

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA RENTA DE
VEHÍCULOS**

FRONTEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

MATEO SEBASTIAN CONGO FOLLECO

DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO

DMQ, MARZO 2024

CERTIFICACIONES

Yo, **Congo Folleco Mateo Sebastian** declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Congo Folleco Mateo Sebastian

mateo.congo@epn.edu.ec

aidenperse@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por **Congo Folleco Mateo Sebastian**, bajo mi supervisión.

ING. JUAN PABLO ZALDUMBIDE PROAÑO

juan.zaldumbide@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Congo Folleco Mateo Sebastian

DEDICATORIA

Dedico este proyecto a mi abuela, a mi madre y a mis tías, todas ellas son para mí personas influyentes en mi desarrollo académico, ya que siempre han estado apoyándome a perseguir mis sueños, mis anhelos y mis metas por cumplir, eh podido llegar aquí gracias a su constante esfuerzo, coraje y sacrificio. Siendo honesto no creo que todas las palabras del mundo me fueran a alcanzaran para demostrar lo agradecido que estoy por todo esto que me han dado, tuve la suerte de nacer en una gran familia, una familia que está dispuesta a apoyarse mutuamente y darse la mano en tiempos de crisis, una familia que me ha demostrado que con esfuerzo y disciplina se puede alcanzar la cima de la montaña, sin importar el tiempo u adversidad, donde solo importa tu objetivo y las ganas que tengas de llegar a el, donde cada día que te levantas para ser la mejor versión de ti, dispuesta a enseñar tus aciertos y dispuesta a aprender de tus errores.

AGRADECIMIENTO

Quiero agradecer especialmente a mi madre Pilar Folleco, que con esfuerzo llevo a darme el estudio, a mi tía Elena Padilla que me apoyo durante todo este proceso que es la universidad, a mi tía Rosa Padilla que desde tengo uso de razón a estado muy presente en mi vida, dándome un apoyo incondicional para alcanzar mis metas, a toda mi familia que me dieron la motivación para subir cada peldaño, enseñándome que se debe luchar por lo que quieres hasta alcanzarlo y finalmente agradecerme a mí por darme esta oportunidad, por fijarme un camino en la vida y no descansar hasta llegar a el, para mi este no es el fin del camino, sé que puedo llegar mucho más lejos y más importante que eso, quiero llegar más allá, quiero ver de qué puedo ser capaz de alcanzar con mi ambición y mi enfoque por ser la mejor versión de mí mismo.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico.....	3
2 METODOLOGÍA	4
2.1 Metodología de Desarrollo	4
2.1.1 Roles.....	6
2.1.2 Artefactos	6
2.2 Diseño de interfaces.....	10
2.3 Diseño de la arquitectura	11
2.4 Herramientas de desarrollo	12
3 RESULTADOS.....	13
3.1 Sprint 0.....	13
3.2 Sprint 1.....	15
3.3 Sprint 2.....	18
3.4 Sprint 3.....	26
3.5 Sprint 4.....	30
3.6 Sprint 5.....	32
4 CONCLUSIONES.....	36
5 RECOMENDACIONES	37
6 REFERENCIAS BIBLIOGRÁFICAS	38
7 ANEXOS	41
ANEXO I	42
ANEXO II	43

ANEXO III	71
ANEXO IV.....	72

RESUMEN

El *frontend* a desarrollar está enfocado en satisfacer las necesidades de una empresa de renta de vehículos, en donde existen 3 tipos de usuarios por un lado se tiene al usuario propietario, este puede registrar distintos números de vehículos que estén a su disposición para que puedan ser rentados por los diferentes clientes de la plataforma, también se cuenta con el administrador, este puede ver todas las solicitudes hechas por los propietarios y aprobarlas o rechazarlas según sea el caso, por último se cuenta con el usuario cliente, este es el que puede rentar los vehículos por una cantidad de días de su elección, a él se le muestran solo los vehículos aprobados por el administrador. Para acceder a este servicio tanto el usuario propietario como el usuario cliente tienen la posibilidad de registrarse en la página web a través de su correo electrónico, el administrador no necesita un registro ya que esta es una cuenta que es habilitada directamente por el backend. Independientemente de los roles todos los usuarios cuentan con la opción de recuperar su contraseña en caso de ser olvidada y de igual forma se les proporciona a todos un apartado para que puedan actualizar sus perfiles y sus respectivos campos como lo son: nombre, apellido, dirección y celular, el correo y la cédula no son actualizables ya que estos son datos únicos. Todo este componente se encuentra realizado en *react* una librería de *javascript*, proporcionando una interfaz intuitiva para navegar donde se aplican los principios de la experiencia de usuario.

PALABRAS CLAVE: propietario, administrador, cliente, *react*, renta, vehículos

ABSTRACT

The frontend to be developed is focused on meeting the needs of a car rental company, where there are 3 types of users on the one hand we have the owner user, this can register different numbers of vehicles that are available to be rented by different customers of the platform, There is also the administrator, who can see all the requests made by the owners and approve or reject them as the case may be. Finally, there is the client user, who can rent the vehicles for a number of days of his choice, only the vehicles approved by the administrator are shown to him. To access this service both the owner user and the client user have the possibility of registering on the web page through their email, the administrator does not need to register since this is an account that is enabled directly by the backend. Regardless of the roles, all users have the option to recover their passwords in case they are forgotten, and they are also provided with a section where they can update their profiles and their respective fields such as: name, surname, address and cell phone number, the email and ID are not updateable since these are unique data. All this component is made in react a javascript library, providing an intuitive interface to navigate where they apply the information in their profiles.

KEYWORDS: owner, manager, customer, react, challenge, vehicles.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Este proyecto está dirigido a la renta de vehículos a través de una aplicación web. Esta iniciativa tiene como finalidad brindar un servicio de renta de vehículos de manera eficiente y accesible. La implementación de esta permite a los clientes de CarGod acceder de forma fácil y rápida a una amplia gama de vehículos disponibles para su alquiler [1]. A través de la plataforma, los usuarios pueden seleccionar el tipo de vehículo que deseen y poder elegir en los días que disponen del vehículo [2]. La aplicación web también ofrece a los clientes la posibilidad de personalizar su experiencia de alquiler, brindando información detallada sobre cada vehículo, como características generales, capacidad de pasajeros, propietario del vehículo, marca del vehículo y costo del alquiler por día, lo que permite tomar decisiones específicas para las necesidades de cada usuario. Uno de los beneficios clave de esta aplicación web es que simplifica y agiliza el proceso de contratación de vehículos para los clientes de CarGod [3].

Como primer lugar se observa que la falta de accesibilidad las 24 horas del día, los 7 días de la semana, limita la capacidad de los clientes para interactuar con la empresa según sus necesidades, lo que llega a afectar negativamente la satisfacción de sus usuarios y su fidelidad. Además, sin un sitio web disponible la empresa queda rezagada en términos de competencia con las demás entidades, en un mundo tan moderno como lo es hoy en día entrar al mundo digital representa un mejor servicio y ganancia. La falta de presencia en la web puede dificultar la captación de nuevos clientes y llegar a ser una pérdida de oportunidades en ventas frente a competidores más adaptados a las necesidades del mercado actual [4].

La solución propuesta para esta problemática es desarrollar una plataforma en línea para alquiler de vehículos en la que su servicio permanezca accesible en todo momento sin interrupción. Esta página web permite a las empresas de alquiler de vehículos ampliar su alcance de clientes y ofrecer servicios de una manera más accesible y conveniente. El componente que permite realizar esta solución es el *frontend*, que está basado en lenguajes de programación HTML, CSS y JavaScript. La implementación de este componente facilita el registro de los clientes y su acceso eficiente a los servicios de alquiler, mejorando la experiencia y promoviendo el crecimiento de las grandes empresas [5].

1.1 Objetivo general

Desarrollar un *frontend* que permita a los usuarios de la empresa CarGod gestionar la renta de vehículos

1.2 Objetivos específicos

OE1. Determinar los requerimientos de la aplicación.

OE2. Diseñar la interfaz de usuario.

OE3. Implementar el *frontend* de la aplicación.

OE4. Comprobar el funcionamiento del *frontend*.

OE5. Desplegar el componente en un ambiente de producción.

1.3 Alcance

El componente actual se centra en crear una interfaz interactiva y fácil de usar, con el propósito de facilitar la gestión del alquiler de vehículos para la empresa CarGod. Esta interfaz busca ofrecer a los usuarios una experiencia sin complicaciones y eficiente, permitiendo a los clientes realizar reservas, seguimiento de vehículos disponibles de una forma sencilla y amigable.

Este proyecto es desarrollado con base en la metodología *scrum* [6], esto con el objetivo de permitir un desarrollo rápido y eficaz, junto con una distribución organizada de las tareas y su proceso correspondiente. Para diseñar las interfaces y los prototipos se emplea la herramienta de gráficos vectoriales Figma [7]. Para codificar y desarrollar dichas interfaces con sus funcionalidades, se hace uso del *framework React* [8], conocido por ser una biblioteca dinámica y eficiente que facilita la programación de componentes de una manera simple y estructurada. Permitiendo al usuario propietario registrar un vehículo, al usuario administrador aceptar o rechazar los registros hechos por los propietarios y finalmente el usuario cliente donde este puede ver todas solicitudes aceptadas por el administrador y escoger el vehículo que mejor se adapte a sus necesidades.

1.4 Marco Teórico

Frontend

El *frontend* le permite al usuario interactuar dentro de la página web, es decir, que de primera mano va a familiarizarse con la interfaz sus estilos y animaciones, esto gracias a varios lenguajes como lo son HTML, CSS y JavaScript [9]. Todo esto se realiza con el fin de garantizar que los usuarios que visiten la página tengan una experiencia satisfactoria encuentren su uso sencillo.

Metodología ágil

Dentro de las empresas se necesita una manera eficiente donde se pueda entregar un producto de alta calidad con costos y tiempos reducidos con esta idea en mente es donde entran las metodologías ágiles, estas son las que permiten adaptar la forma de trabajo a las condiciones del proyecto consiguiendo flexibilidad y rapidez a la hora de realizar un proyecto [10]. Una de las metodologías más utilizadas es *scrum* que ha permitido llevar el desarrollo de cada tarea con un *sprint* esto da la pauta para definir una tarea y el tiempo en que se realiza.

Interfaz de usuario

Se puede definir a la interfaz de usuario como el medio a través del cual el cliente interactúa con los distintos dispositivos tecnológicos, aquí es donde se producen todos los puntos de contacto entre el equipo y la persona, en términos simples se puede decir que uno transmite lo que quiere realizar dentro del dispositivo y esta contesta dicha acción [11]. Figma es una herramienta utilizada para el diseño de experiencias de usuario (UX) e interfaces de usuario (UI), ya que esta plataforma es colaborativa y *online*, gracias a esto varios miembros del equipo pueden estar interactuar con el proyecto al mismo tiempo [7]. Adicionalmente se desarrolla esta página web con lenguajes de programación tales como *html*, que nos sirve para estructurar nuestra página, *css* que permite darle estilo y formato al contenido *html* y finalmente *javascript* que hace el contenido de nuestra página interactivo.

2 METODOLOGÍA

Se ha identificado un problema para las grandes empresas o empresas en ascenso, para el cual, mediante una hipótesis, se accede a una resolución exitosa o insatisfactoria utilizando la metodología cualitativa. Esto proporciona un criterio preciso a los participantes de esta investigación frente a la problemática

Para desarrollar este componente se plantea utilizar el diseño de investigación-acción, centrado en encontrar soluciones prácticas realistas a las diferentes dificultades y problemas de las distintas empresas u organizaciones. En primer lugar, se identifica el problema, dependiendo del tipo de problema se buscan diferentes acciones para solventarlo, después se miden los resultados obtenidos para identificar si los resultados obtenidos llegan a ser realmente satisfactorios o se necesita replicar el proceso [12].

2.1 Metodología de Desarrollo

Para la resolución del desarrollo de *software* se tiene vigentes una alta variedad de metodologías ágiles estas permiten al equipo llevar el desarrollo de una manera eficiente controlando la calidad del producto. La metodología ágil planteada para esta investigación es *scrum* ya que permite abordar problemas de una manera efectiva para la entrega de productos eficientes al mercado, proporcionando un plan de valores, roles y pautas para el apoyo del equipo de desarrollo permitiéndoles la mejora continua de grandes proyectos [6].

Fases de la metodología scrum

Para el uso de la metodología *scrum* existen distintas fases de desarrollo que se observan en la **Tabla 2.1** [13].

Tabla 2.1: Fases de la metodología scrum

Inicio	En esta se estudia y analiza el proyecto para identificar las necesidades básicas del <i>sprint</i> , en este caso el flujo para rentar un vehículo
Planificación y estimación	La planificación y estimación ayuda a establecer metas fijas y a cumplir con los distintos tiempos asignados. Para el proyecto se realiza distintas reuniones para realizar el proceso.
Implementación	En esta fase se analiza cómo se puede optimizar el trabajo de cada grupo para darle forma definitiva al proyecto. Se realiza una investigación referente a las tecnologías útiles para una página web.
Revisión y retrospectiva	Una vez que todo haya sido maquetado e implementado, se lleva a cabo una revisión del proceso que consiste en una evaluación del grupo sobre el resultado obtenido. Para el proyecto se realizaron pruebas de aceptación, compatibilidad y accesibilidad donde nos proporciona información útil del contenido.
Lanzamiento	Esta es la fase en la que se despliega el producto a una plataforma en concreto. La página del proyecto se despliega en la plataforma netlify.

2.1.1 Roles

Dentro de la metodología *scrum* se cuenta con distintos roles como lo llegan a ser:

Product Owner

Él tiene la responsabilidad de decidir el cómo se va a llevar el proyecto, gestionando prioridades, debido a que este es el dueño del producto [14].

Scrum Master

Es responsable de ayudar al equipo con la organización y usar la metodología *scrum* de la manera más conveniente, se lo considera el líder del proyecto [14].

Team

Recalcando el *team* es el equipo de desarrollo son los profesionales que llevan a cabo el desarrollo del producto de manera conjunta trayendo a la vida cada una de las historias que se propone en cada sprint [14].

La asignación de funciones para este proyecto se detalla en la **Tabla 2.2**

Tabla 2.2: Roles

Roles	Nombres
Product Owner	Mateo Congo
Scrum Master	Ing. Juan Pablo Zaldumbide Proaño
Team (Programador)	Mateo Congo

2.1.2 Artefactos

Los artefactos son elementos tangibles que se crean, modifican o utilizamos durante el proceso de desarrollo para documentar, respaldar y facilitar la colaboración en el trabajo. Estos artefactos pueden incluir documentos, diagramas, modelos, código fuente, entre otros.

Product Backlog

Esta es una lista ordenada gestionada por el *product owner* donde existen todos los requerimientos a implementarse en el producto priorizando los temas que tengan más peso en aquel momento [14].

Sprint Backlog

Este es un plan detallado de los distintos *sprint* a desarrollar es gestionado por el equipo de desarrollo, se puede visualizar todo el trabajo pendiente en un *sprint*, así se llegan a identificar las acciones faltantes para la culminación del desarrollo [14].

Recopilación de Requerimientos

Gracias a una exhaustiva investigación sobre las necesidades de las empresas en crecimiento o las grandes empresas que aún no cuenta con un servicio en línea referente a la rentabilidad de vehículos se a logrado obtener las necesidades esenciales para este rubro, esto ayuda a definir el alcance del proyecto en cuanto a la creación y diseño del *frontend*.

Historias de Usuario

Las historias de usuario son componentes esenciales en el desarrollo de las metodologías ágiles, estas proporcionan información crucial al equipo de desarrollo colocando a los usuarios finales en el foco principal del proceso y se puede llegar a comprender porque se está haciendo la compilación de ciertos requerimientos. El propósito principal de las historias de usuario es proporcionar una función específica del *software* al cliente, es decir, es el objetivo final expresado desde la perspectiva del usuario, es dar un vistazo al resultado deseado [15].

Las historias de usuario son presentadas en el formato mostrado en la **Tabla 2.3** y todas estas están disponibles en el **ANEXO II**.

Tabla 2.3: Historias de usuarios - página de inicio

HISTORIA DE USUARIO	
Identificador (ID): HU000	Usuario: Propietario
Nombre de la historia: Mostrar página de inicio	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Sprint asignado: 0	
Responsable: Mateo Congo	
Descripción: Como usuario, quiero poder ver la página principal de la empresa de alquiler de vehículos	
Observaciones: La página principal deberá ser observada por todos los usuarios que lleguen a entrar al sitio web.	

2.1.3 Sprints

Los *sprints* son períodos de tiempo fijos en los que el equipo *scrum* realiza una tarea establecida para el desarrollo del *software*, los *sprints* están presentes de una manera esencial en el desarrollo de las metodologías ágiles. La idea principal de un *sprint* es dividir proyectos grandes y complejos en proporciones minúsculas para la gestión de tareas [16]. La lista de *sprints* se puede observar en la **Tabla 2.4**.

Tabla 2.4: Sprints para la aplicación web

Sprint	Descripción	Historia de usuario
0	<ul style="list-style-type: none"> • Configuración del entorno de desarrollo. • Diseño y prototipado. • Codificación de la página de inicio. 	<ul style="list-style-type: none"> • HU000
1	<ul style="list-style-type: none"> • Formularios de inicio de sesión tanto como de los clientes como los propietarios y administrador • Control de inicio de sesión utilizando información de un punto final de datos o también conocido como <i>endpoint</i>. 	<ul style="list-style-type: none"> • HU001 • HU002 • HU003 • HU004 • HU005
2	<ul style="list-style-type: none"> • Recuperación de contraseña • Registro vehículo propietario • Aprobación de vehículos por parte del administrador • Visualización de los vehículos registrados por parte de los propietarios 	<ul style="list-style-type: none"> • HU006 • HU007 • HU008 • HU009
3	<ul style="list-style-type: none"> • Contratación de un vehículo registrado • Cantidad de días que puede rentar un vehículo 	<ul style="list-style-type: none"> • HU010 • HU011
4	<ul style="list-style-type: none"> • Visualización del historial de pedidos 	<ul style="list-style-type: none"> • HU012
5	<ul style="list-style-type: none"> • Pruebas y despliegue 	<ul style="list-style-type: none"> • HU013

2.2 Diseño de interfaces

El diseño de interfaz es una idea para controlar mejor la interacción entre la persona y su dispositivo, generalmente un computador o un teléfono inteligente, la interfaz permite al usuario controlar efectivamente acciones de la máquina y responder dependiendo de la acción para corroborar el uso correcto de la interacción. Tomando en cuenta estas consideraciones el diseñador de la interfaz es el responsable de que el proceso entre el cliente y el sitio web sea de una manera fácil e intuitiva, para que el usuario pueda acceder a la información o ejecutar acciones de acuerdo con sus peticiones [17].

