



**UNIVERSIDAD SANTO TOMÁS**  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
T U N J A

Software multiplataforma para la empresa Crediautos Colombia

**PROPONENTE(S)**

Brayan Snader Sierra Riveros  
1049654940  
Código

**DIRECTOR**

Martha Susana Contreras Ortiz


Tunja  
Fecha de presentación (18, 12, 2024)

## CONTENIDO

<b>1. FICHA TÉCNICA DEL PROYECTO.....</b>	<b>4</b>
<b>2. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>5</b>
<b>3. JUSTIFICACIÓN.....</b>	<b>7</b>
<b>4. OBJETIVOS.....</b>	<b>8</b>
4.1. OBJETIVO GENERAL .....	8
4.2. OBJETIVOS ESPECÍFICOS.....	8
<b>5. MODELO DE DESARROLLO.....</b>	<b>9</b>
<b>6. DESARROLLO DEL PROYECTO .....</b>	<b>11</b>
<b>6.1. FASE 1. DEFINICIÓN DE REQUISITOS.....</b>	<b>11</b>
6.1.1. DESCRIPCIÓN GENERAL DEL SISTEMA .....	11
6.1.2. CATÁLOGO DE REQUISITOS .....	12
6.1.3. REQUISITOS FUNCIONALES .....	13
<b>6.2. FASE 2. SELECCIÓN DE TECNOLOGÍA .....</b>	<b>16</b>
<b>6.3. FASE 3. DISEÑO DEL SISTEMA .....</b>	<b>17</b>
6.3.1. DIAGRAMA DE ENTIDAD-RELACIÓN.....	17
6.3.2. DIAGRAMA DE SECUENCIA.....	19
6.3.3. DIAGRAMA DE CLASES .....	26
6.3.4. DISEÑO DEL MÓDULO ADMINISTRATIVO .....	29
6.3.5. DISEÑO DE USUARIO FINAL.....	38
6.3.6. DISEÑO DE APLICACIÓN MÓVIL.....	43
<b>6.4. FASE 4. DESARROLLO DE LA APLICACIÓN .....</b>	<b>50</b>
6.4.1. DESARROLLO DE LA BASE DE DATOS.....	50
6.4.2. CREACIÓN DE PROYECTO .....	51
6.4.3. CONEXIÓN CON FIREBASE .....	51
6.4.4. CREACIÓN DE SERVICIOS.....	52

6.4.5. CREACIÓN DEL MÓDULO ADMINISTRATIVO .....	54
6.4.6. CREACIÓN DEL MÓDULO DE USUARIO FINAL.....	59
6.4.7. CREACIÓN DE LA APLICACIÓN MÓVIL .....	68
<b>6.5. FASE 5. PRUEBAS.....</b>	<b>76</b>
<b><u>7. CONCLUSIONES Y ANÁLISIS DE RESULTADOS .....</u></b>	<b><u>83</u></b>
<b><u>8. REFERENCIAS.....</u></b>	<b><u>97</u></b>

## 1. FICHA TÉCNICA DEL PROYECTO

Título	Software multiplataforma para la empresa Crediautos Colombia
Nombre Estudiante	Brayan Snader Sierra Riveros
Documento estudiante	1049654940
Correo electrónico estudiante	brayan.sierra@usantoto.edu.co
Director	Martha Susana Contreras Ortiz
Lugar de ejecución del proyecto	Crediautos
Duración	6 meses
Palabras claves	Ventas y compras de automóvil, metodología SCRUM, Aplicación Web y Aplicación móvil.
Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente (incluir nombres y firmas de estudiantes y director).	
	
Firma del autor Brayan Snader Sierra Riveros	
Firma del director Martha Susana Contreras Ortiz	

