pagar una tarifa adicional.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"> <li>1. El cliente inspecciona el exterior del vehículo en busca de daños.</li> <li>2. El cliente abre el vehículo e inspecciona el interior.</li> <li>3. El cliente comprueba el estado de los neumáticos, el aceite y el combustible.</li> </ol>	
<b>Curso alternos:</b> <ul style="list-style-type: none"> <li>• El vehículo está en malas condiciones: El cliente informa al sistema del estado del vehículo. El sistema genera un informe de daños y el cliente debe pagar una tarifa adicional.</li> </ul>	

<b>Caso de Uso</b>	Pago del alquiler
<b>Actores</b>	Cliente
<b>Propósito</b>	Pagar el importe correspondiente al alquiler del vehículo.
<b>Tipo</b>	Secundario
<b>Descripción:</b> El cliente proporciona los datos de su tarjeta de crédito o débito para pagar el alquiler. El sistema verifica la validez de los datos y calcula el importe a pagar. El cliente confirma el pago y el sistema procesa el pago y actualiza el estado del vehículo	



disponible.

**Curso Normal de los eventos:**

1. El cliente proporciona los datos de su tarjeta de crédito o débito.
2. El sistema verifica la validez de los datos de la tarjeta.
3. El sistema calcula el importe a pagar.
4. El sistema solicita al cliente que confirme el pago.
5. El sistema procesa el pago y actualiza el estado del vehículo a disponible.
6. El sistema procede a enviar un correo electrónico para confirmación de pago.

**Curso alternos:**

- El cliente no tiene una tarjeta de crédito o débito: El cliente debe pagar en efectivo. El sistema imprime un recibo de pago.

• **Inventario de vehículos:**

<b>Caso de Uso</b>	El usuario será capaz de visualizar los vehículos en inventario.
<b>Actores</b>	Administrador, Empleado, Cliente
<b>Propósito</b>	Permitir a los usuarios ver los un listado de los automóviles en inventario.
<b>Tipo</b>	Primario
<b>Descripción:</b> Este módulo permite a diferentes tipos de usuarios (clientes, empleados y administradores) acceder a la lista de los vehículos en inventario y sus diferentes operaciones.	

• **Detalles de vehículos:**

<b>Caso de Uso</b>	Alquiler de vehículo
<b>Actores</b>	Cliente / Administrador
<b>Tipo</b>	Secundario
<b>Descripción:</b> Permite a los actores poder visualizar los detalles de un automóvil en específico en el que están interesados en alquilar, así como un botón para solicitar el alquiler de dicho automóvil. Se mostrará así mismo la fecha de inicio y final del alquiler.	

<b>Caso de Uso</b>	Gestión del costo
<b>Actores</b>	Empleado / Administrador
<b>Tipo</b>	Secundario
<b>Descripción:</b> Permite a los actores seleccionar un vehículo y realizar cambios en las	

cuotas establecidas. Estos cambios deberán verse afectados y aplicados en el alquiler del vehículo inmediatamente.

- **Solicitud de renta:**

<b>Caso de Uso</b>	Aceptación de alquiler
<b>Actores</b>	Empleado
<b>Propósito</b>	Alquilar un vehículo y que el cliente proceda a pagar el alquiler.
<b>Tipo</b>	Primario
<b>Descripción:</b> El empleado tendrá la opción de aceptar la solicitud del alquiler de un cliente, mostrando el nombre del cliente que desea solicitar el vehículo. Se debe de enviar una notificación sobre la aceptación.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"><li>1. El empleado observará una lista de la solicitudes</li><li>2. Se centra en una solicitud y le da click al botón 'aceptar'</li><li>3. Ingresa el mensaje con la notificación que se le enviará al cliente.</li><li>4. Ingresa la fecha para la cual fue solicitado el vehículo.</li><li>5. Le da click al botón 'enviar'</li></ol>	
<b>Curso alternos:</b> Ninguno	