Herramienta utilizada para el diseño

Para la creación de mockups existen diversos tipos de herramientas, pero una de las más eficientes y principales es *figma*, esta herramienta resulta sumamente beneficiosa para la elaboración de prototipos web y al momento de la edición de gráficos vectoriales, en la **Figura 2.1** se muestra un ejemplo del diseño para este proyecto mientras que los prototipos adicionales se detallan en el **ANEXO II**.

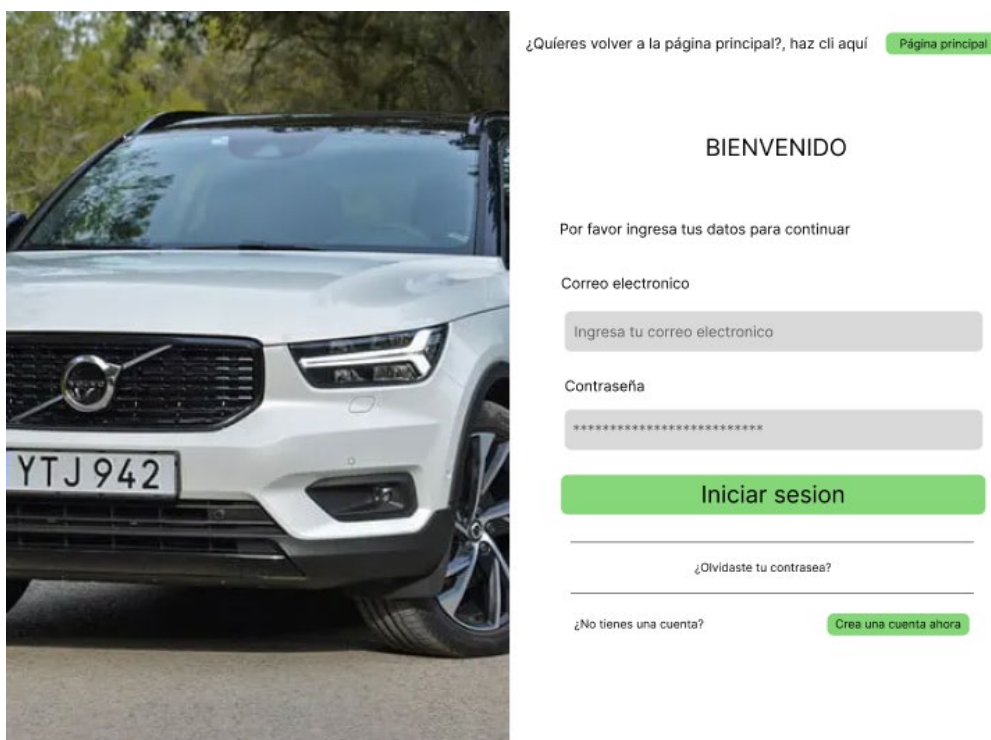


Figura 2.1: Login

2.3 Diseño de la arquitectura

Tener un diseño de la arquitectura de software que se ha utilizado para el proyecto es una parte esencial para su creación ya que permite tener un pensamiento claro de cómo está estructurado en caso de querer agregarse más detalles, esto permite que el equipo pueda también visualizar todo el panorama del componente a desarrollar [18].

Patrón arquitectónico

En el lado del cliente, la navegación es gestionada mediante *react router dom*, facilitando la creación de rutas dinámicas y optimizando la experiencia del usuario. Los *hooks* desempeñan un papel importante en la aplicación, encapsulando la lógica y asegurando un flujo de datos consistente. La comunicación con el *backend* se realiza a través de json, estableciendo un intercambio de datos entre el *frontend* y el servidor. Al enviar los datos se reciben *endpoints* dinámicamente generados que se utilizan para mantener la sincronización entre las dos capas de la aplicación, el diseño de esto lo podemos ver en la **Figura 2.3**.

El despliegue se realiza en la plataforma netlify, este simplifica el proceso de implementación continua, la integración automática asegura que las actualizaciones se reflejen rápidamente en el entorno de producción.

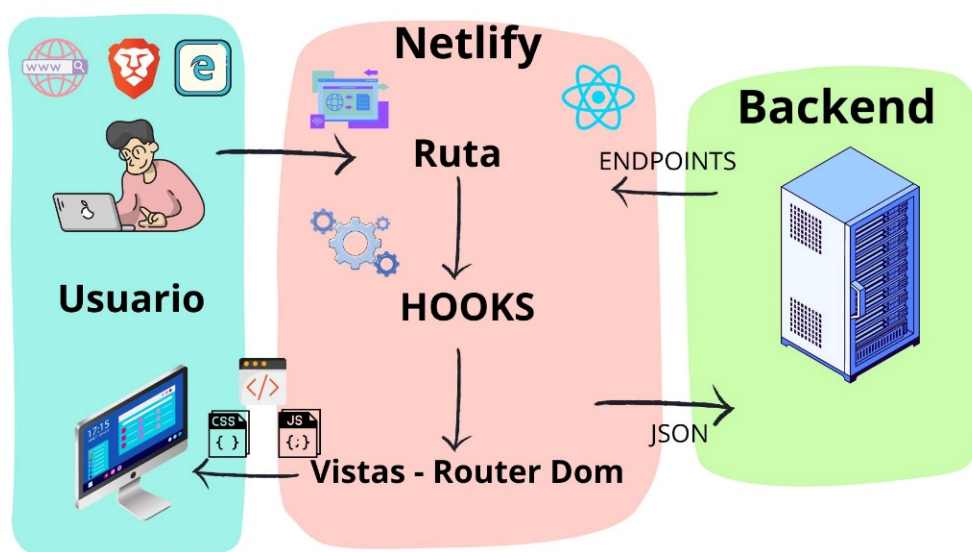


Figura 2.2: Arquitectura

2.4 Herramientas de desarrollo

En el progreso de la creación de sitios web se han implementado formas y herramientas que permiten simplificar el desarrollo de software, en la **Tabla 2.5** se pueden observar las herramientas de todo este proyecto.

Tabla 2.5: Herramientas de desarrollo

Herramienta	Justificación
React	Con <i>react</i> , es posible desarrollar componentes reutilizables que facilitan y aceleran el proceso de desarrollo de <i>software</i> .
Node.js	Este entorno permite la comunicación en tiempo real, es decir fuera de un servicio web y resulta útil para las aplicaciones que requieren diferentes tipos de datos de manera inmediata.
Git y Github	<i>Git</i> es una herramienta que ayuda a controlar el versionado de código lo cual permite mantener el control sobre el proyecto y por su parte <i>github</i> permite tener este versionamiento en la nube.
Visual Studio Code	Un entorno de desarrollo flexible para la programación en múltiples lenguajes.

3 RESULTADOS

En este apartado podremos ver todos los sprints que componen el proyecto, donde se visualiza el diseño de la página con sus respectivas secciones de código donde se muestra la lógica implementada para el consumo de las respectivas *api's*.

3.1 Sprint 0.

Como se puede ver en la **Tabla 3.1** este sprint comprende las siguientes historias de usuario.

Tabla 3.1: Sprint 0

Sprint	Historia de usuario #	Descripción
0	HU000 – Página de inicio	Como usuario desea poder ver una página principal de la empresa de renta de vehículos

En esta sección se detalla el proceso de instalación y la configuración para el ambiente de desarrollo donde se instalan las herramientas que permitan realizar el componente.

Proceso de instalación de herramientas

Para el desarrollo de este componente se utiliza el entorno de desarrollo llamado Visual Studio Code este es un editor de código desarrollado por la empresa Microsoft, este editor es de software libre y multiplataforma, posee un sinnúmero de extensiones que te permiten escribir código en cualquier lenguaje de programación [19].

Node.js es un entorno de tiempo de ejecución de JavaScript que ofrece capacidades en tiempo real, esta herramienta puede ser personalizada dependiendo de las necesidades del usuario [20]. La instalación de estas herramientas se encuentra en el apartado de **ANEXO II**

Finalizada la instalación de las herramientas a utilizar, se crea la estructura de la página web que incluye las carpetas necesarias para alojar el código, tal como se muestra en la **Figura 3.1**

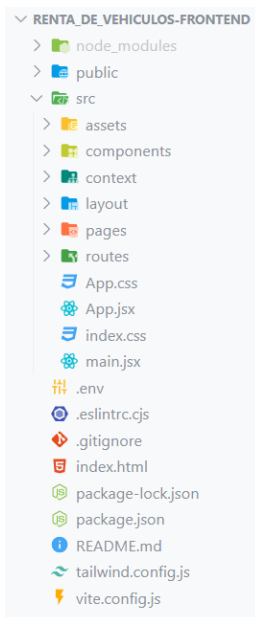


Figura 3.1: Estructura

Implementación de la página de inicio

Este apartado se ha creado para que los usuarios puedan tener una visión de lo que la empresa puede ofrecer y de los servicios que van a encontrar dentro.

Para la realización de esta página se utiliza el lenguaje de marcado HTML y los estilos CSS y JavaScript para darle interactividad y dinamismo para que llegue ser más atractiva para los usuarios de la empresa CarGod se puede observar el diseño de la misma en la **Figura 3.2**.



Figura 3.2: Página de inicio

3.2 Sprint 1.

En la **Tabla 3.2** se puede observar que este *sprint* comprende a las historias de usuario 1,2,3,4 y 5

Tabla 3.2: Sprint 1

Sprint	Historia de usuario#	Descripción
1	HU001 – Registro sitio web propietario	Como propietarios quieren poder registrarse en la página web
1	HU002 – Registro sitio web cliente	Como clientes quieren poder registrarse en la página web
1	HU003 – Inicio de sesión propietario	Como propietarios quieren poder ingresar a la página una vez se encuentre registrado
1	HU004 – Inicio sesión cliente	Como clientes quieren poder ingresar a la página web una vez se encuentre registrado
1	HU005 – Inicio de sesión administrador	Como administrador de la página quiero deseo poder iniciar sesión en el sitio web.

Registro sitio web propietario

Este apartado se crea especialmente para que las personas interesadas en generar ingresos con la ayuda de la página y tengan un vehículo disponible para ofrecer a los clientes. Como se puede observar en **Figura 3.3** el propietario puede ingresar sus datos personales para poder crearse una cuenta, la misma que puede ser utilizada para poder iniciar sesión.

BIENVENIDO

Como propietario tienes la opción de poder rentar vehículos a nuestros clientes.
Ingreso tus datos para poder registrarte:

Nombre (0/30):
Ingresar tu nombre

Apellido (0/30):
Ingresar tu apellido

Dirección (0/60):
Ingresar tu dirección

Teléfono (0/10):
Ingresar tu teléfono

Correo electrónico (0/50):
Ingresar tu correo electrónico

Contraseña (0/20):

Registrar

¿Ya tienes una cuenta? Dirígete al login para poder iniciar sesión.

Iniciar sesión

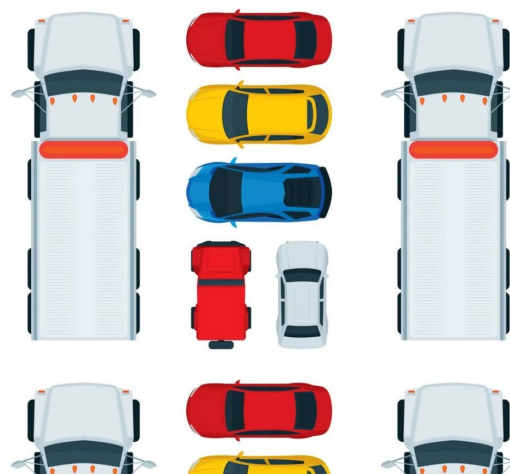


Figura 3.3: Registro propietario

Registro sitio web cliente

Este apartado es creado especialmente para las personas que están interesadas en adquirir un vehículo por un periodo de tiempo referente a días, en esta sección al igual que el propietario el puede registrar diferentes datos personales, que son utilices para crear su cuenta y poder iniciar sesión, tal como se muestra en la **Figura 3.4.**

BIENVENIDO

Con el rol de cliente tu puedes alquilar vehículos
Ingreso tus datos para poder registrarte.

Nombre (0/30):
Ingresar tu nombre

Apellido (0/30):
Ingresar tu apellido

Dirección (0/60):
Ingresar tu dirección

Teléfono (0/10):
Ingresar tu teléfono

Correo electrónico (0/50):
Ingresar tu correo electrónico

Contraseña (0/20):

Registrar

¿Ya tienes una cuenta? Dirígete al login para poder iniciar sesión.

Iniciar sesión



Figura 3.4: Registro cliente

Inicio de sesión propietario

Una vez que el propietario haya podido registrarse ingresando sus datos puede ir al apartado de iniciar sesión para colocar sus credenciales como se puede ver en la **Figura 3.6** el proceso para poder iniciar sesión como ya se encuentra en la variable *handleSubmit* como se puede observar en la **Figura 3.5** aquí se utiliza la URL proporcionada por parte del *backend* para el consumo de los *endpoints*.

```
const handleSubmit = async (e) => {
  e.preventDefault();

  try {
    const url = `${import.meta.env.VITE_BACKEND_URL}/login`;
    const respuesta = await axios.post(url, form);

    if (respuesta && respuesta.data) {
      localStorage.setItem('token', respuesta.data.token);
      localStorage.setItem('role', respuesta.data.role || '');
      setAuth(respuesta.data);
      console.log("Token almacenado en localStorage:", localStorage.getItem('token'));
      console.log("Rol almacenado en localStorage:", localStorage.getItem('role'));

      // Aquí podrías redirigir a diferentes rutas según el rol del usuario
      if (respuesta.data.role === 'propietario') {
        navigate('/propietario'); // Cambia la ruta según tu estructura de rutas
      } else if (respuesta.data.role === 'admin') {
        navigate('/admin'); // Cambia la ruta según tu estructura de rutas
      } else {
        // Otro rol, redirige a una ruta predeterminada
        navigate('/default-route'); // Cambia la ruta según tu estructura de rutas
      }
    } else {
      console.error('La respuesta o su propiedad "data" no están definidas correctamente:', respuesta);
      // Manejo de errores cuando la respuesta o su propiedad "data" no están definidas correctamente
    }
  } catch (error) {
    if (error.response) {

```

Figura 3.5:Lógica login

Iniciar sesión cliente

Una vez que el cliente haya podido crear su cuenta y tenga sus credenciales, el puede dirigirse a iniciar sesión para poder poner sus datos tal como puede observar en la **Figura 3.6** perteneciente al *login* este cuenta con el mismo proceso de la **Figura 3.5**

Inicio de sesión administrador

El usuario administrador no necesita tener un apartado de registro, este directamente tiene una cuenta creada por parte del *backend* donde puede directamente dirigirse al apartado de iniciar sesión mostrado en la **Figura 3.6** y poner sus credenciales para poder ingresar a su apartado directamente.

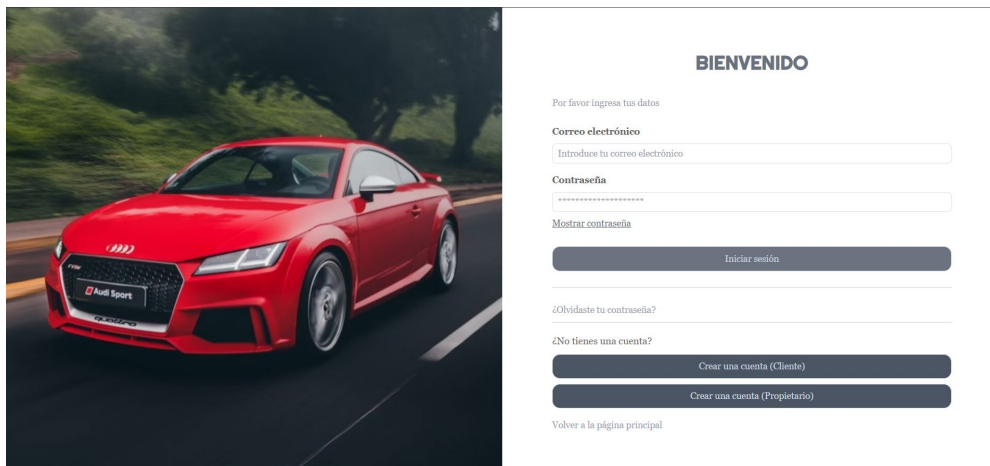


Figura 3.6: Inicio de sesión

3.3 Sprint 2.

En la **Tabla 3.3** se puede observar que las historias de usuario 6, 7, 8 corresponden al *sprint* número 2.