## 2. PLANTEAMIENTO DEL PROBLEMA

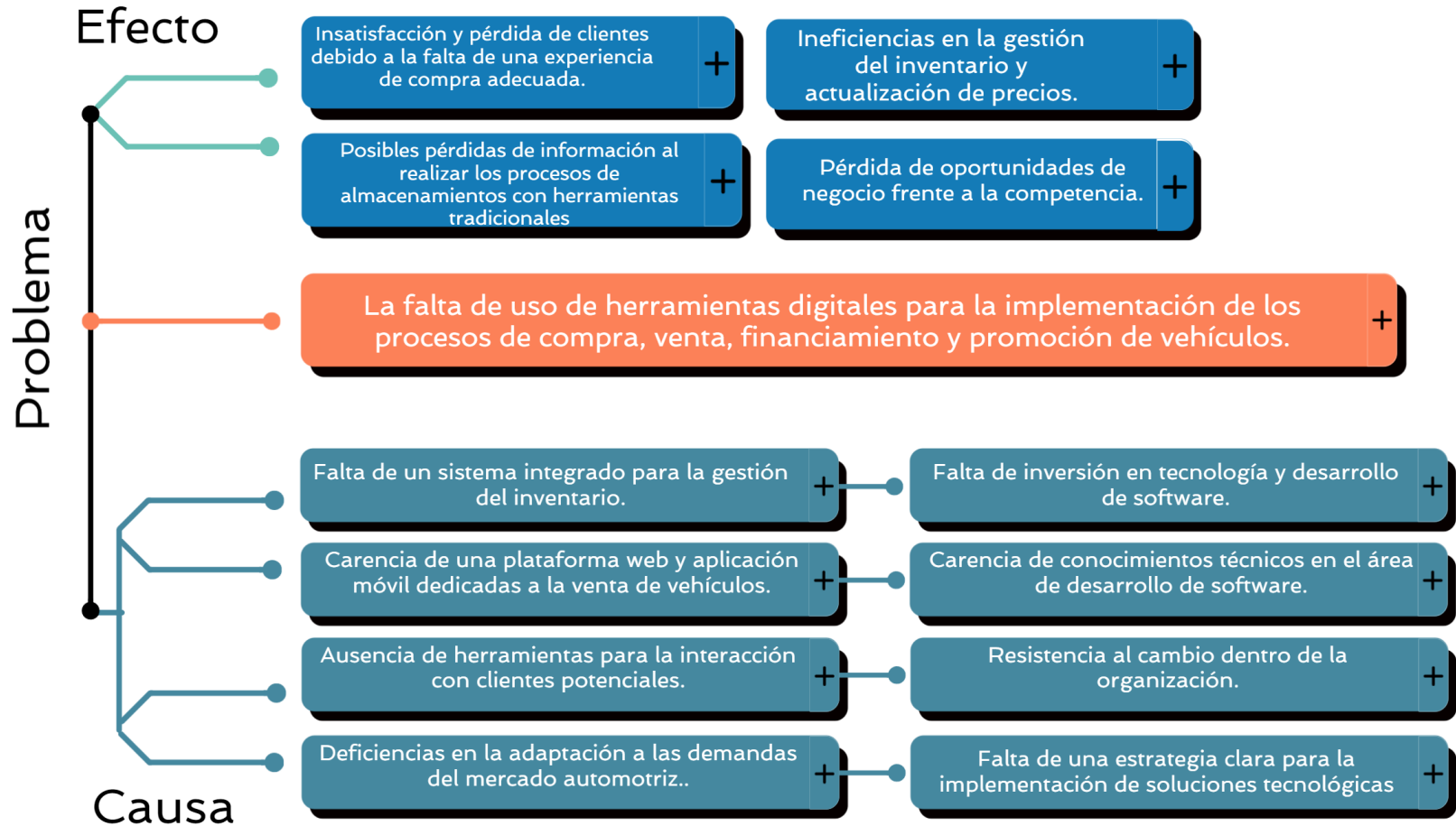
Crediautos Colombia es una empresa especializada en la compra y venta de automóviles, así como en la comercialización de artículos automovilísticos de alta calidad. Con más de 10 años de experiencia en la industria automotriz, se ha consolidado como un referente en la ciudad de Yopal.

La empresa Crediautos Colombia debe enfrentar un importante desafío en sus operaciones de venta de vehículos y artículos automovilísticos. El proceso actual carece de una plataforma integral que permita que la compañía cuente con facilidades para gestionar el inventario, como también, una interacción rápida con sus clientes potenciales. La falta de un sistema integrado con recursos específicos para esa comercialización provoca ineficiencia en la administración del stock, dificulta la comunicación con los clientes y limita el poder de acompañamiento en el proceso de venta.

La falta de una solución tecnológica adecuada también limita la capacidad de Crediautos para adaptarse a las demandas cambiantes del mercado y competir eficazmente con otras empresas del mismo sector que ya han implementado soluciones similares. Esto significa una pérdida de oportunidades de negocio, menor satisfacción del cliente y una disminución en la rentabilidad a largo plazo.

Asimismo, Crediautos podría recopilar datos valiosos de los clientes si contara con aplicaciones web y móvil. Esto es importante ya que este tipo de información permite cambiar y adaptar los productos de Crediautos, haciendo que las ofertas y promociones personalizadas sean más atractivas para los clientes, como también puede utilizar los datos recopilados para mejorar sus productos y/o servicios, que satisfaga las necesidades de sus clientes. En la Figura 1 se puede observar un árbol de problemas donde se plantea la problemática de una forma visual.

Figura 1 Árbol de problemas



Fuente: Autor

### 3. JUSTIFICACIÓN

El uso de herramientas de Tecnologías de la Información y la Comunicación (TIC) en la digitalización de procesos en la franquicia Crediautos Colombia necesita tener un software multiplataforma para poder competir en el mercado y atraer a nuevos clientes. Por ende, las razones por las que Crediautos debe considerar tener una presencia en línea son:

- ✓ Mayor alcance: el software multiplataforma ayudaría Crediautos a llegar a una audiencia más amplia y diversa. La mayoría de las personas acceden a Internet y utilizan sus dispositivos móviles para buscar información y realizar compras. Una presencia en línea significa que Crediautos puede conectarse con diferentes personas, ampliar su alcance y generar más ventas.
- ✓ Proporcionar comodidad: la gente quiere comodidad y confort en su vida diaria, y esto incluye en la compra de un automóvil. Las aplicaciones móviles y los sitios web hacen que el proceso de compra sea más fácil y conveniente para sus clientes. Las aplicaciones móviles permiten a los clientes ver información detallada del vehículo, comparar vehículos y comprar en línea sin tener que ir a una tienda. El sitio web también permite a los clientes encontrar información detallada sobre el vehículo que les interesa o comunicarse con el personal de ventas si tienen preguntas.
- ✓ Proporcionar información precisa: los clientes pueden realizar una investigación detallada sobre sus automóviles favoritos antes de tomar una decisión de compra, lo que ayuda a aumentar la confianza en la marca y las ventas.
- ✓ Experiencia de usuario mejorada: el software multiplataforma mejora la experiencia del usuario al proporcionar una interfaz de usuario simple y accesible. Los clientes pueden buscar vehículos y obtener más información sobre opciones de financiamiento sin tener que visitar una tienda física. Además, las aplicaciones móviles pueden informar a los clientes sobre ofertas y promociones especiales, lo que puede atraer clientes y aumentar la fidelidad a Crediautos.