<b>Caso de Uso</b>	Declinación de alquiler
<b>Actores</b>	Empleado
<b>Propósito</b>	Niega el alquiler de un vehículo al cliente.
<b>Tipo</b>	Primario
<b>Descripción:</b> El empleado tendrá la opción de declinar/rechazar la solicitud del alquiler de un automóvil a un cliente, mostrando el nombre del cliente que desea solicitar el vehículo. Se debe de enviar una notificación sobre la razón del porqué no se aceptó la solicitud.	
<b>Curso Normal de los eventos:</b> <ol style="list-style-type: none"><li>6. El empleado observará una lista de la solicitudes</li><li>7. Se centra en una solicitud y le da click al botón 'declinar'</li><li>8. Ingresa el mensaje con la razón del porqué se le ha negado la solicitud.</li><li>9. Le da click al botón 'enviar'.</li></ol>	

<b>Curso alternos:</b> Ninguno
-----------------------------------

- **Historial de alquileres:**

<b>Caso de Uso</b>	Historial de alquileres
<b>Actores</b>	Administrador, Empleado, Cliente
<b>Propósito</b>	Permitir a los usuarios ver los un listado de los automóviles que han alquilado.
<b>Tipo</b>	Secundario
<b>Descripción:</b> Este módulo permite a diferentes tipos de usuarios (clientes, empleados y administradores) acceder a un resumen detallado de los alquileres realizados en el sistema.	

## HERRAMIENTA DE GESTIÓN DE PROYECTOS

<https://trello.com/invite/b/DmlicnW5/ATTI7759741a1d504273b6db479757737ada830F9DB0/ayd1p2fase1>

Link Grabaciones:

<https://drive.google.com/drive/folders/1uWBrDQF-GkohzGiHfPU0dW7VkK1g6CZd?usp=sharing>

# TOMA DE REQUERIMIENTOS

## **Registro:**

Requisitos funcionales:

- El sistema debe permitir a los usuarios registrarse
- El sistema debe registrar al nuevo usuario en la base de datos
- El sistema debe conectarse correctamente entre frontend, backend y base de datos.
- El sistema debe enviar correos de confirmación a los empleados

Requerimientos no funcionales:

- El sistema debe encriptar los datos importantes como la contraseña del usuario
- El sistema debe ser fácil de usar y accesible para todos los usuarios.
- El sistema debe asegurar la consistencia de los datos.
- El sistema debe notificar en caso de error o de éxito al usuario

## **Inicio de Sesión:**

Requisitos funcionales:

- El sistema debe permitir a los usuarios iniciar sesión con su email y contraseña.
- El sistema debe verificar la identidad del usuario en la base de datos.
- El sistema debe autenticar al usuario si el email y la contraseña son correctos.
- El sistema debe mostrar la interfaz de usuario correspondiente al rol del usuario.

Requerimientos no funcionales:

- El sistema debe ser seguro y proteger la información personal de los usuarios.
- El sistema debe ser fácil de usar y accesible para todos los usuarios.
- El sistema debe ser escalable para soportar un número creciente de usuarios.

## **Devolución Vehículos:**

Requerimientos Funcionales

- El sistema debe permitir al cliente verificar el estado del vehículo antes de devolverlo.
- El sistema debe calcular el importe a pagar por el alquiler del vehículo.
- El sistema debe permitir al cliente pagar el alquiler del vehículo con tarjeta de crédito o débito.
- El sistema debe actualizar el estado del vehículo a disponible una vez que se haya pagado el alquiler.

Requerimientos No Funcionales:

- El sistema debe utilizar cifrado para proteger los datos del cliente.
- El sistema debe poder procesar un gran número de transacciones de forma simultánea.
- El sistema debe tener una interfaz de usuario intuitiva y fácil de entender.

**Detalles de vehículo:**

## Requerimientos funcionales:

- El sistema debe mostrar el modelo del vehículo.
- El sistema debe mostrar la marca del vehículo.
- El sistema debe mostrar el tipo de transmisión del vehículo.
- El sistema debe mostrar la cantidad de asientos del vehículo.
- El sistema debe mostrar el tipo de combustible del vehículo (Gasolina, Diesel o Eléctrico).
- El sistema debe mostrar la cuota de alquiler por día del vehículo.
- El sistema debe mostrar un listado de fotos del vehículo, con al menos una foto disponible.
- El sistema debe mostrar el estado actual del vehículo (disponible, en alquiler, en mantenimiento, etc.).
- El sistema debe permitir cambiar el estado del vehículo (por ejemplo, de disponible a en alquiler) según sea necesario.
- El sistema debe mostrar la categoría del vehículo (Sedan, Bus, Camioneta, Pickup, Panel, Camión).
- El sistema debe permitir la búsqueda y filtrado de vehículos por categoría.
- El cliente debe poder seleccionar la fecha de inicio y fin del período de alquiler del vehículo.
- El sistema debe verificar la disponibilidad del vehículo para el período seleccionado.