Tabla 3.3: Sprint 2

Sprint	Historia de usuario#	Descripción
2	HU006 – Recuperación de contraseña	Como usuarios, deseamos contar con la opción de restablecer la contraseña mediante el <i>email</i> .
2	HU007 – Registro vehículo propietario	Como propietarios quieren poder registrar un vehículo una vez que hayan accedido a la página web.
2	HU008 – Aprobación de vehículos	Como administrador quiero poder aceptar las solicitudes legítimas que realizan los propietarios
2	HU009 – Visualización de vehículos	Como clientes quieren poder visualizar los vehículos que se encuentran en oferta por parte de los propietarios y son aceptados por el administrador

Recuperación de contraseña para los usuarios

Una vez ya que los diferentes usuarios puedan crear sus cuentas mediante un registro, ellos en caso de necesitarlo pueden pedir una restauración de su contraseña, en este caso podrán irse al apartado de: “¿Haz olvidado tu contraseña?” como se puede observar en la **Figura 3.6** para dirigirse al apartado en el que pueden enviar su correo electrónico como se puede observar en la **Figura 3.8**, la lógica para esta página se encontrara dentro de la **Figura 3.7**. Una vez les salga el mensaje de confirmación ellos pueden dirigirse a su correo electrónico para poder aceptar el mensaje y llenar los campos correspondientes para poder formatear su contraseña se encuentra un ejemplo de esto en la **Figura 3.9** y **Figura 3.10**.

```
export const Forgot = () => {
  const [mensaje, setMensaje] = useState({});
  const [mail, setMail] = useState({});

  const handleChange = (e) => {
    setMail({
      ...mail,
      [e.target.name]: e.target.value,
    });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();

    try {
      const url = `${import.meta.env.VITE_BACKEND_URL}/forgot-password`;
      const respuesta = await axios.post(url, mail);
      setMensaje({
        respuesta: respuesta.data.msg || "Revisa tu correo electronico, para restablecer la contraseña",
        tipo: true });
      setMail("");
    } catch (error) {
      setMensaje({ respuesta: error.response.data.msg || "Lo sentimos el usuario no se encuentra registrado",
        tipo: false });
    }
  };
};
```

Figura 3.7: Lógica recuperación de contraseña

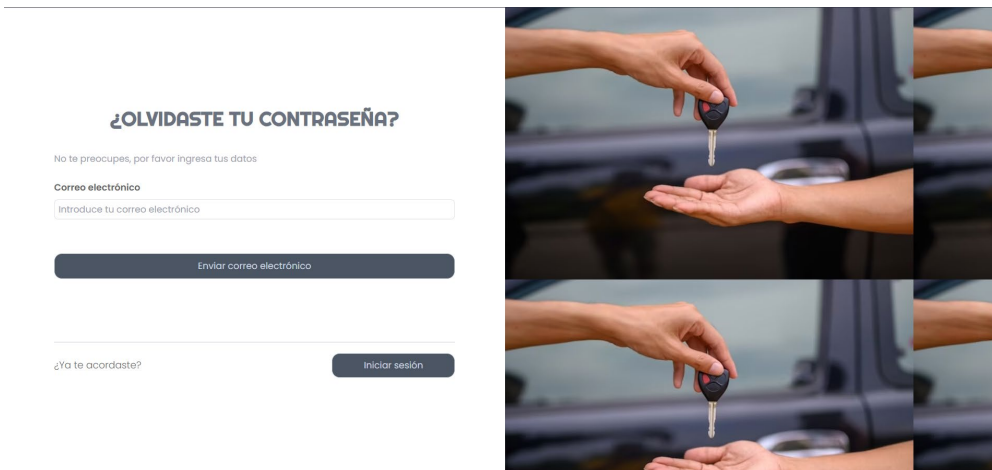


Figura 3.8: Recuperación contraseña

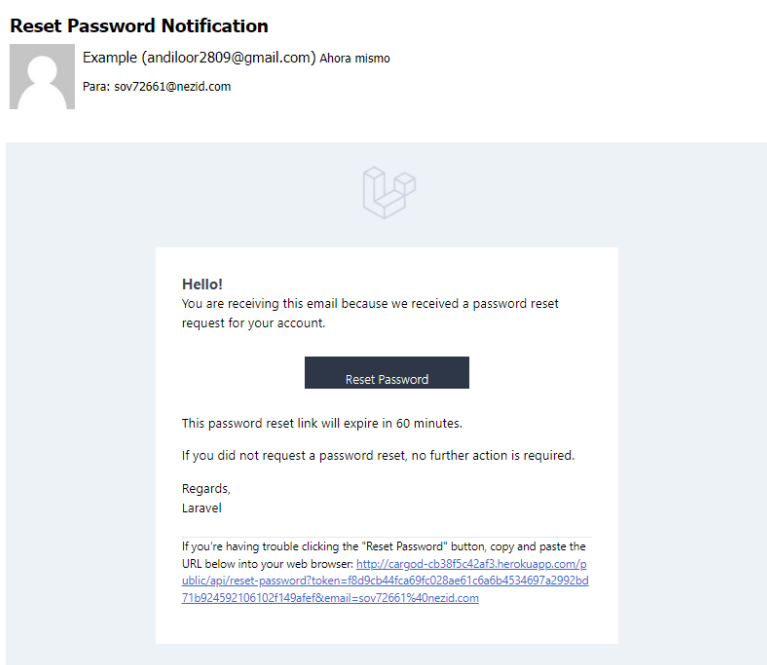


Figura 3.9: Correo para recuperar contraseña

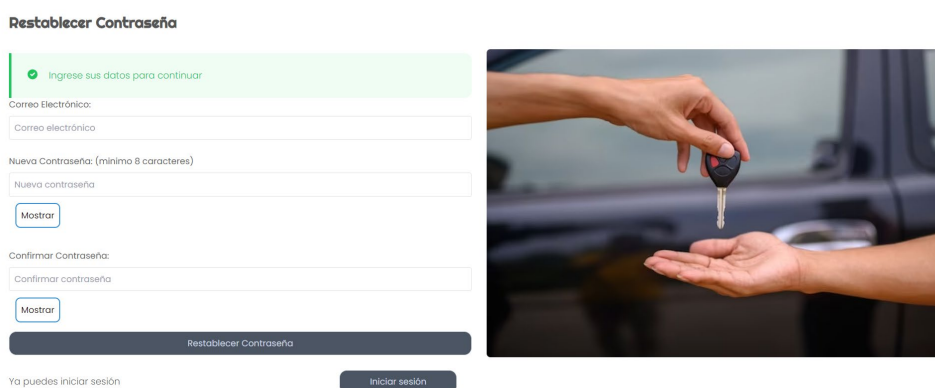


Figura 3.10: Restablecer Contraseña

Registrar un vehículo

Una vez que el propietario haya podido ingresar a la página web mediante el uso de credenciales, el puede registrar un vehículo donde encuentra los siguientes campos: “Tipo de vehículo”, “Marca”, “Placas del vehículo”, “Número máximo de pasajeros”, “Imagen”, “Costo de alquiler”, “Número de contacto”, “Descripciones generales”.

En el apartado de imagen este puede ingresar solo una imagen del vehículo para que el administrador lo vea al aceptar su solicitud y el cliente pueda observarlo cuando quiere contratarlo, como se visualiza en **Figura 3.13**, la lógica para este apartado lo podemos encontrar en la **Figura 3.11** y en la **Figura 3.12**.

```
const handleSubmit = async (e) => {
  e.preventDefault();

  const autenticado = localStorage.getItem('token');

  try {
    const formData = new FormData();
    formData.append("tipo_vehiculo", vehiculoData.tipo_vehiculo);
    formData.append("marca", vehiculoData.marca);
    formData.append("placas", vehiculoData.placas);
    formData.append("numero_pasajero", vehiculoData.numero_pasajero);
    formData.append("image_url", vehiculoData.image_url);
    formData.append("costo_alquiler", vehiculoData.costo_alquiler);
    formData.append("contacto", vehiculoData.contacto);
    formData.append("descripcion", vehiculoData.descripcion);

    const url = `${import.meta.env.VITE_BACKEND_URL}/vehiculos`;
    const respuesta = await axios.post(url, formData, {
      headers: {
        Authorization: `Bearer ${autenticado}`,
        "Content-Type": "multipart/form-data",
      },
    });

    setVehiculoData({
      tipo_vehiculo: "",
      marca: "",
      placas: "",
      numero_pasajero: 0,
      image_url: null,
    });
  }
};
```

Figura 3.11: Lógica para registrar un vehículo

```

66     costo_alquiler: 0,
67     contacto: "",
68     descripcion: "",
69   });
70
71   console.log("Respuesta del servidor:", respuesta.data);
72
73   setMensaje({
74     respuesta: respuesta.data.msg || "Tu solicitud fue enviada correctamente",
75     tipo: true,
76   });
77 } catch (error) {
78   console.error("Error en la solicitud:", error);
79   if (error.response) {
80     console.error("Respuesta del servidor con error:", error.response.data);
81     setMensaje({
82       respuesta: error.response?.data.msg || "Respuesta errónea del servidor",
83       tipo: false,
84     });
85   } else if (error.request) {
86     console.error("No se recibió respuesta del servidor:", error.request);
87     setMensaje({
88       respuesta: error.response?.data.msg || "No se recibió respuesta del servidor",
89       tipo: false,
90     });
91   } else {
92     console.error("Error durante la solicitud:", error.message);
93     setMensaje({
94       respuesta: error.response?.data.msg || "Error durante la solicitud",
95       tipo: false,

```

Figura 3.12: Lógica para registrar un vehículo, parte 2

The screenshot displays the 'Nuevo vehículo' registration form in the CarGod application. The form is set against a blue header with the CarGod logo and navigation links. The form fields are as follows:

- Tipo de vehículo:** A text input field.
- Marca:** A text input field.
- Placas del vehículo:** A text input field.
- Número máximo de pasajeros:** A text input field with the value '0'.
- Imagen:** A file selection area with a button labeled 'Seleccionar archivo' and the text 'Sin archivos seleccionados'.

Figura 3.13: Registro de vehículo

Aprobación de vehículos

Una vez que el propietario llega a registrar su vehículo y envía el formulario este puede visualizar el apartado de “Aceptar Solicitudes” perteneciente al administrador esto lo encontraremos en la **Figura 3.16** y donde este puede ver la información que ha proporcionado el cliente en el botón de “Más información”, y aceptar o rechazar la solicitud del usuario propietario según sea el caso correspondiente, la

visualizaremos en la **Figura 3.17** la implementación de la lógica perteneciente a este apartado se encuentra en la **Figura 3.14** y en la **Figura 3.15**

```
useEffect(() => {
  const obtenerVehiculosPendientes = async () => {
    try {
      const autenticado = localStorage.getItem('token');
      const respuesta = await axios.get(`${import.meta.env.VITE_BACKEND_URL}/solicitudes/pendientes`, {
        headers: {
          Authorization: `Bearer ${autenticado}`,
        },
      });
      console.log('Respuesta del servidor:', respuesta.data);

      if (respuesta.data.success && Array.isArray(respuesta.data.vehiculos_pendientes) && respuesta.data.vehiculos_pendientes.length > 0) {
        console.log('Vehículos pendientes del servidor:', respuesta.data.vehiculos_pendientes);
        setSolicitudes(respuesta.data.vehiculos_pendientes);
        setLoading(false);
      } else {
        setLoading(false);
        setNoVehiclesMessage('No hay vehículos pendientes.');
```

Figura 3.14: Lógica para aprobación de vehículos

```
const handleAccept = async (solicitudId) => {
  try {
    const autenticado = localStorage.getItem('token');
    await axios.put(
      `${import.meta.env.VITE_BACKEND_URL}/estado/${solicitudId}`,
      { estado: 'aceptado' },
      {
        headers: {
          Authorization: `Bearer ${autenticado}`,
        },
      }
    );
    console.log('Solicitud de aceptación exitosa');
    obtenerVehiculosPendientes(); // Actualiza la lista después de aceptar
  } catch (error) {
    console.error('Error al aceptar la solicitud:', error);
  }
};

const handleReject = async (solicitudId) => {
  try {
    const autenticado = localStorage.getItem('token');
    await axios.put(
      `${import.meta.env.VITE_BACKEND_URL}/estado/${solicitudId}`,
      { estado: 'rechazado' },
      {
        headers: {
          Authorization: `Bearer ${autenticado}`,
        },
      }
    );
    console.log('Solicitud de rechazo exitosa');
```

Figura 3.15: Lógica para aprobación de vehículos, parte 2



Figura 3.16: Barra de navegación admin

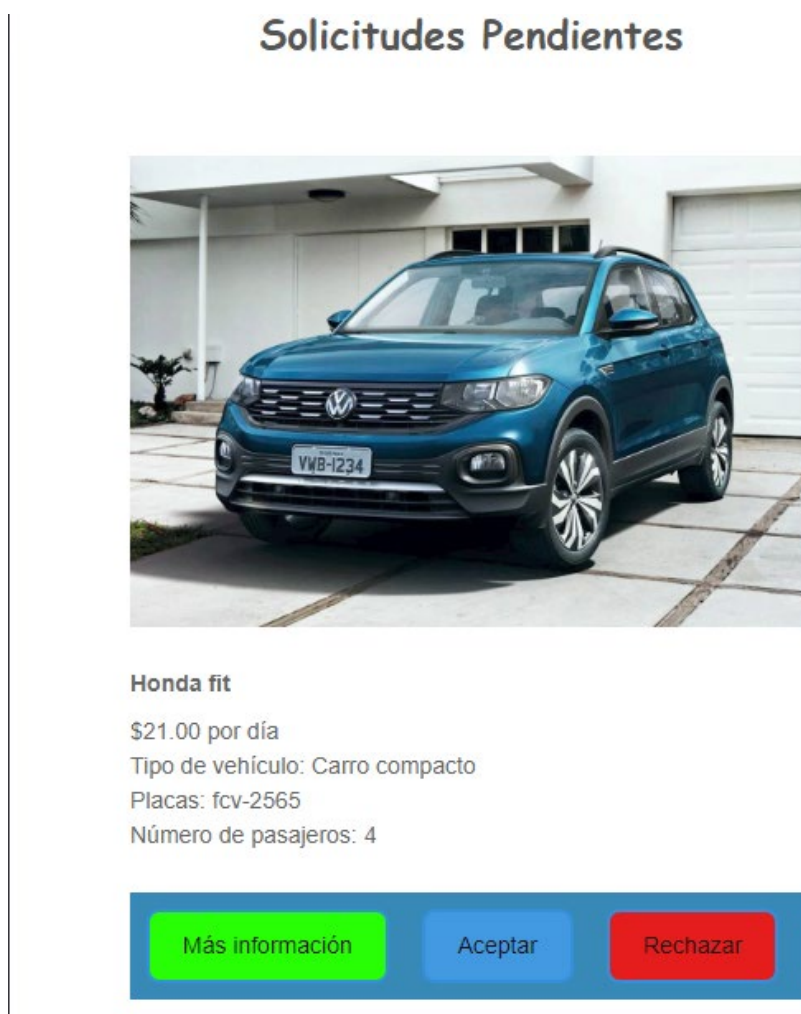


Figura 3.17: Solicitudes pendientes

Visualización de vehículos

Cuando el administrador haya aceptado las solicitudes de los diferentes propietarios estas son visibles para los clientes, dándole una vista general de las diferentes solicitudes y así poder tomar una decisión, la vista de esta pantalla se observa en la **Figura 3.19** y en la **Figura 3.20**, adicionalmente se puede ver la lógica de este apartado en la **Figura 3.18**.

```
useEffect(() => {
  const fetchVehiculos = async () => {
    try {
      const token = localStorage.getItem('token');
      console.log('Token:', token);
      const response = await axios.get(`${import.meta.env.VITE_BACKEND_URL}/solicitudes/aceptados`, {
        headers: {
          Authorization: `Bearer ${token}`,
        },
      });

      if (response.data && response.data.success) {
        setVehiculos(response.data.vehiculos_aceptados);

        if (response.data.vehiculos_aceptados.length === 0) {
          setMessage('Aun no contamos con vehículos disponibles para su disposición. Lo sentimos, vuelve pronto.');
        }
      } else {
        console.error('La respuesta de la API no tiene el formato esperado:', response.data);
        setError('Error al obtener la lista de vehículos. Formato inesperado. Por favor, inténtalo de nuevo más tarde.');
      }

      setLoading(false);
    } catch (error) {
      console.error('Error al obtener la lista de vehículos', error);
      setError('Error al obtener la lista de vehículos. ${error.response ? error.response.data.message : error.message}');
      setLoading(false);
    }
  };
});
```

Figura 3.18: Lógica visualización vehículos



Vehículos Disponibles



Figura 3.19: Navegación cliente



\$25.00 por día
 Tipo de vehículo: Deportivo
 Placas: gtx-897
 Número de pasajeros: 0



Figura 3.20: Vista de vehículos para contratar

3.4 Sprint 3.

En la **Tabla 3.4** se puede observar que las historias de usuario 10 y 11 corresponden al *sprint 3*

Tabla 3.4: Sprint 3

Sprint	Historia de usuario#	Descripción
3	HU010 – Contratación de un vehículo registrado.	Como cliente se quiere poder observar una descripción general de las solicitudes de los carros, para poder tomar una decisión.
3	HU011- Cantidad de días que puede rentar un vehículo.	Una vez se haya tomado una decisión por parte del cliente este desea poder seleccionar la cantidad de días por la cual va a rentar el vehículo.