## 4. OBJETIVOS

### 4.1. Objetivo General

Desarrollar un software multiplataforma web y Android, para la venta de vehículos y artículos automovilísticos en la empresa Crediautos Colombia, utilizando UML como lenguaje para modelar el sistema y MVC como patrón de arquitectura de software.

### 4.2. Objetivos específicos

A continuación, se presentan los objetivos específicos.

Tabla 1. Objetivos específicos

Nro.	Objetivo específico
1	Hacer el levantamiento de requerimientos, para poder identificar las necesidades que tiene la empresa Crediautos a través de la observación y las entrevistas realizadas al personal de la empresa.
2	Diseñar la plataforma que cumpla con los requerimientos establecidos en el objetivo anterior, usando la metodología de prototipado.
3	Desarrollar una plataforma web y una aplicación móvil que cumplan con los requerimientos establecidos por la empresa Crediautos, implementando las operaciones CRUD para la gestión de datos, utilizando el framework Angular para la versión web y Flutter para la versión móvil en Android.
4	Realizar un plan de evaluación del software para verificar el cumplimiento de los requerimientos a través de pruebas funcionales.

Fuente: Autor

## 5. MODELO DE DESARROLLO

Para este proyecto, se ha elegido el marco de trabajo SCRUM por sus múltiples ventajas, tales como la flexibilidad para adaptarse a cambios en los requisitos del proyecto mediante ciclos de desarrollo cortos y permitir ajustar el backlog de producto, priorizando el trabajo en función del valor entregado, los elementos de mayor valor se desarrollan primero, lo que permite a los equipos brindar avances del proyecto al cliente de manera temprana y constante.

SCRUM fomenta la colaboración y la comunicación constante entre los miembros del equipo de desarrollo, lo que ayuda a evitar problemas y aumentar la calidad del producto, este modelo ofrece una mayor transparencia y visibilidad del progreso del proyecto a través de reuniones diarias, lo que permite a los equipos de desarrollo tener un mayor control y predictibilidad sobre el proceso de desarrollo del software, esto se logra a través de la planificación y seguimiento cuidadoso de los Sprints y de la iteración constante en el proceso de desarrollo (Gallego).

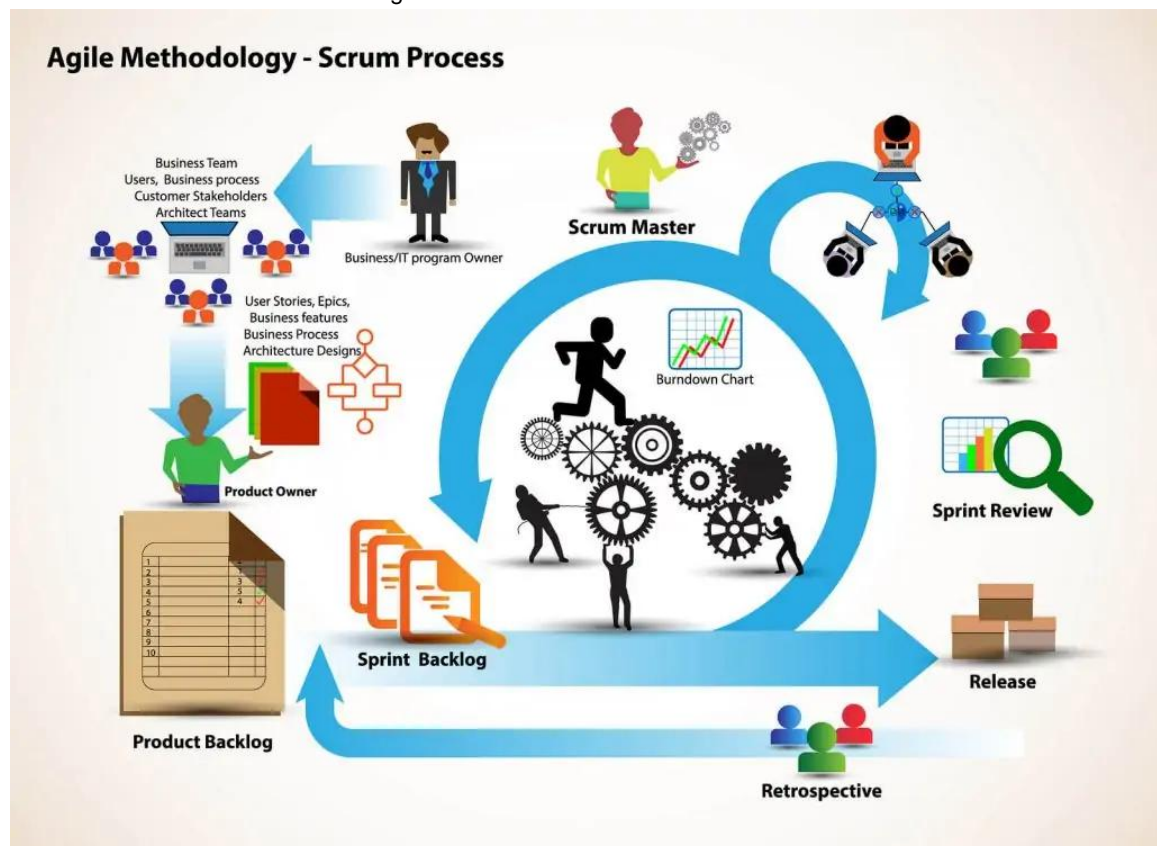
SCRUM se utiliza como una metodología de desarrollo de software porque ayuda a los equipos a trabajar de manera más eficiente y a entregar software de mayor calidad en menos tiempo. Además, SCRUM fomenta la mejora continua y la retroalimentación constante, lo que ayuda a los equipos a identificar y solucionar problemas de manera proactiva (Sonia I. Mariño, 2014).