## Requerimientos no funcionales:

- La interfaz de usuario debe ser intuitiva y fácil de usar.
- La carga de la información y las imágenes debe ser rápida para mejorar la experiencia del usuario.
- Los datos del vehículo y la información del cliente deben almacenarse de manera segura.
- Se debe implementar un sistema de autenticación seguro para garantizar que solo usuarios autorizados puedan acceder a los detalles del vehículo.
- El sistema debe ser capaz de manejar un número considerable de solicitudes de información de detalles de vehículos simultáneamente.
- Las consultas de disponibilidad para las fechas de alquiler deben procesarse de manera eficiente.
- El sistema debe ser fácil de mantener y actualizar para agregar nuevos modelos de vehículos o realizar modificaciones en los detalles existentes.
- El sistema debe ser escalable para manejar un aumento en el número de vehículos y usuarios sin comprometer el rendimiento.
- El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, para permitir a los clientes acceder a los detalles del vehículo en cualquier momento.

**Solicitudes de renta:**

## Requerimientos funcionales:

- El sistema debe mantener una lista de todas las solicitudes de reserva realizadas por los clientes.
- El empleado debe tener acceso a una interfaz que muestre la lista completa de solicitudes pendientes.

- El empleado debe tener la capacidad de revisar y aceptar las solicitudes de reserva.
- Al aceptar una solicitud, el sistema debe registrar la aceptación y actualizar el estado de la reserva.
- Debería ser posible aceptar solicitudes para fechas específicas.
- El sistema debe enviar automáticamente una notificación al cliente indicando que su solicitud de reserva ha sido aceptada.
- La notificación debe incluir detalles sobre la fecha y cualquier otra información relevante.
- El empleado debe tener la capacidad de rechazar las solicitudes de reserva.
- Al rechazar una solicitud, el sistema debe permitir al empleado ingresar una razón para el rechazo.
- Después de rechazar una solicitud, el sistema debe enviar automáticamente una notificación al cliente explicando la razón del rechazo.
- El sistema debe mantener un registro de todas las acciones realizadas por los empleados en relación con las solicitudes de reserva (aceptaciones, rechazos, comentarios, etc.).

#### Requerimientos no funcionales:

- La información de las solicitudes y las razones de rechazo deben almacenarse de manera segura.
- El acceso a la lista de solicitudes debe ser controlado mediante un sistema de autenticación y autorización.
- La interfaz para revisar y gestionar las solicitudes debe ser intuitiva y fácil de usar para los empleados.
- Las notificaciones enviadas a los clientes deben ser claras y comprensibles.
- El sistema debe estar disponible en todo momento para que los empleados puedan revisar y gestionar las solicitudes.
- Las notificaciones deben enviarse de manera oportuna después de aceptar o rechazar una solicitud.
- El sistema debe ser capaz de generar informes o registros para auditar las acciones realizadas por los empleados, incluyendo cambios de estado y razones de rechazo.
- El sistema debe ser escalable para manejar un crecimiento en el número de solicitudes y usuarios sin comprometer el rendimiento.
- El sistema debe ser fácil de mantener y actualizar para agregar nuevas funcionalidades o realizar mejoras en la gestión de reservas.

#### **Inventario de Vehículos:**

##### Funcionales:

- Los clientes podrán ver una lista de vehículos disponibles en la página web.
- Los vehículos se podrán buscar por marca y categoría.
- Los vehículos se podrán filtrar por marca y categoría.
- Se proporcionará paginación para evitar sobrecargar la vista de la lista de vehículos.
- Los clientes podrán navegar a través de diferentes páginas de vehículos disponibles.
- El administrador y los empleados podrán visualizar la lista de vehículos teniendo estados disponibles, ocupados o reservados.
- En caso de vehículos ocupados y reservados, se mostrará el nombre del cliente que solicitó el vehículo.