Contratación de un vehículo registrado

Una vez que se haya pasado por todo este proceso de publicaciones acerca de los vehículos el cliente quiere hacer una contratación de este, buscando así una solicitud que se ajuste a sus necesidades, ya sea un vehículo para 2 pasajeros o 4, hasta un vehículo con un precio más ajustado a su valor económico, una vista general de este apartado, se puede encontrar en la **Figura 3.24** y la lógica perteneciente se encuentra en las figuras: **Figura 3.21**, **Figura 3.22**, **Figura 3.23**.

```
const handleRentClick = (vehiculoId) => {
  const vehiculo = vehiculos.find((v) => v.id === vehiculoId);
  setSelectedVehiculo(vehiculo);
  setShowRentModal(true);
};
```

Figura 3.21: Lógica para el botón contratar

```
const handleConfirmRent = async () => {
  try {
    const token = localStorage.getItem('token');
    const response = await axios.post(
      `${import.meta.env.VITE_BACKEND_URL}/contratos/${selectedVehiculo.id}/${duracionContrato}`,
      {
        vehiculoId: selectedVehiculo.id,
        duracion: duracionContrato,
      },
      {
        headers: {
          Authorization: `Bearer ${token}`,
        },
      }
    );

    console.log('Alquiler exitoso:', response.data);

    // Mostrar mensaje de éxito
    setSuccessMessage('¡Su contrato fue realizado exitosamente!');
    setHideSuccessMessage(false); // Mostrar el mensaje

    // Ocultar el mensaje después de 5 segundos
    setTimeout(() => {
      setHideSuccessMessage(true);
    }, 5000);

    // Cerrar la ventana emergente
    setShowRentModal(false);
  } catch (error) {
    console.error('Error durante el proceso de alquiler:', error);
  }
};
```

Figura 3.22: Lógica para aceptar el contrato

```
const handleShowMoreInfo = (vehiculo) => {  
  setSelectedVehiculo(vehiculo);  
  setShowMoreInfoModal(true);  
};  
  
const handleCloseModal = () => {  
  setShowMoreInfoModal(false);  
  setShowRentModal(false);  
};
```

Figura 3.23: Lógica para mostrar más info y cerrar ventanas emergentes

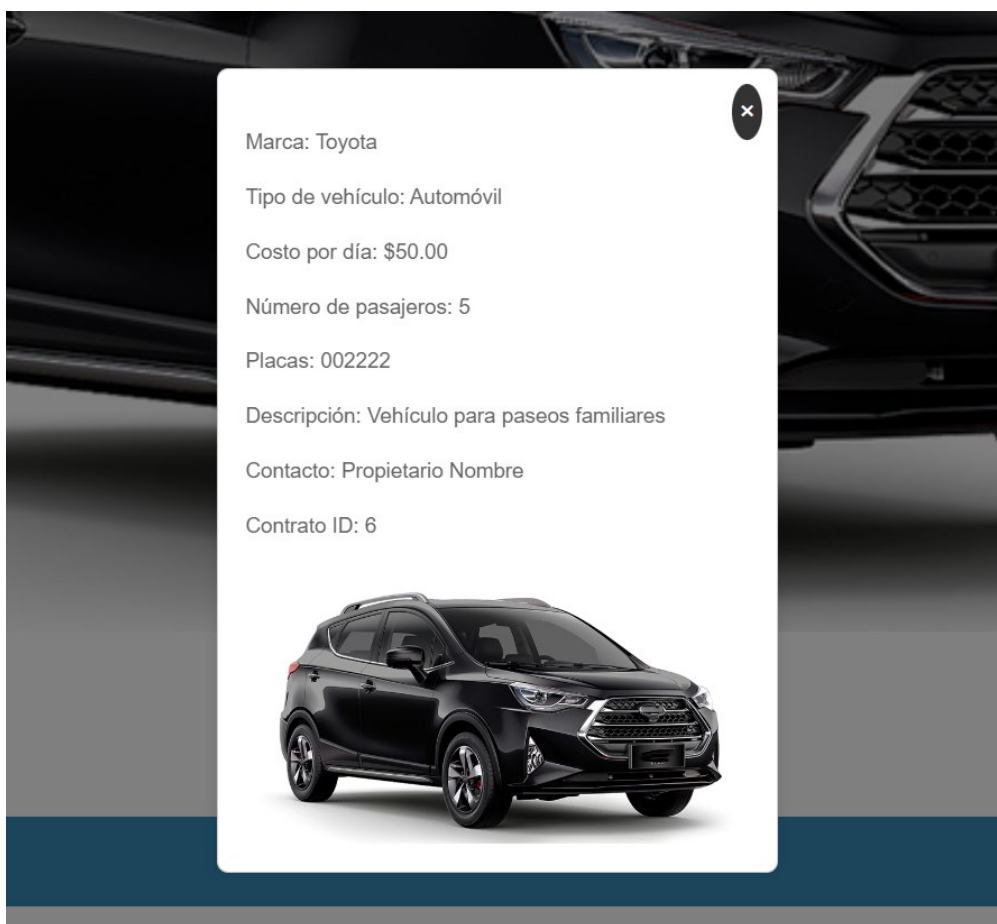
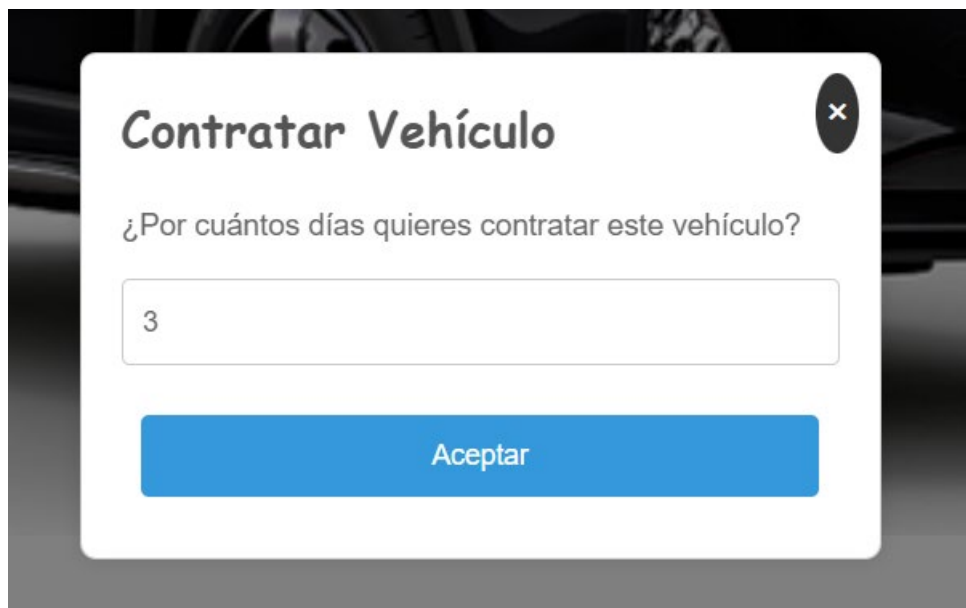


Figura 3.24: Ventana emergente, cliente.

Cantidad de días que puede rentar un vehículo

Cuando el cliente ya haya logrado tomar una decisión con respecto a que vehículo rentar, este puede decidir por que cantidad de días podrá llegarlo a tener, es decir, por cuanto tiempo deseara contratar este vehículo, se podrá ver esto en la **Figura 3.25** dando un mensaje de éxito como se puede observar en la **Figura 3.26**.



Contratar Vehículo ✕

¿Por cuántos días quieres contratar este vehículo?

Figura 3.25: Número de días para contratar

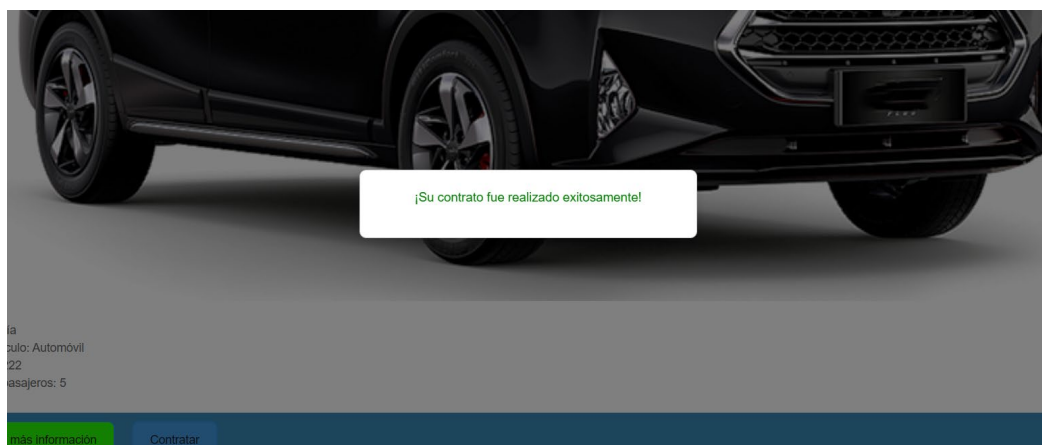


Figura 3.26: Mensaje de éxito para contratar

3.5 Sprint 4.

En la **Tabla 3.5** se puede observar que la historia de usuario 12 corresponde al *sprint 4*

Tabla 3.5: Sprint 4

Sprint	Historia de usuario#	Descripción
4	HU012 – Visualización del historial de pedidos	Una vez que el cliente haya realizado ya su reservación, podrá visualizar en un apartado todos los pedidos que ha logrado realizar con éxito.

Visualización del historial de pedidos

El cliente una vez haya realizado la contratación de un vehículo este podrá dirigirse a su apartado correspondiente donde puede observar todo el historial de pedidos que haya realizado, dándole así una visión más general de sus consumos, con esto se busca que el cliente recaude la mejor información posible para que siga realizando pedidos dentro de la página, el modelo de este apartado se puede observar en **Figura 3.29** y en la **Figura 3.30** con su lógica correspondiente perteneciente a la **Figura 3.27**.

```
useEffect(() => {
  const fetchHistorialPedidos = async () => {
    try {
      const token = localStorage.getItem('token');
      if (!token) {
        console.error('Token de autenticación no encontrado');
        return;
      }

      const response = await axios.get(`${import.meta.env.VITE_BACKEND_URL}/contratados`, {
        headers: {
          Authorization: `Bearer ${token}`,
        },
      });

      // Verificar si la respuesta contiene un array antes de actualizar el estado
      if (response.data.success && Array.isArray(response.data.contratos_detallados)) {
        setHistorialPedidos(response.data.contratos_detallados);
      } else {
        console.error('La respuesta del servidor no es válida', response.data);
      }
    } catch (error) {
      console.error('Error al obtener el historial de pedidos', error);
    }
  };

  fetchHistorialPedidos();
}, []);
```

Figura 3.27: Lógica para visualización



Figura 3.28: Navegación cliente



Vehículo: Honda fit - Electrico
Placas: fgn-456
Duración: 5 días

Ver más detalles

Figura 3.29: Visualización del contrato

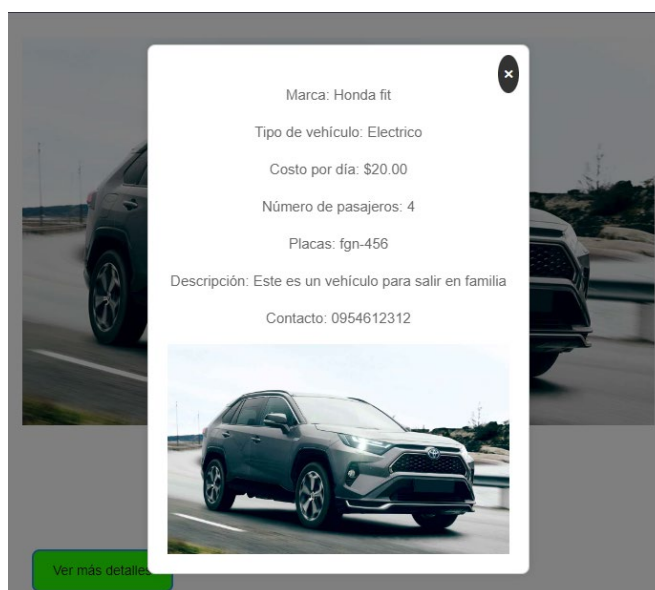


Figura 3.30: Detalles del contrato

3.6 Sprint 5.

En la **Tabla 3.6** se puede observar que la historia de usuario 13 corresponde al *sprint 5*

Tabla 3.6: Sprint 5

Sprint	Historia de usuario#	Descripción
5	HU013 – Pruebas y despliegue	Una vez ya todas estas funciones estén completas la aplicación deberá ser desplegada mediante un servicio web.

Pruebas

Hacer pruebas es fundamental en el desarrollo de aplicaciones web, ya que aseguran su calidad y estabilidad. Estas pruebas son útiles para identificar errores y garantizar el correcto funcionamiento del sistema. En esta sección, se evalúa el rendimiento de la aplicación a través de diferentes pruebas.

Pruebas de aceptación

Las pruebas de aceptación están hechas para garantizar que el software cumpla con las expectativas y requisitos de los clientes antes de subirlo a un entorno de producción, esto con el objetivo de evitar que una aplicación sea desplegada cuando no es de agrado para los usuarios, para el desarrollo ágil estas pruebas son parte primordial del proceso de desarrollo [21]. Las pruebas de aceptación para este proyecto son a través de una encuesta proporcionada para los estudiantes de tecnología superior en desarrollo de software y usuarios interesados en la realización de este proyecto. Los resultados de estas pruebas de aceptación se pueden encontrar en el **ANEXO II**.

Pruebas de compatibilidad

Realizar *test* de multi navegador, estas son realizadas para verificar en que navegadores es compatible el *website* ya que no todas las personas llegan a usar el conocido navegador de Google Chrome, la conexión de internet puede estar bien, pero la página puede mostrar diversos errores al momento de visualizar el contenido [22]. Los resultados de este *test* se encuentran en la **Tabla 3.7**, estas pruebas son realizadas en *testingbot*.

Tabla 3.7: Compatibilidad navegadores

Sistema	Navegador	Versión	Dispositivo	Resultado
Windows 10	Chrome	119.0	Escritorio	Compatible
Windows 10	FireFox	120	Escritorio	Compatible
Windows 10	Edge	119	Escritorio	Compatible
Windows 10	Opera	102	Escritorio	Compatible
Windows 7	Chrome	108	Escritorio	Compatible
Windows 7	FireFox	114.0	Escritorio	Compatible
IOS	Safari	iOS 16.2	iPhone 14 Pro Max	Compatible

Pruebas de accesibilidad

Para estas pruebas se utiliza la herramienta *pagespeed insights*, esta está enfocada a encontrar y solucionar problemas que llegan a ralentizar las aplicaciones web, es de código abierto que recopila y analiza los datos de laboratorio que se combinan y los compara con información del mundo real referente a la experiencia de usuario del usuario de Google Chrome, proporcionando una puntuación que resume el rendimiento y algunas sugerencias para quienes desarrollan la página [23]. En la **Figura 3.31** se puede observar los resultados de escritorio y en la **Figura 3.32** se observará los resultados para dispositivos móviles. Los detalles de los resultados se pueden encontrar en el apartado de **ANEXO II** pruebas de accesibilidad.

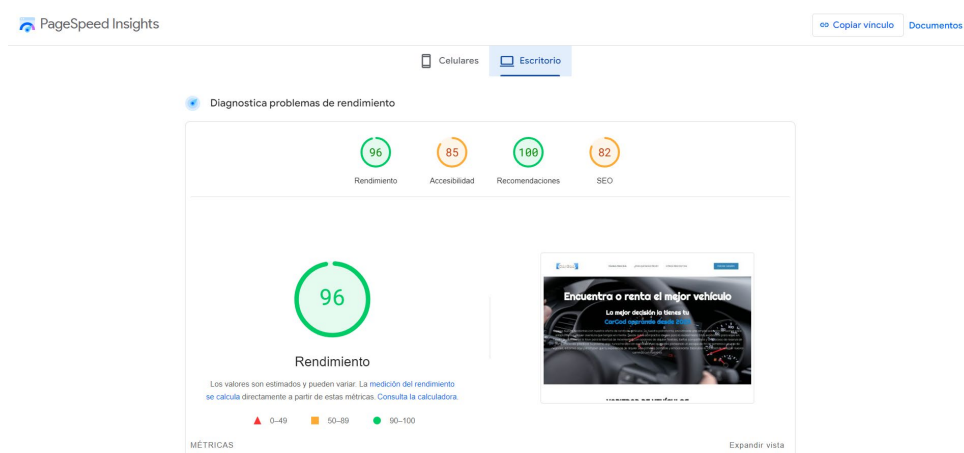


Figura 3.31: Resultados escritorio

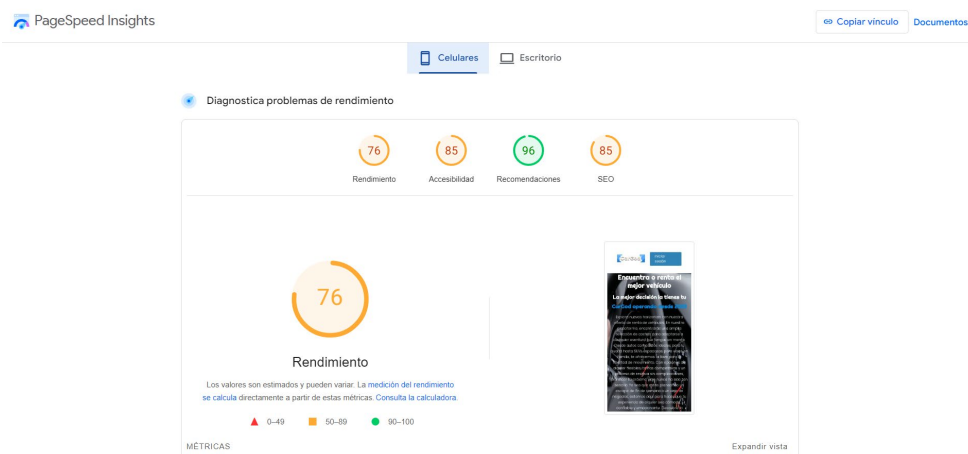


Figura 3.32: Resultados celulares

Despliegue

Para la realización del despliegue se utiliza la plataforma *netlify*, este es un servicio de hosting para sitios webs estáticos, este ofrece poder alojar un sitio web con el servicio *git*, en este caso se está utilizando el servicio *github* para el despliegue. Esto es muy útil a la hora de mantener una web, ya que todos los cambios aplicados en el repositorio se ven reflejados inmediatamente en el sitio web [24].

En la **Figura 3.33** se puede observar una vista breve del sitio web, como el link del despliegue con el respectivo nombre que le haya dado, en este caso es: <https://cargod.netlify.app/> al igual que se encuentra a que servicio git esta arraigado,

quien es el propietario del sitio web y hace cuanto fue la última actualización de este.

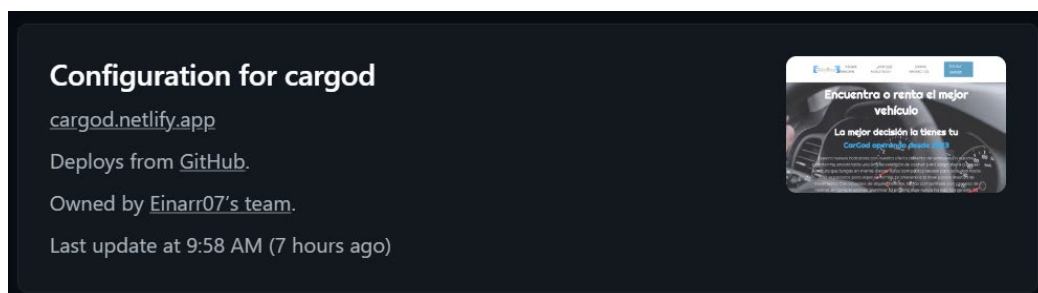


Figura 3.33: Despliegue

4 CONCLUSIONES

A continuación, se presentan las conclusiones alcanzadas en el proceso de integración curricular.