Este modelo de desarrollo se divide en varias fases, como se muestra en la Figura 2, que facilitan el planteamiento del desarrollo y pruebas del aplicativo multiplataforma. Las principales fases incluyen:

- ✓ Planificación del Sprint: donde el equipo SCRUM define los objetivos del Sprint, analiza el backlog del producto y planifica las tareas necesarias para completarlo.
- ✓ Sprint: fase de desarrollo real, donde el equipo trabaja en las tareas planificadas durante un período de tiempo fijo.
- ✓ Reuniones diarias: sesiones diarias durante el Sprint para revisar el progreso e identificar posibles problemas.

- ✓ Revisión del Sprint: una vez que se ha completado el Sprint, el equipo SCRUM se reúne para revisar y demostrar el trabajo completado al Product Owner y al resto del equipo.
- ✓ Retrospectiva del Sprint: en esta fase, el equipo SCRUM analiza su rendimiento durante el Sprint y determina cómo puede mejorar en el futuro. El equipo también discute lo que salió bien y lo que salió mal durante el Sprint y toma medidas para mejorar en el siguiente Sprint.

Figura 2 Modelo SCRUM



Fuente: (Nimble Humanize Work, 2021)

## 6. DESARROLLO DEL PROYECTO

Para el desarrollo de este proyecto se contemplaron 6 fases secuenciales que permitieron completar el proyecto de manera exitosa. Estas fases sirvieron como guía y permiten organizar y estructurar el trabajo de forma eficiente.

### 6.1. Fase 1. Definición de requisitos

En esta fase se enfoca en la planificación y definición de requerimientos del proyecto, por ende, durante esta fase se realizaron actividades como la recopilación de información, el análisis de los requisitos y la definición de los alcances del proyecto. Para esto se realizaron reuniones tanto presenciales como virtuales con el gerente y el administrador de inventarios de la empresa crediautos, donde se exponía las dificultades que tenían y los procesos que se podían automatizar a través del aplicativo.

#### 6.1.1. Descripción general del sistema

Teniendo en cuenta las entrevistas realizadas en el levantamiento de requisitos de se pudo conocer mejor las necesidades del cliente en este caso la empresa crediautos y definir un alcance para el proyecto.

##### 6.1.1.1 Descripción de la situación actual

Crediautos utilizaba listas e inventarios en papel para el control de número de automóviles y artículos automovilísticos vendidos y comprados, como también el control y seguimiento de satisfacción de los clientes. Además, la empresa no contaba con una página web, lo cual ocasionaba presencia nula en internet.

##### 6.1.1.2 Características principales

Con el desarrollo de este proyecto se quiere que Crediautos pueda administrar su inventario de una forma más funcional, por ende, el software necesita un módulo administrativo y otro dirigido al cliente. Para esto, se deben tomar en cuenta diferentes tipos de roles, tanto en el módulo administrativo como en el módulo de usuario final.

##### 6.1.1.3 Entorno del sistema

El entorno de este Software es en su mayoría independiente ya que para su correcto funcionamiento no se necesita de ningún sistema mayor, excepto en la pasarela de pago que se quiere implementar.

#### 6.1.1.4 Restricciones

Las restricciones que tiene este software contemplan que para el lado móvil solo se incluyen los Smartphones cuyo sistema operativo sea Android. El módulo administrativo debe ser independiente del módulo de usuario final y solo se accederá a este por medio de plataforma web. Por otro lado, el módulo de usuario final estará disponible al público sin tener que iniciar sesión, a menos que un usuario quiera comprar y/o vender un producto.