No Funcionales:

- La página web debe cargar y mostrar el inventario de vehículos de manera rápida y eficiente, incluso cuando hay una gran cantidad de vehículos.
- La información sensible, como los nombres de los clientes, estará protegida y solo accesible para administradores y empleados.
- La interfaz de usuario debe ser intuitiva y fácil de usar, permitiendo a los clientes encontrar rápidamente los vehículos que desean alquilar.
- La página web debe estar disponible de manera continua para que los clientes puedan acceder al inventario de vehículos en cualquier momento.



## **Administración de Vehículos (CRUD):**

### **Requerimientos Funcionales:**

- **Agregar Vehículo:** El administrador debe poder agregar nuevos vehículos al inventario con la siguiente información:
  - Modelo
  - Marca
  - Transmisión
  - Cantidad de asientos
  - Tipo de combustible (Gasolina o diesel o eléctrico)
  - Cuota de alquiler por día.
  - Listado de fotos del vehículo (al menos una foto por vehículo)
  - Estado, por defecto este estará como disponible.
  - Categoría (Sedan, Bus, Camioneta, Pickup, panel, camion)
- El administrador debe poder ver una lista de todos los vehículos existentes en el inventario, incluyendo detalles como marca, modelo, categoría y estado.
- El administrador debe poder editar la información de un vehículo existente, incluyendo la marca, modelo, categoría, estado de disponibilidad y detalles adicionales.
- El administrador debe poder eliminar un vehículo del inventario.

### **Requerimientos No Funcionales:**

- Se debe implementar un sistema de autenticación y autorización sólido para garantizar que solo los administradores tengan acceso a la función CRUD de vehículos.
- Se deben aplicar medidas de seguridad para proteger la información de los vehículos y prevenir accesos no autorizados.
- Se deben implementar controles de validación de datos para asegurarse de que la información ingresada sea coherente y esté en el formato correcto.
- La función CRUD debe estar disponible de manera continua para que los administradores puedan realizar las operaciones en cualquier momento.

## **Pago del Alquiler del Vehículo:**

### **Requerimientos Funcionales:**

- **Registro de Detalles de Pago:** El sistema debe ser capaz de registrar y almacenar de forma segura los datos de pago proporcionados por el usuario, incluyendo el nombre que aparece en la tarjeta, número de tarjeta, fecha de vencimiento y CVC.
- **Validación de Datos de Tarjeta:** El sistema debe verificar la validez de los datos de la tarjeta de crédito o débito, incluyendo la longitud correcta del número de tarjeta, la fecha de vencimiento futura y la autenticidad del CVC.
- **Generación de Confirmación de Pago:** Después de que se hayan proporcionado y validado los datos de pago, el sistema debe generar una confirmación de pago que incluya los detalles de la transacción, como el monto pagado, la fecha y la hora de la transacción, y un número de referencia único.

#### Requerimientos No Funcionales:

- Seguridad de los Datos: Los datos de pago deben estar encriptados durante la transmisión y almacenamiento..
- Tiempo de Respuesta del Sistema: El sistema debe procesar las transacciones de pago de manera eficiente y oportuna, con un tiempo de respuesta mínimo para que los usuarios no experimenten retrasos significativos durante el proceso de pago.
- Disponibilidad y Tolerancia a Fallos: El sistema de pago debe ser capaz de manejar situaciones de fallos, como caídas temporales del servidor, sin pérdida de datos o interrupciones significativas en el servicio.

#### Historial de Alquiler del Vehículo:

##### Requerimientos Funcionales:

- El sistema debe permitir a los clientes ver un resumen de los alquileres que han realizado. Sólo deben mostrarse los alquileres asociados a la cuenta del cliente que está iniciando sesión.
- El sistema debe permitir a los empleados ver un resumen de todos los alquileres realizados, independientemente del cliente que los haya efectuado.
- El sistema debe permitir a los administradores ver un resumen de todos los alquileres realizados, incluyendo información sobre el empleado que aprobó o rechazó el alquiler, junto con la razón del rechazo si aplica.

##### Requerimientos No Funcionales:

- Los datos de los alquileres y la información del cliente deben estar protegidos mediante técnicas de encriptación. Además, el acceso a los resúmenes de alquiler debe estar restringido según los roles (cliente, empleado, administrador) para evitar accesos no autorizados.
- El sistema debe ser capaz de manejar grandes volúmenes de datos y proporcionar resúmenes de alquiler de forma rápida y eficiente, incluso durante períodos de alta demanda.