- Realizar el diseño de los prototipos fue esencial al momento de implementar este proyecto y al momento de construir cada componente, ya que permite tener una idea clara tanto por parte del *frontend* como del *backend* permitiendo así compartir ideas para mejorar el sistema y al momento de escoger una estrategia que garantice el desarrollo eficiente del mismo.
- Tener en consideración que patrón arquitectónico se va a realizar en un proyecto que tiene un estilo dinámico es importante y esencial, para este trabajo se realizó el patrón arquitectónico MVC (Modelo Vista Controlador), teniendo en cuenta que debe enviar y recibir datos por parte del *backend* con el uso de llamadas API's, se puede lograr una respuesta satisfactoria.
- La metodología *scrum* resulto de gran ayuda para tener claro los tiempos a considerar entre cada avance que se debe realizar en este trabajo. Al enfocarse en un sprint semanal, facilita una visión clara y progresiva de todo el desarrollo.
- Se elige utilizar *react* para el marco de desarrollo debido a su eficacia en la creación de interfaces de usuario, permitiendo hacer distintos tipos de componentes y poder conectarlos entre sí, para dar un enfoque dinámico al proyecto, permitiendo la reutilización de su estructura mejorando la experiencia del usuario final significativamente.
- Para corroborar el correcto funcionamiento de esta aplicación web se realizan diversos tipos de pruebas donde se calificó el desempeño de la aplicación que son útiles para corregir distintos errores, en cuando a su vista y estructura, logrando de esta manera un sitio web más estable.
- El despliegue de esta aplicación se ha realizado en *netlify* gracias a que esta plataforma permite alojar contenido dinámico y se la puede conectar directamente con *github*, el sitio en el que se encuentra el repositorio permitiendo de esta manera tener un sitio web siempre actualizado a todos los cambios que sean aplicados dentro de este.

5 RECOMENDACIONES

A continuación, se listarán las recomendaciones sobre el proceso del trabajo de integración curricular.

- Para el correcto funcionamiento de un sitio web, se debe tener en cuenta lo que se desea transmitir, ya sea como individuo o como empresa, definiendo el propósito de esta. Para tener claras las pautas de lo que podrán ver todos los usuarios que visiten la página, se deben establecer objetivos claros y funcionales, los recursos que están a nuestra disposición y como sacar provecho de estos.
- Para hacer la implementación de manera efectiva se debe ser claros con la metodología empleada, se debe escoger una capaz de cumplir todos los requisitos necesarios para todo lo que se tiene en mente realizar y cuánto tiempo se le dedicara, saber distinguir si beneficiara una metodología ágil respecto a una metodología tradicional. Cuando el panorama está claro esto, se podrá estar en sintonía con los demás miembros de desarrollo del proyecto, facilitando la revisión progresiva del mismo, identificando de manera pronta si existe algún obstáculo en los avances del trabajo.
- Siempre que se llega a desarrollar una aplicación se presentarán obstáculos o errores que retrasen las tareas de esta, por lo cual es recomendable siempre visitar los sitios en los que se encuentra la documentación oficial esta siempre va a proporcionar ejemplos claros para el tipo de trabajo que se está llevando a cabo.
- Para tener una buena implementación dentro del proyecto se deben verificar todas las herramientas a utilizar, tener un claro qué es lo que se necesita para este desarrollo, es decir, tener presente que entorno de desarrollo es favorable, que librerías son adecuadas, que dependencias tendrá el proyecto, ya que implementar una aplicación web se puede realizar en distintos lenguajes de programación, con diferentes directrices y obtener los mismos resultados, ya depende de nuestras capacidades para determinar si estos serán buenos o malos dependiendo únicamente de cómo se prefiera afrontar.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] R. d. encanto, «¿Cómo alquilar un auto en 3 simples pasos en Quito? | Rutas de Encanto | Ecuador.,» Youtube, 16 Febrero 2023. [En línea]. Available: <https://www.youtube.com/watch?v=P7jbJT-sJrQ>. [Último acceso: 19 Noviembre 2023].
- [2] Localiza, «Renta un auto en Localiza,» Localiza Rent a car, [En línea]. Available: <https://www.localiza.com/ecuador/es-ec>. [Último acceso: 19 Noviembre 2023].
- [3] «Cómo alquilar un carro,» La comisión dedederal de comercio consejos para consumidores, Septiembre 2023. [En línea]. Available: <https://consumidor.ftc.gov/articulos/como-alquilar-un-carro>. [Último acceso: 19 Noviembre 2023].
- [4] N. M. Mayra Moso, «Proyecto de factibilidad para la creación de una microempresa dedicada al servicio de alquiler de automóviles ubicada en el sector norte de la ciudad de Quito,» UCE repositorio institucional, Abirl 2016. [En línea]. Available: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.dspace.uce.edu.ec/server/api/core/bitstreams/dc032ee0-0de5-4526-aab2-e22ae861106e/content>. [Último acceso: 19 Noviembre 2023].
- [5] R. KeepCoding, «¿Qué es el desarrollo fronend?,» KeepCoding, 8 Marzo 2023. [En línea]. Available: <https://keepcoding.io/blog/que-es-el-desarrollo-front-end/>. [Último acceso: 19 Noviembre 2023].
- [6] J. Martins, «Scrum: conceptos clave y cómo se aplica en la gestión de proyectos,» asana, 19 Junio 2023. [En línea]. Available: <https://asana.com/es/resources/what-is-scrum>. [Último acceso: 19 Noviembre 2023].
- [7] G. Blandino, «Figma: qué es y cómo funciona,» blog., 4 Enero 2023. [En línea]. Available: <https://www.pixartprinting.es/blog/figma-que-es/>. [Último acceso: 19 Noviembre 2023].
- [8] D. A, «Qué es React: definición, características y funcionamiento,» Hostinger, 29 Junio 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-react>. [Último acceso: 19 Noviembre 2023].
- [9] M. Coppola, «Frontend y backend: qué son, en qué se diferencian y ejemplos,» HubSpot, 19 Septiembre 2022. [En línea]. Available: <https://blog.hubspot.es/website/frontend-y-backend>. [Último acceso: 19 Noviembre 2023].
- [10] S. G. SOTOMAYOR, «Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa,» IEBS, 14 Noviembre 2023. [En línea]. Available: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/#:~:text=Por%20definici%C3%B3n%2C%20las%20metodolog%C3%ADas%20%C3%A1giles,las%20circunstancias%20espec%C3%ADficas%20del%20entorno..> [Último acceso: 19 Noviembre 2023].

- [11] J. A. Corrales, «Interfaz de usuario o UI: ¿qué es y cuáles son sus características?,» rockcontent, 2 Agosto 2019. [En línea]. Available: <https://rockcontent.com/es/blog/interfaz-de-usuario/>. [Último acceso: 19 Noviembre 2023].
- [12] C. Ortega, «Investigación-acción: Qué es, etapas y ejemplos,» QuestionPro, 29 Julio 2022. [En línea]. Available: https://www.questionpro.com/blog/es/investigacion-accion/#Que_es_la_investigacion-accion. [Último acceso: 20 Noviembre 2023].
- [13] G. Mancuzo, «5 Fases de la Metodología Scrum,» Compara Software, 10 Julio 2020. [En línea]. Available: <https://blog.comparasoftware.com/fases-metodologia-scrum/>. [Último acceso: 20 Noviembre 2023].
- [14] «Roles de Scrum y sus responsabilidades,» Integrait, [En línea]. Available: <https://integrait.com.mx/blog/roles-de-scrum/#:~:text=Est%C3%A1%20conformado%20por%203%20roles,Miembros%20del%20Equipo%20de%20Desarrollo>.. [Último acceso: 21 Noviembre 2023].
- [15] M. REHKOPF, «Historias de usuario con ejemplos y plantilla,» ATLISSIAN, [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 25 Noviembre 2023].
- [16] M. REHKOPF, «Sprints de scrum,» ATLISSIAN, [En línea]. Available: <https://www.atlassian.com/es/agile/scrum/sprints#:~:text=gesti%C3%B3n%20de%20proyectos%3F-Un%20sprint%20es%20un%20per%C3%ADodo%20breve%20de%20tiempo%20fijo%20en,con%20menos%20quebraderos%20de%20cabeza>.. [Último acceso: 25 Noviembre 2023].
- [17] «Diseño de Interfaces,» Universidad oberta de catalunya, [En línea]. Available: <https://multimedia.uoc.edu/blogs/dii/es/que-es-una-interficie/>. [Último acceso: 25 Noviembre 2023].
- [18] «Cómo diseñar una arquitectura de software: Consejos y prácticas,» Lucidchart, 2023. [En línea]. Available: <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software>. [Último acceso: 6 Diciembre 2023].
- [19] F. Flores, «Qué es Visual Studio Code y qué ventajas ofrece,» OpenWebinars, 22 Julio 2022. [En línea]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. [Último acceso: 28 Noviembre 2023].
- [20] J. Lucas, «Qué es NodeJS y para qué sirve,» OpenWebinars, 4 Septiembre 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-nodejs/>. [Último acceso: 28 Noviembre 2023].
- [21] «Pruebas de Aceptación: El Qué y Porqué + los tipos que hay que conocer,» nimble Humaniza Work, 2023. [En línea]. Available: <https://www.nimblework.com/es/agile/pruebas-de-aceptacion/>. [Último acceso: 4 Enero 2024].

- [22] «Qué es un test multi navegador: definición y ejecución,» Comparium, 2023. [En línea]. Available: <https://www.electronic.us/es/cross-browser-testing-tool/what-is-cross-browser-testing/>. [Último acceso: 5 Enero 2024].
- [23] A. López, «Guía completa de Page Speed Insights: qué es, cómo usarlo y cómo optimizar contenidos con él,» rockcontent, 9 Mayo 2022. [En línea]. Available: <https://rockcontent.com/es/blog/page-speed-insights/>. [Último acceso: 12 Enero 2024].
- [24] M. Merino, «Cómo publicar una web gratis usando Netlify y Github,» Genbeta, 10 Mayo 2020. [En línea]. Available: <https://www.genbeta.com/desarrollo/como-publicar-web-gratis-usando-netlify-github>. [Último acceso: 28 Diciembre 2023].

7 ANEXOS

ANEXO I. Turnitin

ANEXO II. Manual técnico

ANEXO III. Manual de usuario

ANEXO IV. Manual de instalación

ANEXO I

ANEXO II

Manual Técnico

Este anexo describe los componentes utilizados para el desarrollo de este componente mediante la metodología *scrum* que se utilizó para la implementación de los requerimientos, historias de usuario, interfaz gráfica y las pruebas necesarias para corroborar el uso correcto de esta aplicación.

Historias de usuario

Este apartado presenta las historias de usuario de manera secuencial.

Tabla 7.1: Historias de usuarios - Registro propietario

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Propietario
Nombre de la historia: Mostrar registro sitio web propietario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 1	
Responsable: Mateo Congo	
Descripción: Como propietario quiero poder registrarme en el sitio web	
Observaciones: Si el propietario olvida su contraseña podrá recuperarla mediante su correo electrónico.	

Tabla 7.2: Historias de usuarios - Registro cliente

HISTORIA DE USUARIO	
Identificador (ID): HU002	Usuario: Cliente
Nombre de la historia: Mostrar registro sitio web cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 1	
Responsable: Mateo Congo	
Descripción: Como cliente quiero poder registrarme en el sitio web	
Observaciones: Si el cliente olvida su contraseña podrá recuperarla mediante su correo electrónico.	

Tabla 7.3: Historias de usuarios - Inicio sesión propietario

HISTORIA DE USUARIO	
Identificador (ID): HU003	Usuario: Propietario
Nombre de la historia: Mostrar inicio de sesión propietario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 1	
Responsable: Mateo Congo	
Descripción: Como propietario quiero iniciar sección una vez que ya me haya registrado	

Tabla 7.4: Historias de usuarios - Inicio sesión cliente

HISTORIA DE USUARIO	
Identificador (ID): HU004	Usuario: Cliente
Nombre de la historia: Mostrar inicio de sesión cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 1	
Responsable: Mateo Congo	
Descripción: Como cliente quiero iniciar sección una vez que ya me haya registrado.	

Tabla 7.5: Historias de usuarios - Inicio sesión administrador

HISTORIA DE USUARIO	
Identificador (ID): HU005	Usuario: Administrador
Nombre de la historia: Mostrar inicio de sesión administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 1	
Responsable: Mateo Congo	
Descripción: Como administrador quiero poder iniciar sesión en el sitio web sin tener que registrarme	

Tabla 7.6: Historias de usuarios - Recuperación contraseña

HISTORIA DE USUARIO	
Identificador (ID): HU006	Usuario: Cliente – Propietario - Administrador
Nombre de la historia: Mostrar recuperación de contraseña	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 1	
Responsable: Mateo Congo	
Descripción: Como usuario quiero tener la posibilidad de poder recuperar la contraseña en caso de ser olvidada.	

Tabla 7.7: Historias de usuarios - Registro propietario

HISTORIA DE USUARIO	
Identificador (ID): HU007	Usuario: Propietario
Nombre de la historia: Mostrar registro vehículo propietario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 2	
Responsable: Mateo Congo	
Descripción: Como propietario quiero poder tener un apartado donde pueda registrar un vehículo.	

Tabla 7.8: Historias de usuarios - Aprobación de vehículos

HISTORIA DE USUARIO	
Identificador (ID): HU008	Usuario: Administrador
Nombre de la historia: Mostrar aprobación de vehículos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 2	
Responsable: Mateo Congo	

Descripción: Como administrador quiero poder aceptar las solicitudes de vehículos disponibles

Tabla 7.9: Historias de usuarios - Visualización vehículos

HISTORIA DE USUARIO	
Identificador (ID): HU009	Usuario: Cliente
Nombre de la historia: Mostrar visualización vehículos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 2	
Responsable: Mateo Congo	
Descripción: Como cliente quiero poder ver las ofertas de los propietarios que registraron sus vehículos.	

Tabla 7.10: Historias de usuarios - Contratación vehículo

HISTORIA DE USUARIO	
Identificador (ID): HU010	Usuario: Cliente
Nombre de la historia: Mostrar contratación de vehículo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 3	
Responsable: Mateo Congo	
Descripción: Como cliente quiero poder contratar el vehículo que más se ajuste a mis comodidades.	
Observaciones: Se tiene que ver las especificaciones hechas por el propietario del vehículo.	

Tabla 7.11: Historias de usuarios - Tiempo de contratación

HISTORIA DE USUARIO	
Identificador (ID): HU011	Usuario: Cliente
Nombre de la historia: Mostrar tiempo de contratación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto

Sprint asignado: 3
Responsable: Mateo Congo
Descripción: Como cliente quiero poder escoger el tiempo de contrato en el que voy a poseer el vehículo.
Observaciones: Se incrementa el tiempo de contrato, se incrementa el valor.

Tabla 7.12: Historias de usuarios - Historial de pedidos

HISTORIA DE USUARIO	
Identificador (ID): HU012	Usuario: Cliente
Nombre de la historia: Mostrar historial de pedidos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: 4	
Responsable: Mateo Congo	
Descripción: Como cliente quiero poder ver el historial de todos los pedidos que eh hecho en la página.	
Observaciones: Se debe observar la información completa	

Prototipado

Este apartado muestra los prototipos utilizados para la creación de la aplicación web para la renta de vehículos de manera secuencial por parte de cada usuario.

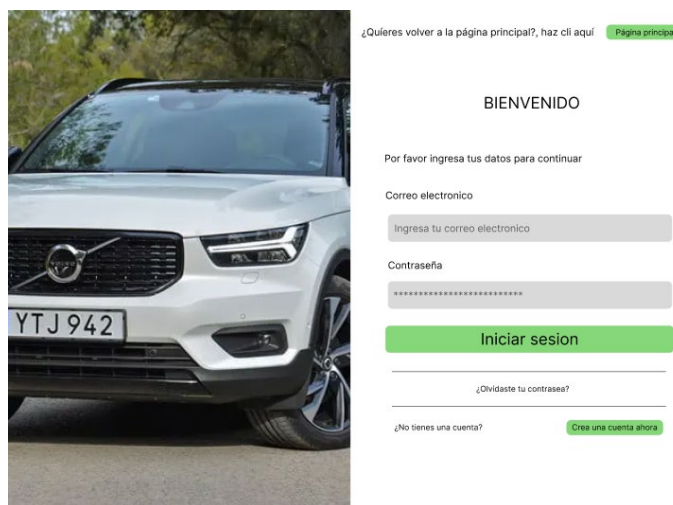


Figura 7.1: Login figma



Figura 7.2: Creación de cuenta figma

BIENVENIDO

Con el rol PROPIETARIO tienes la opción de poder rentar vehículos a nuestros clientes
Ingresa tus datos para poder registrarte

Nombre (0/30)
Ingresa tu nombre

Apellido (0/30)
Ingresa tu apellido

Dirección (0/60)
Ingresa tu dirección

Teléfono (0/10)
Ingresa tu teléfono

Correo electrónico (0/50)
Ingresa tu correo electrónico

Contraseña (0/20)

[Registrar](#)

¿Ya tienes una cuenta? Dirígete al login para iniciar sesión [Iniciar sesión](#)



Figura 7.3: Registro usuario propietario figma

¿OLVIDASTE TU CONTRASEÑA?