#### 6.1.2. Catálogo de requisitos

A continuación, se muestra el catálogo completo de requisitos del software teniendo en cuenta que los tipos de requisitos son:

- F: Requisito Funcional
- NF: Requisito No Funcional
- TC: Requisito Técnico

Tabla 2. Catálogo de requerimientos

Id. REQ	Descripción corta	Tipo (F, NF, TC)
REQ-1	El software tiene que ser multiplataforma	TC
REQ-2	El software por el lado móvil solo es para Android	TC
REQ-3	El módulo administrativo es independiente de los otros módulos	F
REQ-4	El módulo administrativo solo es para plataforma web	F
REQ-5	No es necesario iniciar sesión para poder ver los productos en el módulo de usuario final	F
REQ-6	Para comprar y/o vender un producto el usuario tiene que iniciar sesión en el módulo de usuario final	F
REQ-7	Para el módulo administrativo se necesita dos roles "administrador" y "vendedor"	F
REQ-8	El rol de administrador tiene todos los permisos	F
REQ-9	El rol de vendedor solo puede ver usuarios que crearon una cuenta y poder editarlos sin cambiar roles	F
REQ-10	El rol vendedor puede publicar nuevos productos	F
REQ-11	El rol vendedor puede validar a los usuarios nuevos para saber si son reales	F
REQ-12	El software debe tener una pasarela de pago para los productos de menor precio	F

Id. REQ	Descripción corta	Tipo (F, NF, TC)
<b>REQ-13</b>	El software debe poder filtrar los autos por: Tipo de automóvil, Marca y Modelo	<b>F</b>
<b>REQ-14</b>	Los automóviles deben tener fotos, descripción y características	<b>F</b>
<b>REQ-15</b>	Los usuarios verificados pueden publicar vehículos y artículos automovilísticos	<b>F</b>
<b>REQ-16</b>	Cuando un usuario publica un vehículo o artículo no se debe mostrar al público en general hasta que sea validado por un administrador o un vendedor en el módulo administrativo	<b>F</b>

Fuente: Autor

### 6.1.3. Requisitos funcionales

Teniendo en cuenta los requisitos funcionales de la Tabla 2 se procede a describir mejor cada requisito funcional como se muestran de la Tabla 3 a la Tabla 16.

Tabla 3. Requisito REQ-3

<b>Id. REQ</b>	REQ-3
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	El módulo administrativo es independiente de los otros módulos, por ende, para este módulo debemos crear otra aplicación aparte del módulo administrador.

Fuente: Autor

Tabla 4. Requisito REQ-4

<b>Id. REQ</b>	REQ-4
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Media/Deseado
<b>Descripción del requisito</b>	El módulo administrativo solo es para plataforma web con el fin que se pueda acceder desde cualquier dispositivo con internet sin tener que instalar programas.

Fuente: Autor

Tabla 5. Requisito REQ-5

<b>Id. REQ</b>	REQ-5
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	No es necesario iniciar sesión para poder ver los productos en el módulo de usuario final, para que los usuarios finales puedan entrar y cotizar o comparar precios sin mucha dificultad.

Fuente: Autor

Tabla 6. Requisito REQ-6

<b>Id. REQ</b>	REQ-6
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	Para comprar y/o vender un producto el usuario tiene que iniciar sesión en el módulo de usuario final, para tener más control y evitar estafas o productos no adecuados al software.

Fuente: Autor

Tabla 7. Requisito REQ-7

<b>Id. REQ</b>	REQ-7
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	Para el módulo administrativo se necesita dos roles “administrador” y “vendedor”, porque la empresa Crediautos tiene vendedores que pueden modificar el inventario.

Fuente: Autor

Tabla 8. Requisito REQ-8

<b>Id. REQ</b>	REQ-8
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	El rol de administrador tiene todos los permisos, ya que el administrador es el único quien puede modificar los permisos de usuarios dentro del software, como también crear nuevos usuarios.

Fuente: Autor

Tabla 9. Requisito REQ-9

<b>Id. REQ</b>	REQ-9
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	El rol de vendedor solo puede ver usuarios que crearon una cuenta y poder editarlos sin cambiar roles como también puede validar dichos usuarios y validar los productos o publicaciones que un usuario pueda llegar hacer.

Fuente: Autor

Tabla 10. Requisito REQ-10

<b>Id. REQ</b>	REQ-10
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	El rol vendedor puede publicar nuevos productos y vehículos que se tengan en inventario como también puede activar o desactivar publicaciones.

Fuente: Autor

Tabla 11. Requisito REQ-11

<b>Id. REQ</b>	REQ-11
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial

<b>Descripción del requisito</b>	El rol vendedor puede validar o deshabilitar a los usuarios nuevos para saber si son reales, con el fin de evitar fraudes.
----------------------------------	--

Fuente: Autor

Tabla 12. Requisito REQ-12

<b>Id. REQ</b>	REQ-12
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Baja/ Opcional
<b>Descripción del requisito</b>	El software puede tener una pasarela de pago para los productos de menor precio, es decir para los artículos automovilísticos.

Fuente: Autor

Tabla 13. Requisito REQ-13

<b>Id. REQ</b>	REQ-13
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	El software debe poder filtrar los autos por: Tipo de automóvil, Marca y Modelo, para facilitar la búsqueda de los vehículos por parte de los usuarios finales.

Fuente: Autor

Tabla 14. Requisito REQ-14

<b>Id. REQ</b>	REQ-14
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Media/Deseado
<b>Descripción del requisito</b>	Los automóviles deben tener fotos, descripción y características de todos los accesorios que lo componen.