No te preocupes, ingresa tu correo

Correo electrónico

¿Lo recordaste?, inicia sesión por favor

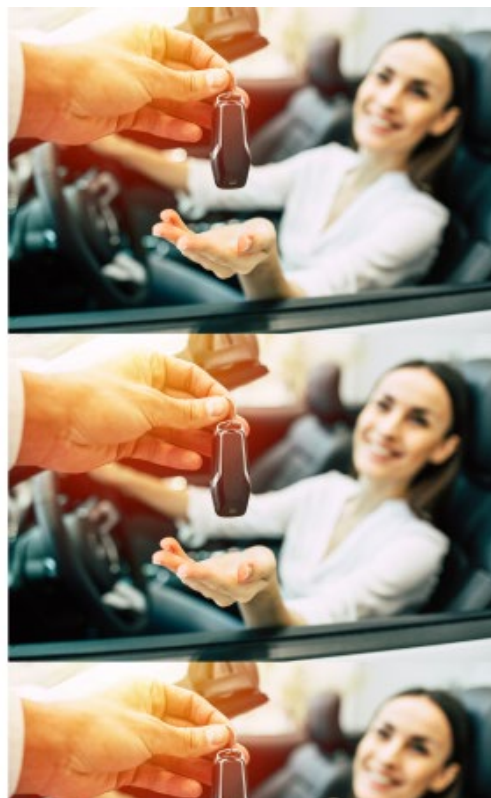


Figura 7.4: Recuperación de contraseña figma

CARGOD
Editar catalogo
Registrar un vehículo
nameUser
Salir

Nuevo vehículo

Tipo de vehículo

Marca

Número de placas

Número máximo de pasajeros

Imagen

Seleccionar archivo

Costo de alquiler

Número de contacto

Descripciones generales

Figura 7.5: Registro de vehículo figma

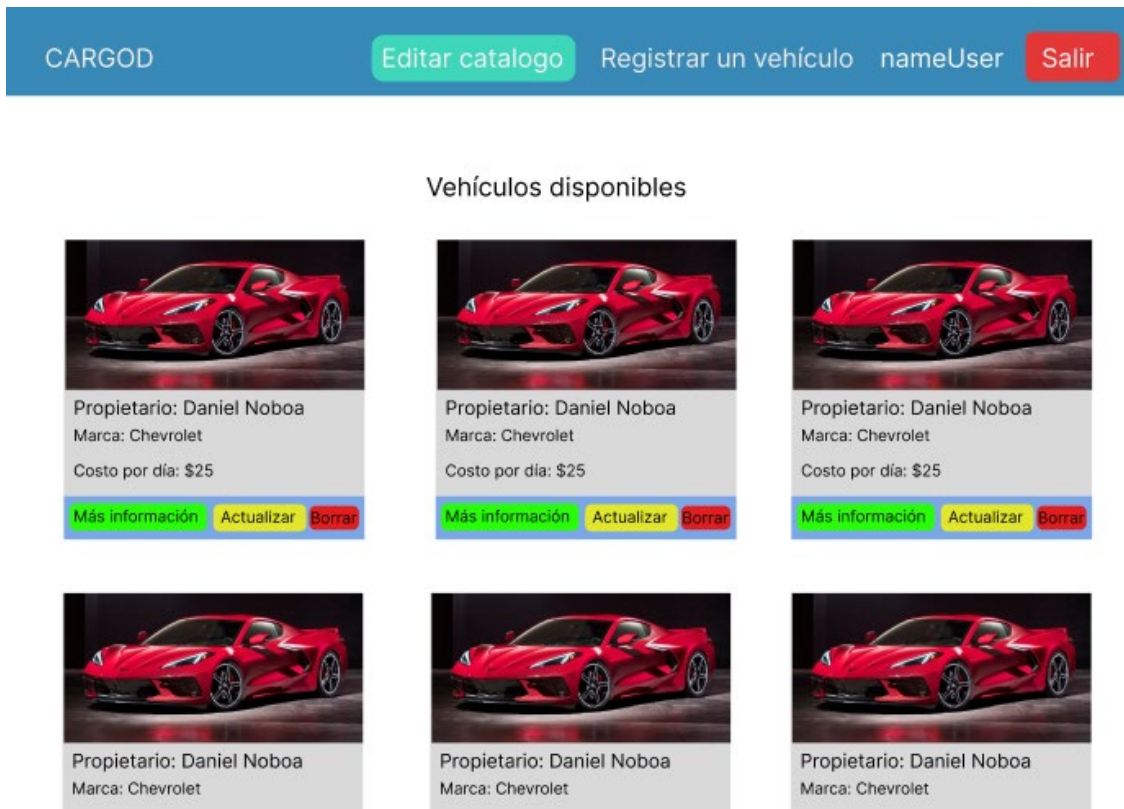


Figura 7.6: Editar catalogo figma



Figura 7.7: Actualización de vehículo figma

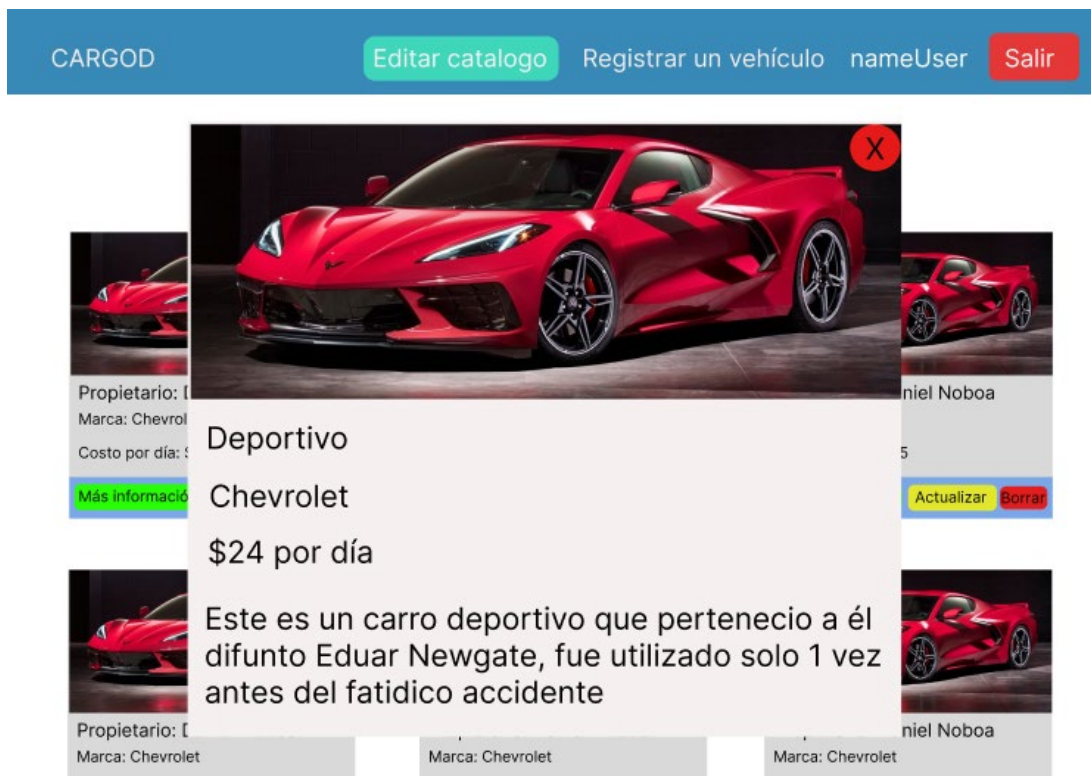


Figura 7.8: Información catalogo figma



Figura 7.9: Confirmación para borrar una publicación figma

BIENVENIDO

Con el rol Cliente tu puedes alquilar vehículos

Ingresar tus datos para poder registrarte

Nombre (0/30)
Ingresar tu nombre

Apellido (0/30)
Ingresar tu apellido

Dirección (0/60)
Ingresar tu dirección

Teléfono (0/10)
Ingresar tu teléfono

Correo electrónico (0/50)
Ingresar tu correo electrónico

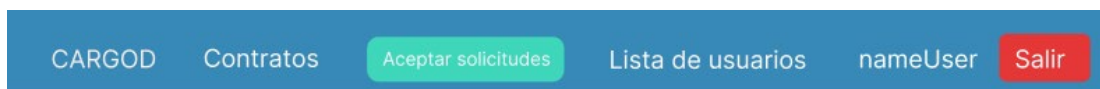
Contraseña (0/20)

Registrar

¿Ya tienes una cuenta? Dirígete al login para iniciar sesión [Iniciar sesión](#)



Figura 7.10: Registro cliente figma



Solicitudes de vehículos en alquiler




 <p>Propietario: Daniel Noboa Marca: Chevrolet Costo por día: \$25</p> <p>Más información Aceptar Rechazar</p>	 <p>Propietario: Ricardo Perez Marca: Chevrolet Costo por día: \$25</p> <p>Más información Aceptar Rechazar</p>	 <p>Propietario: Homero Simpson Marca: Chevrolet Costo por día: \$25</p> <p>Más información Aceptar Rechazar</p>
--	---	--

Figura 7.11: Aceptación de vehículos figma

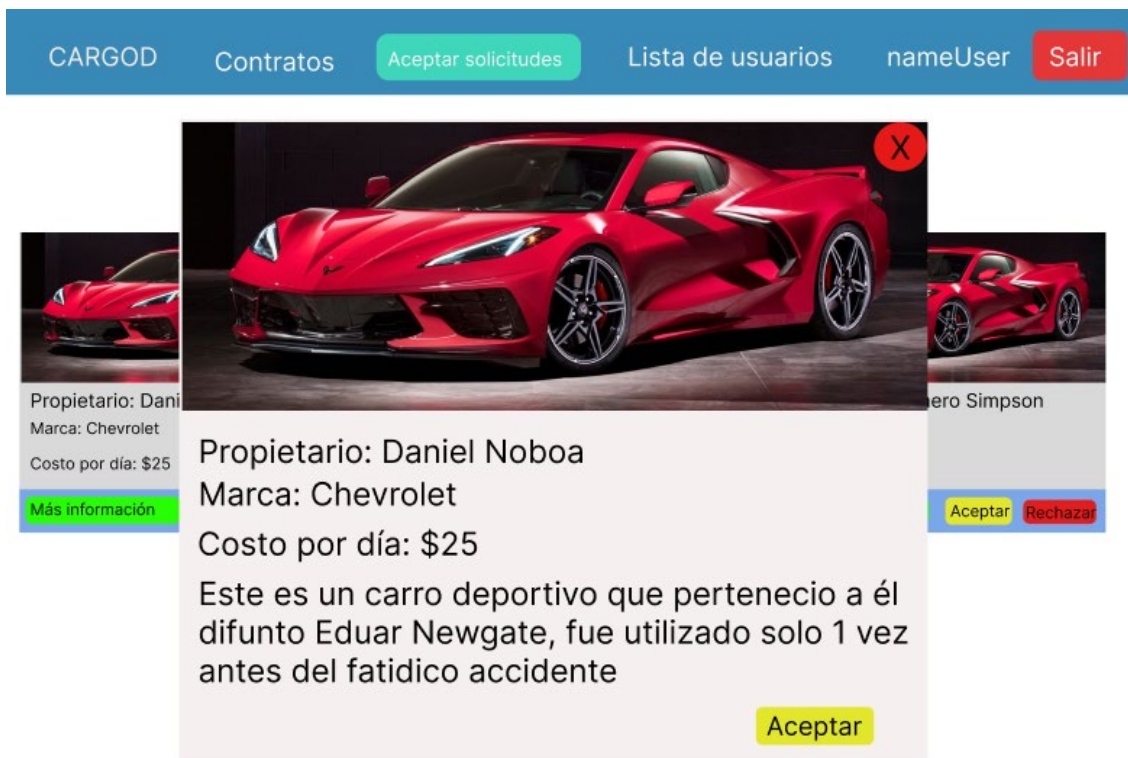


Figura 7.12: Detalles de vehículo administrador figma



Figura 7.13: Confirmación de aceptación de solicitud figma

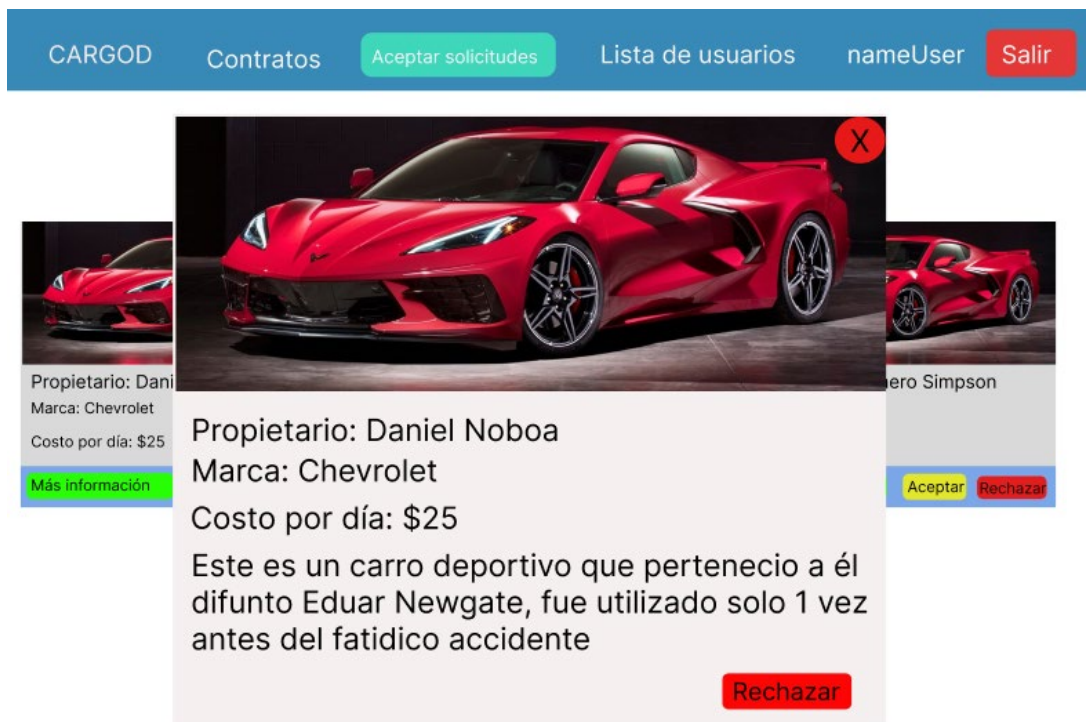


Figura 7.14: Detalles para rechazar una solicitud figma



Figura 7.15: Mensaje de rechazo figma

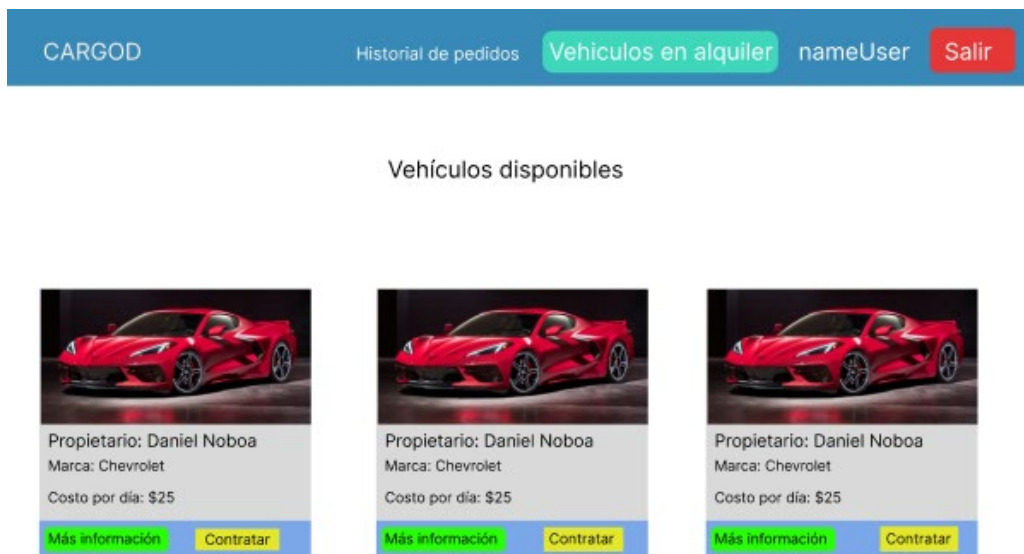


Figura 7.16: Vehículos disponibles para alquilar figma

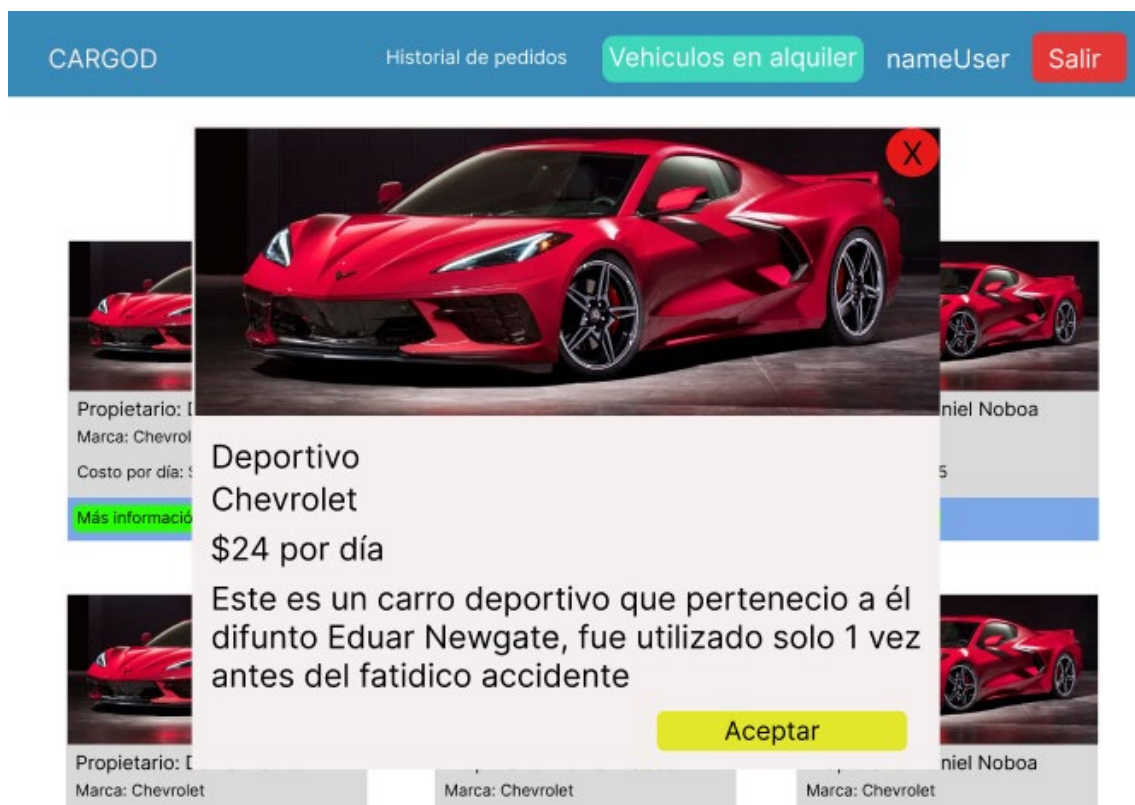


Figura 7.17: Aceptación de contrato figma

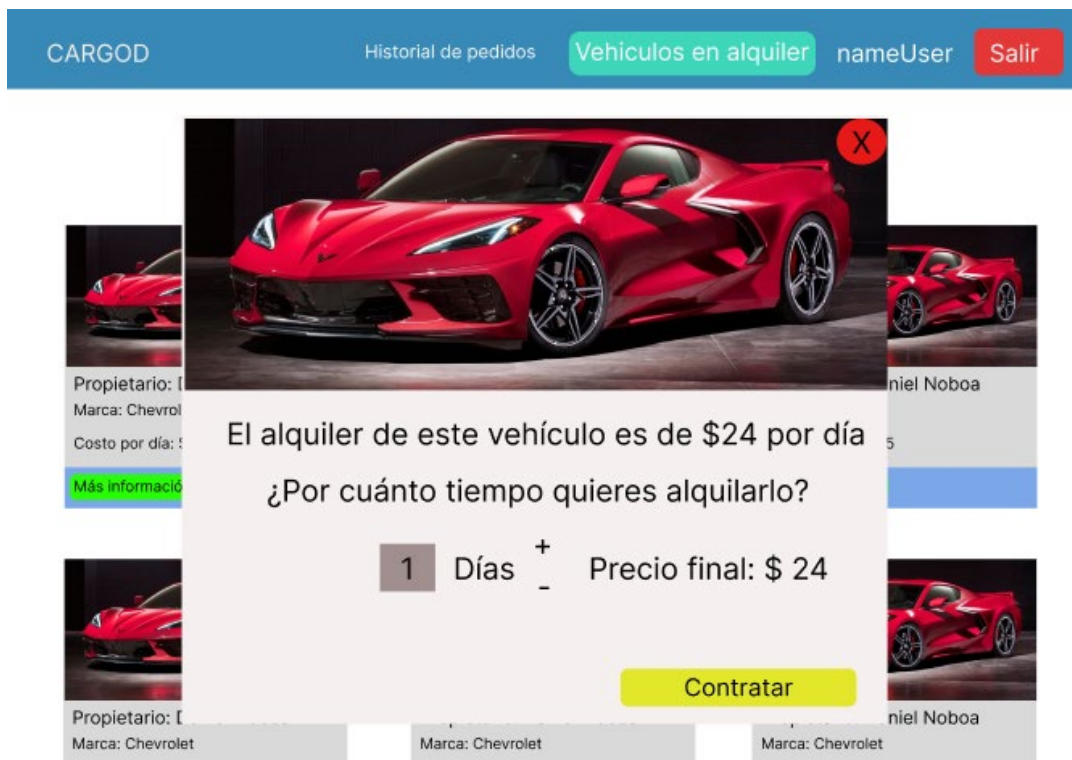


Figura 7.18: Cantidad de días contrato figma

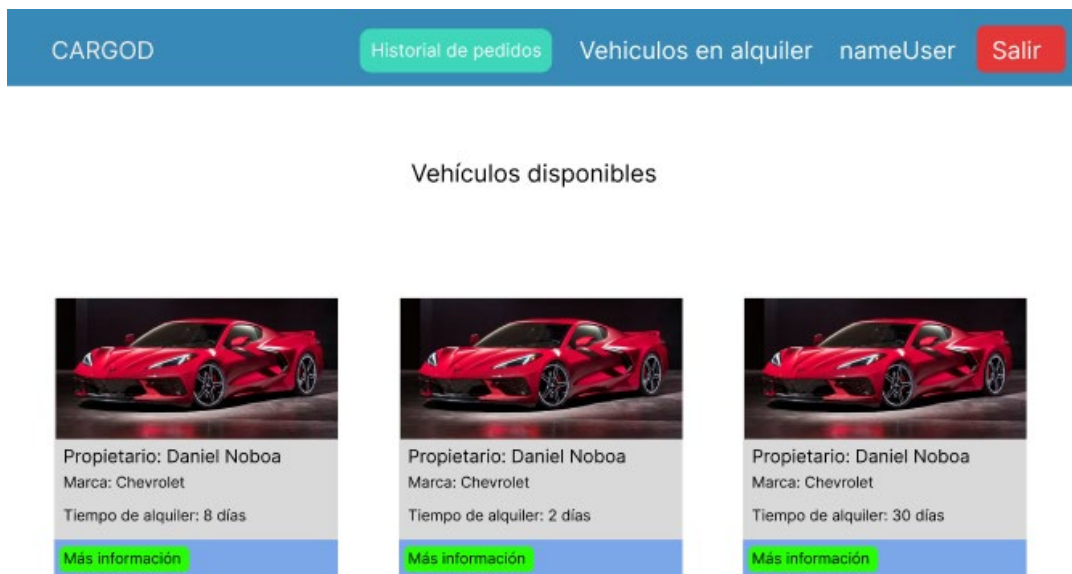


Figura 7.19: Historial de contratos figma

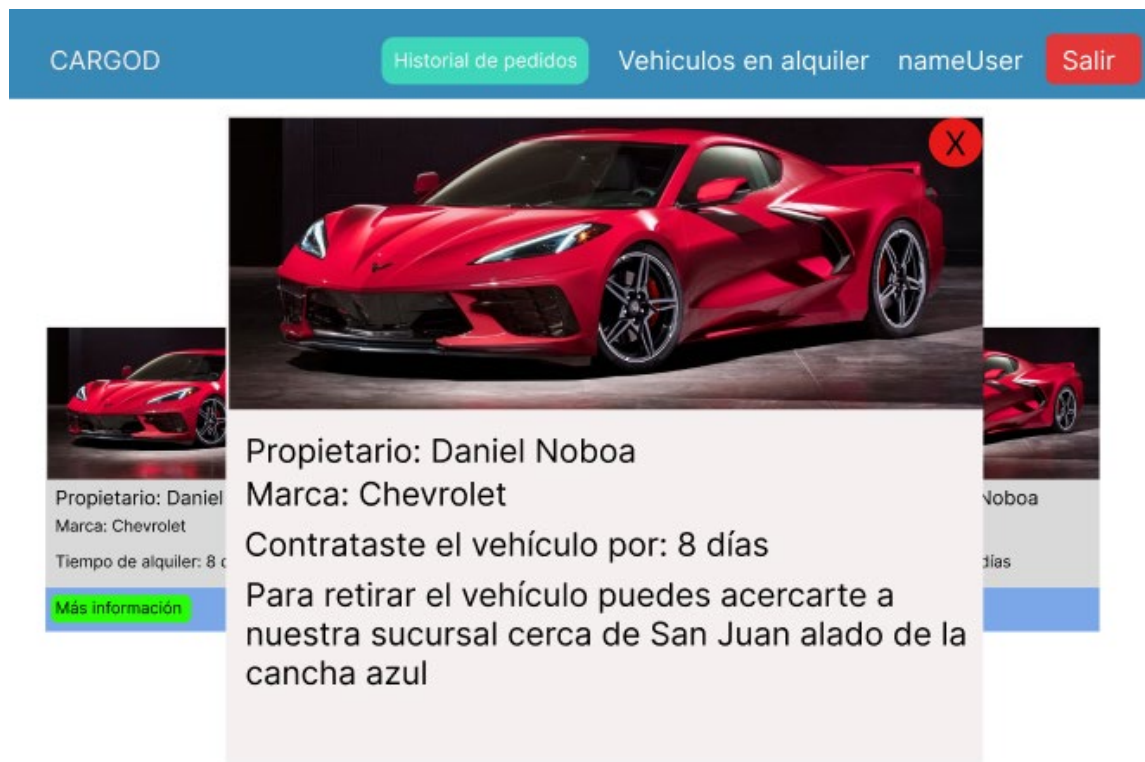


Figura 7.20: Detalles en el historial figma

En el siguiente enlace se puede ver el diseño de todos los prototipos mencionados:

https://www.figma.com/file/t5hg9mdE1KYNkCN5AANq2i/Alquiler_Veh%C3%ADculos?type=design&node-id=0%3A1&mode=design&t=AbYhDLkmQpnfz53z-1

Herramientas

Para que se pueda realizar la instalación de las herramientas se debe iniciar la descarga de Visual Studio Code en el siguiente enlace:

<https://code.visualstudio.com/download>

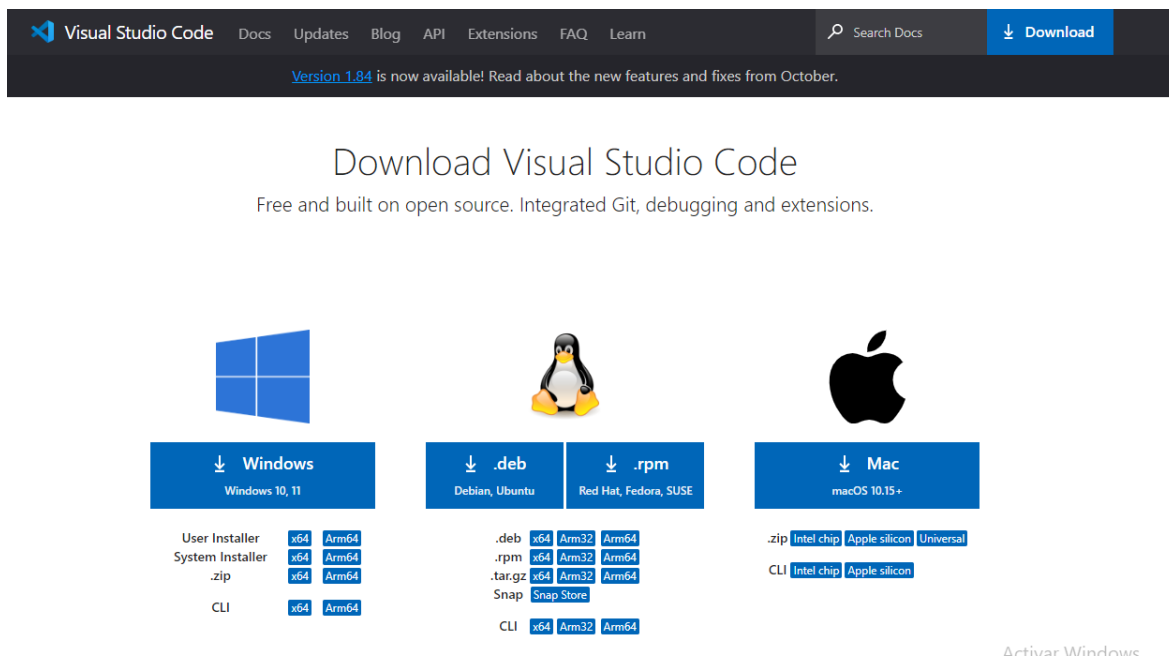


Figura 7.21: Descarga Visual Studio

Una vez que este instalado se procederá a hacer la siguiente descarga de Node.js a través del siguiente enlace: <https://nodejs.org/en/download>

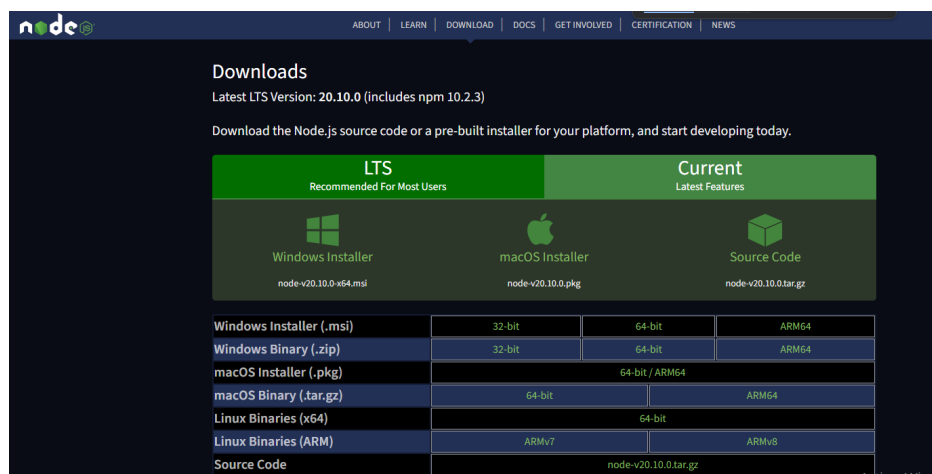
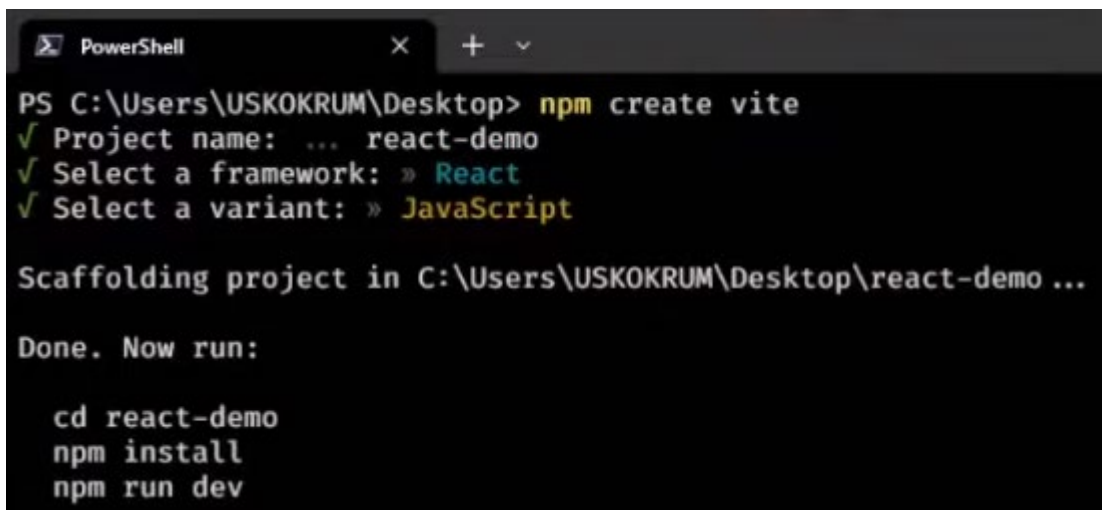


Figura 7.22: Descarga NodeJS

Una vez que nuestras herramientas estén instaladas se procede a inicializar el proyecto como se puede ver en la **Figura 7.23**



```
PowerShell
PS C:\Users\USKOKRUM\Desktop> npm create vite
✓ Project name: ... react-demo
✓ Select a framework: » React
✓ Select a variant: » JavaScript

Scaffolding project in C:\Users\USKOKRUM\Desktop\react-demo ...

Done. Now run:

  cd react-demo
  npm install
  npm run dev
```

Figura 7.23: Inicialización vite

Librerías principales

Tabla 7.13: Librerías

Librería	Comando de instalación	Descripción
Axios	npm install axios	Esta es una herramienta de compilación que permite tener una experiencia de desarrollo mucho más fluida y ágil para los proyectos
Cypress	npm install --save-dev cypress	Herramienta de testeo escrita en javascript, esta automatiza la comprobación de que un producto de software cumple con los requisitos iniciales y funciona de manera óptima
Tailwindcss	npm install tailwindcss npx tailwindcss init	Este es una herramienta para construir sitios web modernos y estilizados a diferencia de CSS este ofrece diseños prediseñados, tailwindcss está centrado en proporcionar utilidades de bajo nivel que permiten la creación de estilos personalizados de manera eficiente.

Pruebas de aceptación

En este apartado se presenta el complemento de pruebas de aceptación

Tabla 7.14: Prueba 1

Pruebas de aceptación	
Numero: 1	HU000
Nombre: Página de inicio	
Descripción: Como usuario desea poder ver una página principal de la empresa de renta de vehículos	
Parámetros de ejecución: Se presentará una pantalla de inicio en la cual se podrá ver de qué se trata la página.	
Entrega: La pantalla de inicio cuenta con una descripción sobre que trata la página.	
Resultado esperado: Al momento de ingresar a la página será lo primero que se le mostrará al usuario.	
Evaluación de la prueba: Prueba parcialmente satisfactoria, se recomienda actualizar la sensibilidad del proyecto.	

Tabla 7.15: Prueba 2

Pruebas de aceptación	
Numero: 2	HU001
Nombre: Registro propietario	
Descripción: Al momento de ingresar a la pantalla de inicio el usuario podrá ver un botón en la parte superior derecha el cual lo dirigirá al <i>login</i> , en el cual se mostrará un apartado en el que se le dará la opción de registrarse como propietario, el cual lo enviará a otra pestaña donde podrá registrarse.	
Parámetros de ejecución: El usuario una vez que se encuentre en el <i>login</i> deberá presionar el botón que diga: "Registrarse como propietario" para poder ser enviado a otra vista donde este podrá ingresar sus datos para registrarse	
Entrega: El <i>login</i> cuenta con un apartado para poder elegir si el usuario desea registrarse como propietario o como cliente.	

Resultado esperado: Al momento de hacer click en el botón este redirigirá al usuario a una nueva vista donde podrá registrarse.
Evaluación de la prueba: Prueba parcialmente satisfactoria, se recomienda actualizar la sensibilidad del proyecto.

Tabla 7.16: Prueba 3

Pruebas de aceptación	
Numero: 3	HU002
Nombre: Registro cliente	
Descripción: Al momento de ingresar a la pantalla de inicio el usuario podrá ver un botón en la parte superior derecha el cual lo dirigirá al <i>login</i> , en el cual se mostrará un apartado en el que se le dará la opción de registrarse como cliente, el cual lo enviará a otra pestaña donde podrá registrarse.	
Parámetros de ejecución: El usuario una vez que se encuentre en el <i>login</i> deberá presionar el botón que diga: "Registrarse como cliente" para poder ser enviado a otra vista donde este podrá ingresar sus datos para registrarse	
Entrega: El <i>login</i> cuenta con un apartado para poder elegir si el usuario desea registrarse como propietario o como cliente.	
Resultado esperado: Al momento de hacer click en el botón este redirigirá al usuario a una nueva vista donde podrá registrarse.	
Evaluación de la prueba: Prueba parcialmente satisfactoria, se recomienda actualizar la sensibilidad del proyecto.	