Fuente: Autor

Tabla 15. Requisito REQ-15

<b>Id. REQ</b>	REQ-15
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	Los usuarios verificados pueden publicar vehículos y artículos automovilísticos, como también comprar dichos productos.

Fuente: Autor

Tabla 16. Requisito REQ-16

<b>Id. REQ</b>	REQ-16
<b>Fuente del requisito</b>	Crediautos
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción del requisito</b>	Cuando un usuario publica un vehículo o artículo no se debe mostrar al público en general hasta que sea validado por un administrador o un vendedor en el módulo administrativo.

Fuente: Autor

Con esta descripción más detallada de los requisitos funcionales se obtiene una idea más específica y centrada de las necesidades de la empresa Crediautos.

## 6.2. Fase 2. Selección de tecnología

Tabla 17. Fase 2

Tecnología	¿Para qué se utilizará?	¿Por qué se utilizará?
Visual Studio Code	Este editor de código es el que vamos a utilizar para desarrollar el proyecto	Es un entorno de desarrollo con muchas ventajas como su velocidad, versatilidad y compatibilidad con muchos lenguajes de programación, tiene una gran comunidad activa, por ende, tiene una robusta documentación.
TypeScript	Es el lenguaje de programación que se utilizara para la codificación del proyecto	Este lenguaje de programación es compatible con angular, ya que se puede utilizar tanto en el frontend como en el backend, por esta razón su comunidad muy activa y cada vez es más numerosa
Bootstrap	Se utilizará para el diseño de las plataformas web.	Bootstrap simplifica el trabajo en el tema de diseño ya que tiene un sistema de cuadrículas y componentes adaptativos con las cuales facilita el responsive
Angular	Es el framework que se maneja para el desarrollo de las plataformas web.	Es un framework muy popular y robusto con una gran documentación, aparte tiene una estructura MVC que hace que las aplicaciones sean más sencillas de crear, también se puede reutilizar código ya que proporciona componentes que pueden ser llamados desde otros componentes o módulos.
Flutter	Es el framework que se maneja para el desarrollo de la aplicación móvil.	Flutter nos facilita el desarrollo de prototipos ya que tiene una gran cantidad de widget y herramientas ya integradas, aparte de esto las aplicaciones hechas con Flutter son nativas, en el tema de diseño tiene una gran ventaja al crear interfaces de usuarios personalizables y creativas, en el tema de rendimiento Flutter es muy eficaz al momento de consumir los recursos de un dispositivo ya que el código es nativo.
Firebase	Es el motor de datos para este proyecto	Es una base de datos no relacional (NoSQL) esto nos permite trabajar con una estructura flexible y escalable, también se puede obtener datos en tiempo real haciendo que sus consultas sean eficaces, también nos facilitara el tema de inicio de sesión por medio de Google o por Correo
ePayco	Es la pasarela de pago	Con respecto a epayco tiene una fuerte documentación para poder consumir su API REST y es compatible con las tecnologías previamente dichas.

Fuente: Autor

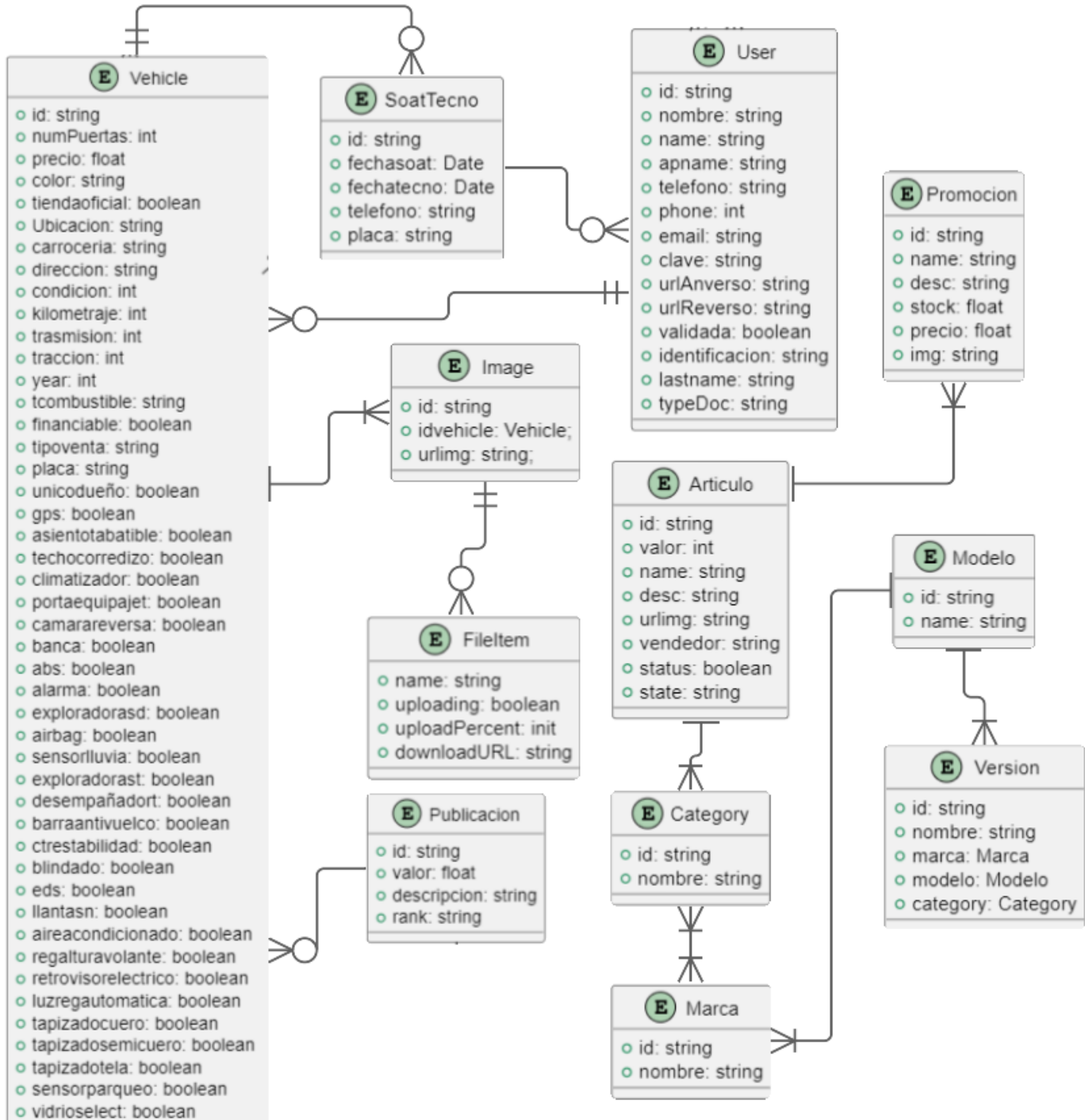
### **6.3. Fase 3. Diseño del sistema.**