Tabla 7.17: Prueba 4

Pruebas de aceptación	
Numero: 4	HU003
Nombre: Inicio de sesión propietario	
Descripción: Una vez que se haya podido registrar el usuario este podrá ver que en la parte inferior de la vista existe un botón que dice iniciar sesión, este lo redirigirá al <i>login</i> donde podrá ingresar su correo electrónico y contraseña para poder iniciar sesión dentro de la página	

Parámetros de ejecución: El usuario deberá ingresar su correo electrónico y su contraseña en el apartado de <i>login</i> para poder entrar a la página.
Entrega: El <i>login</i> cuenta con dos <i>inputs</i> en los cuales se especifica donde debe ingresar el usuario su correo electrónico y su contraseña.
Resultado esperado: Al momento de ingresar el correo electrónico y la contraseña deberá redirigirse a una nueva vista dependiendo del rol del usuario.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 7.18: Prueba 5

Pruebas de aceptación	
Numero: 5	HU004
Nombre: Inicio de sesión cliente	
Descripción: Una vez que se haya podido registrar el usuario este podrá ver que en la parte inferior de la vista existe un botón que dice iniciar sesión, este lo redirigirá al <i>login</i> donde podrá ingresar su correo electrónico y contraseña para poder iniciar sesión dentro de la página	
Parámetros de ejecución: El usuario deberá ingresar su correo electrónico y su contraseña en el apartado de <i>login</i> para poder entrar a la página.	
Entrega: El <i>login</i> cuenta con dos <i>inputs</i> en los cuales se especifica dónde debe ingresar el usuario su correo electrónico y su contraseña.	
Resultado esperado: Al momento de ingresar el correo electrónico y la contraseña deberá redirigirse a una nueva vista dependiendo del rol del usuario.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 7.19: Prueba 6

Pruebas de aceptación	
Numero: 6	HU005
Nombre: Inicio de sesión administrador	
Descripción: El administrador no deberá registrarse en la página este ya contara con una cuenta que es proporcionada por el <i>backend</i> donde <i>directamente puede dirigirse al apartado de login</i> e iniciar sesión con sus respectivas credenciales	

Parámetros de ejecución: El usuario deberá ingresar su correo electrónico y su contraseña en el apartado de <i>login</i> para poder entrar a la página.
Entrega: El <i>login</i> cuenta con dos <i>inputs</i> en los cuales se especifica dónde debe ingresar el usuario su correo electrónico y su contraseña.
Resultado esperado: Al momento de ingresar el correo electrónico y la contraseña deberá redirigirse a una nueva vista dependiendo del rol del usuario.
Evaluación de la prueba: Prueba satisfactoria

Tabla 7.20: Prueba 7

Pruebas de aceptación	
Numero: 7	HU006
Nombre: Recuperación de contraseña	
Descripción: En caso de que alguno de los usuarios haya olvidado su contraseña, debajo de la entrada de datos para el correo y la contraseña se podrá ver un mensaje que dice: “¿Has olvidado tu contraseña”, donde estos podrán hacer click para recuperarla?	
Parámetros de ejecución: Para poder recuperar la contraseña estos deberán hacer click donde se muestra el mensaje esto los redirigirá a una nueva ventana donde se le pedirá que ingrese su correo electrónico para poder enviarle un correo en el cual los llevara a una nueva vista para poder colocar una nueva contraseña.	
Entrega: Un apartado llamado “¿Has olvidado tu contraseña”, para poder ingresar el correo electrónico con el cual se registró para poder recuperar su contraseña?	
Resultado esperado: Recuperar el inicio de sesión restableciendo la contraseña vía correo electrónico.	
Evaluación de la prueba: Prueba parcialmente satisfactoria, se recomienda actualizar la sensibilidad del proyecto y ajustar los campos	

Tabla 7.21: Prueba 8

Pruebas de aceptación	
Numero: 8	HU007
Nombre: Registrar un vehículo	
Descripción: Una vez que el propietario haya podido iniciar sesión, esta vera una barra de navegación donde deberá hacer click en el apartado de “Registrar un vehículo” esto desplegara un formulario en la parte inferior el cual deberá llenar para poder registrar su vehículo.	
Parámetros de ejecución: El usuario al ver la barra de navegación hará click en el apartado “Registrar un vehículo” para poder visualizar el formulario que deberá llenar para registrar su vehículo.	
Entrega: Una barra de navegación con 3 apartados “Registrar un vehículo”, “Nombre del usuario”, “salir”.	
Resultado esperado: Poder visualizar y enviar el formulario una vez se haya hecho click en el apartado de registrar un vehículo.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 7.22: Prueba 9

Pruebas de aceptación	
Numero: 9	HU008
Nombre: Aprobación vehículos	
Descripción: Una vez que el administrador haya podido iniciar sesión, este vera una barra de navegación donde deberá hacer click en el apartado de “Aceptar solicitudes”, esto desplegara todas las solicitudes que hayan sido creadas por los diferentes propietarios.	
Parámetros de ejecución: El usuario al ver la barra de navegación hará click en el apartado “Aceptar solicitudes” para poder visualizar todas las solicitudes de los diferentes propietarios.	
Entrega: Una barra de navegación con 3 apartados “Aceptar solicitudes”, “Nombre del usuario”, “salir”.	

Resultado esperado: Poder visualizar y aceptar o rechazar las diferentes solicitudes que existen.
Evaluación de la prueba: Prueba satisfactoria

Tabla 7.23: Prueba 10

Pruebas de aceptación	
Numero: 10	HU009
Nombre: Visualización vehículos	
Descripción: Una vez que el cliente haya podido iniciar sesión, este vera una barra de navegación donde deberá hacer click en el apartado de “Vehículos en alquiler”, este apartado desplegara toda la lista de vehículos que fueron aprobados por el administrador.	
Parámetros de ejecución: El usuario al ver la barra de navegación hará click en el apartado “Vehículos en alquiler” para poder visualizar todas las solicitudes que fueron aceptadas por parte del administrador.	
Entrega: Una barra de navegación con 4 apartados “Historial de vehículos”, “Vehículos en alquiler”, “Nombre del usuario”, “salir”.	
Resultado esperado: Poder visualizar y contratar las diferentes propuestas que existen.	
Evaluación de la prueba: Prueba parcialmente satisfactoria, se recomienda dar una visualización más pequeña de las imágenes o cards presentadas.	

Tabla 7.24: Prueba 11

Pruebas de aceptación	
Numero: 11	HU010
Nombre: Contratación de vehículo	
Descripción: Una vez que el cliente visualice todas las ofertas disponibles y escoja alguna de su interés este podrá dar click en el botón contratar para poder escoger por cuanto tiempo quiere adquirir este vehículo.	
Parámetros de ejecución: Cuando el cliente haga click en el botón de contratar iniciará el proceso para contratación.	

Entrega: Tarjetas en las cuales se puede ver la foto del vehículo en y detalles como cuánto cuesta el alquiler, las placas del vehículo y entre otros, también contara con 2 botones para seleccionar la contratación del mismo o una vista general.
Resultado esperado: Acceder al apartado de contratación donde podrá escoger el tiempo en el que va adquirir el vehículo.
Evaluación de la prueba: Prueba satisfactoria

Tabla 7.25: Prueba 12

Pruebas de aceptación	
Numero: 12	HU011
Nombre: Tiempo de contratación	
Descripción: Una vez que el cliente visualice todas las ofertas disponibles y escoja alguna de su interés este podrá dar click en el botón contratar para poder escoger por cuanto tiempo quiere adquirir este vehículo.	
Parámetros de ejecución: Cuando el cliente haga click en el botón de contratar iniciará el proceso para contratación.	
Entrega: Tarjetas en las cuales se puede ver la foto del vehículo en y detalles como cuánto cuesta el alquiler, las placas del vehículo y entre otros, también contara con 2 botones para seleccionar la contratación del mismo o una vista general.	
Resultado esperado: Acceder al apartado de contratación donde podrá escoger el tiempo en el que va adquirir el vehículo, donde podrá aumentar la cantidad de días dependiendo de su interés.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 7.26: Prueba 13

Pruebas de aceptación	
Numero: 13	HU012
Nombre: Historial de pedidos	

Descripción: Una vez que el cliente haya contratado algún vehículo este podrá ver su contrato en la parte de “Historial de vehículos”, donde le saldrán de igual forma en una lista todos los contratos que haya realizado este usuario.
Parámetros de ejecución: Cuando el cliente haga click en el apartado “Historial de vehículos” este podrá tener una vista general de sus contratos.
Entrega: Apartado general para visualizar los contratos adquiridos.
Resultado esperado: Al momento de hacer click en el “Historial de vehículos” este deberá visualizar todos sus contratos.
Evaluación de la prueba: Prueba parcialmente satisfactoria, se recomienda dar un cronometro para ver el tiempo que queda de alquiler

A continuación, se integra el link de la encuesta realizada:
<https://forms.gle/tLr1W62xhXHpZ9ew7>

Pruebas de accesibilidad

Una vez que se conocen los resultados que presenta la página web, tanto en el apartado de celulares como el de escritorio se procede a ver el porqué de estos resultados, empezando por la versión de escritorio.

En este apartado que se muestra en la **Figura 7.24** este indica algunas prácticas recomendadas para los usuarios con visión reducida quienes necesitan ampliar la pantalla para ver correctamente el contenido, dado que se menciona en el mensaje introductorio a accesibilidad esto es una detección automática del sistema y dentro de la aplicación web si se puede ampliar la imagen y esta se va adaptando a los diferentes tamaños abordando el tema responsive no presenta un verdadero problema para los usuarios de visión reducida.

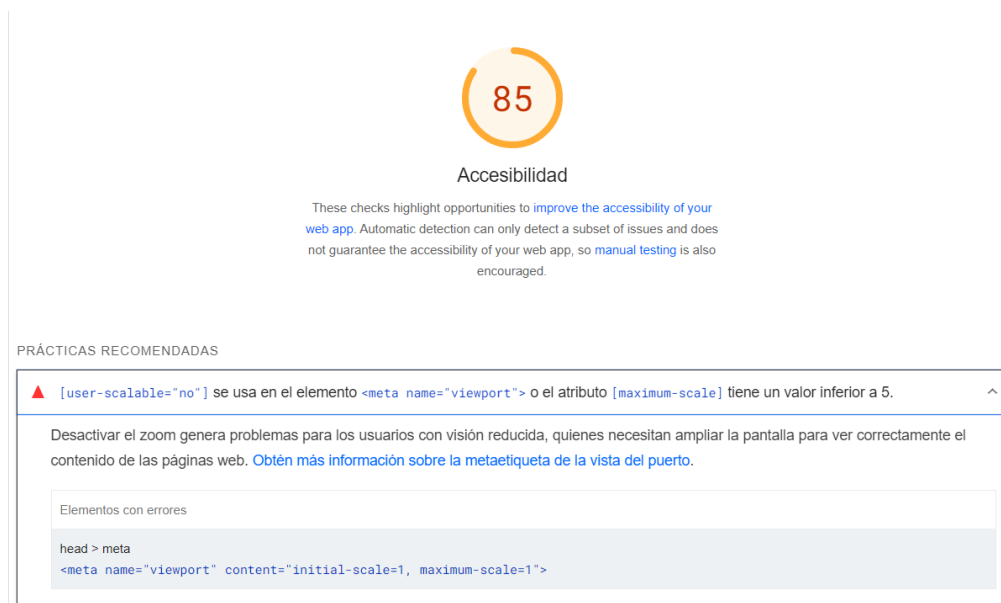


Figura 7.24: escritorio accesibilidad

El contraste presentado en la **Figura 7.25** se menciona que podría llegar a ser difícil de leer para muchos usuarios, con este apartado entiendo que simboliza a lo que contiene el botón, pero dado que el fondo es del mismo esquema de color que contiene las letras del botón se puede apreciar, tanto el botón como su contenido sin dificultad.

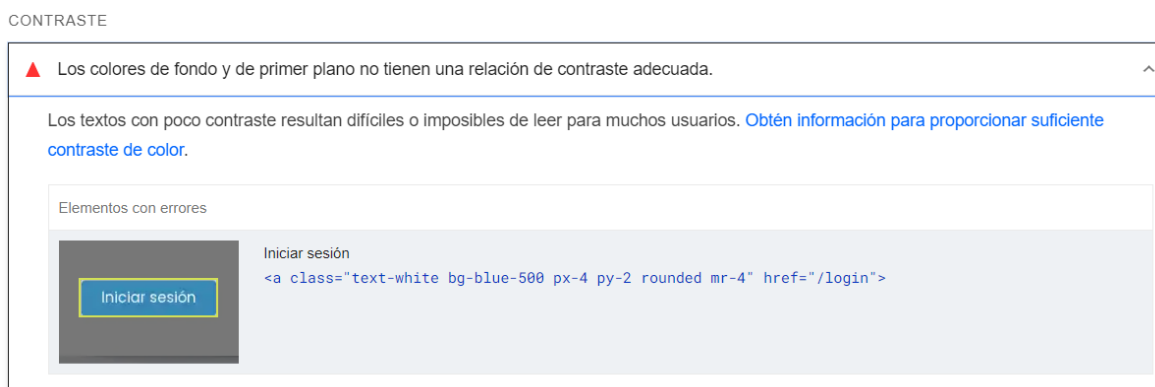


Figura 7.25: contraste escritorio

En el apartado de navegación como se puede observar en la **Figura 7.26** se presenta que se están utilizando mal las etiquetas de encabezado, este error ocurre ya que se está usando una etiqueta h2 en el código para el texto “Desarrolladores del proyecto” como se muestra en la imagen **Figura 7.27** por lo que a continuación

se debería utilizar una etiqueta h3, pero en su lugar se esta proporcionando una etiqueta h4 generando conflicto con la secuencia de los encabezados.

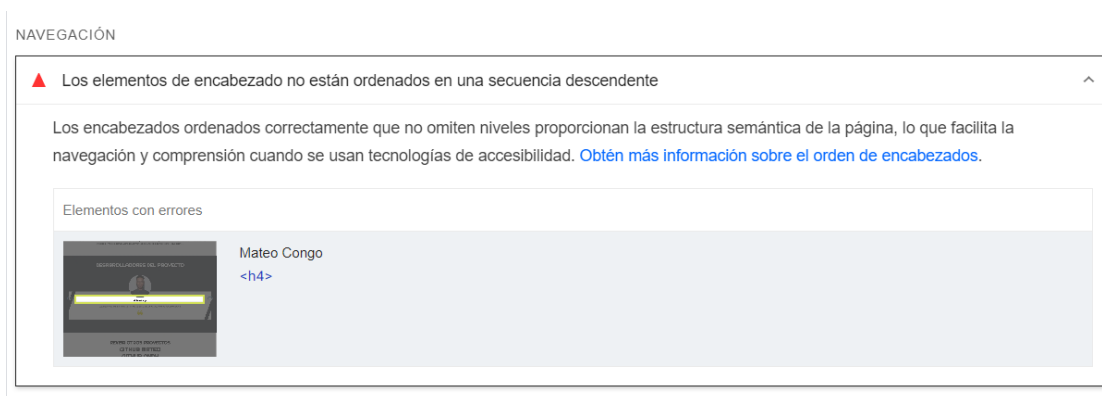


Figura 7.26: Navegación escritorio

```
<div className="titlepage">
  <h2>Desarrolladores del proyecto</h2>
</div>
</div>
```

Figura 7.27: desarrolladores

Ahora se procede con la versión de dispositivos móviles, como se observa en la **Figura 7.28** esta presenta los mismos valores respecto al apartado de accesibilidad con las mismas recomendaciones dichas ya para su versión en escritorio.

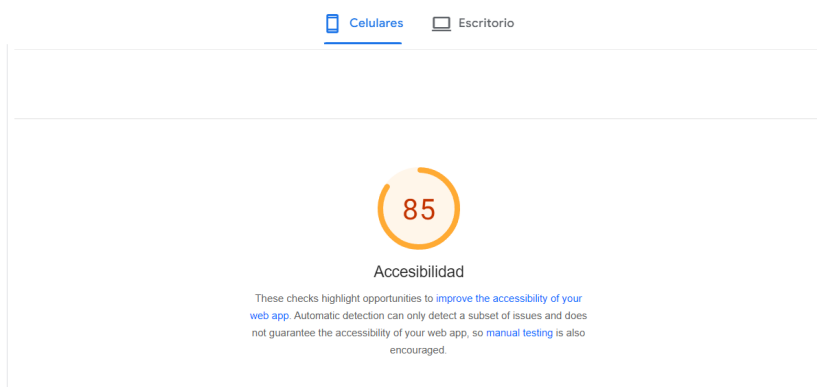


Figura 7.28: Test celulares

ANEXO III

Manual de usuario

En este apartado se puede encontrar con el enlace a un video, este video tiene la intención de informar el uso esperado para la página web, donde se abordan temas como: La creación de cuentas, tanto del propietario como el cliente, el uso del rol administrador, como crear un nuevo vehículo y como contratar un vehículo.

Para acceder a dicho video se deberá hacer click en el siguiente enlace.

- <https://youtu.be/lXywkaym-kw>

ANEXO IV

Manual de instalación

En este apartado se explica el proceso que se debe cumplir si es que de ser el caso otra persona necesite a acceder para utilizar este proyecto.

La aplicación se puede encontrar en el siguiente enlace:

- <https://cargod.netlify.app/>

Asistencia sobre el funcionamiento web

En caso de necesitar ayuda se debe comunicar con el administrador del sistema a través de esta dirección de correo electrónico mateo.congo@epn.edu.ec para que este le puede proporcionar asistencia.

El enlace del código fuente se lo puede encontrar en la siguiente dirección de GitHub:

- https://github.com/Einarr07/Renta_de_vehiculos-FRONTEND.git