En esta fase se comenzó con el diseño tanto visual como de arquitectura del software para eso se realizaron diferentes diagramas como los que se tienen a continuación.

#### **6.3.1. Diagrama de entidad-relación**

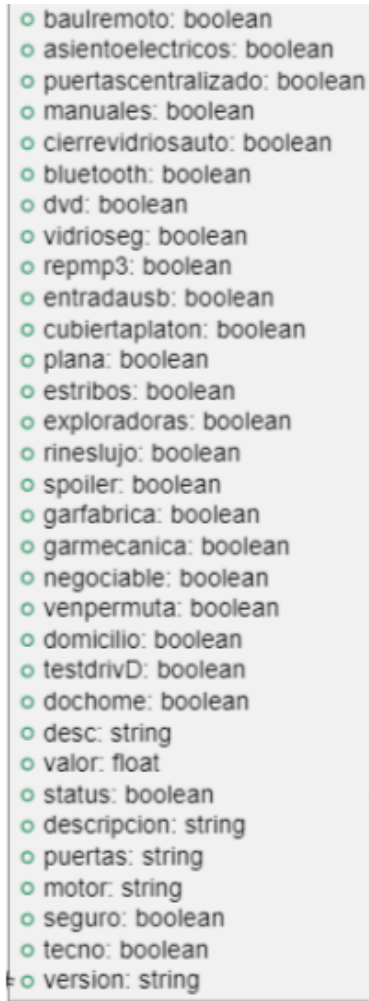
En este diagrama se puede observar las relaciones que tienen cada entidad dentro del sistema.

Figura 3 Diagrama entidad-relación



Fuente: Autor

Figura 4 Fragmento del diagrama entidad-relación



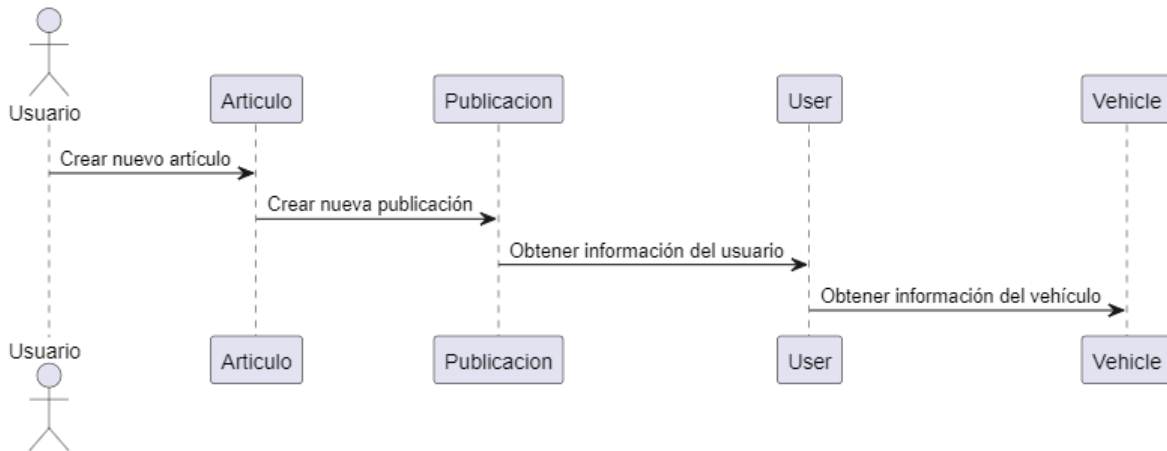
o baulremoto:	boolean
o asientoelectricos:	boolean
o puertacentralizado:	boolean
o manuales:	boolean
o cierrevidriosauto:	boolean
o bluetooth:	boolean
o dvd:	boolean
o vidrioaseg:	boolean
o repmp3:	boolean
o entradausb:	boolean
o cubiertaplato:	boolean
o plana:	boolean
o estribos:	boolean
o exploradoras:	boolean
o rineslujo:	boolean
o spoiler:	boolean
o garfabrica:	boolean
o garmecanica:	boolean
o negociable:	boolean
o venpermuta:	boolean
o domicilio:	boolean
o testdrivD:	boolean
o dochome:	boolean
o desc:	string
o valor:	float
o status:	boolean
o descripcion:	string
o puertas:	string
o motor:	string
o seguro:	boolean
o tecno:	boolean
o version:	string

Fuente: Autor

### 6.3.2. Diagrama de secuencia

En el diagrama secuencia se plasma el flujo y el comportamiento del sistema durante su funcionamiento en la Figura 4 se puede ver el proceso para publicar un vehículo en el aplicativo.

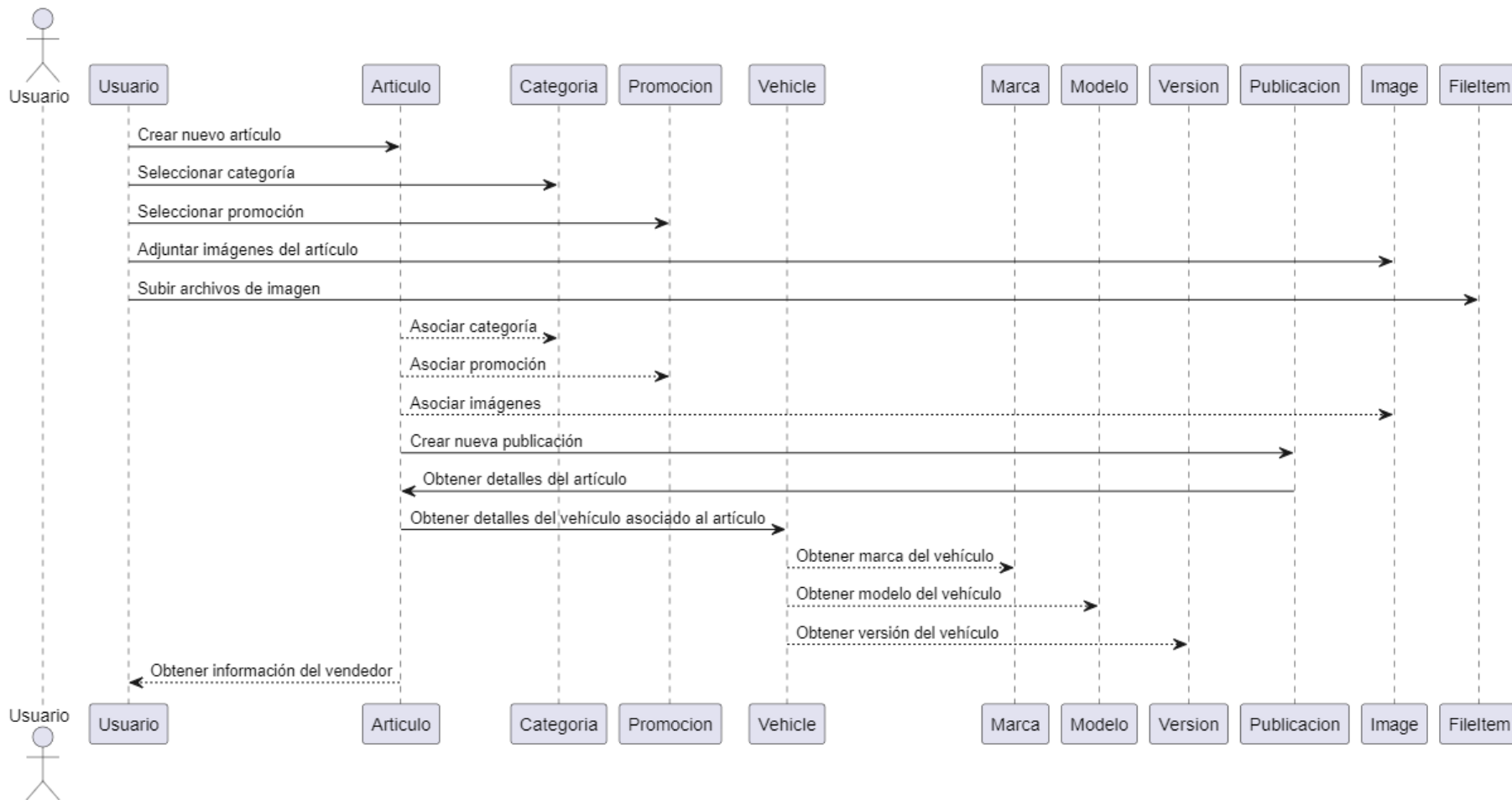
Figura 5 Diagrama de secuencia 1



Fuente: Autor

En el siguiente diagrama Figura 5 se puede evidenciar el proceso de creación de una publicación para un vehículo nuevo a un mayor nivel de detalle.

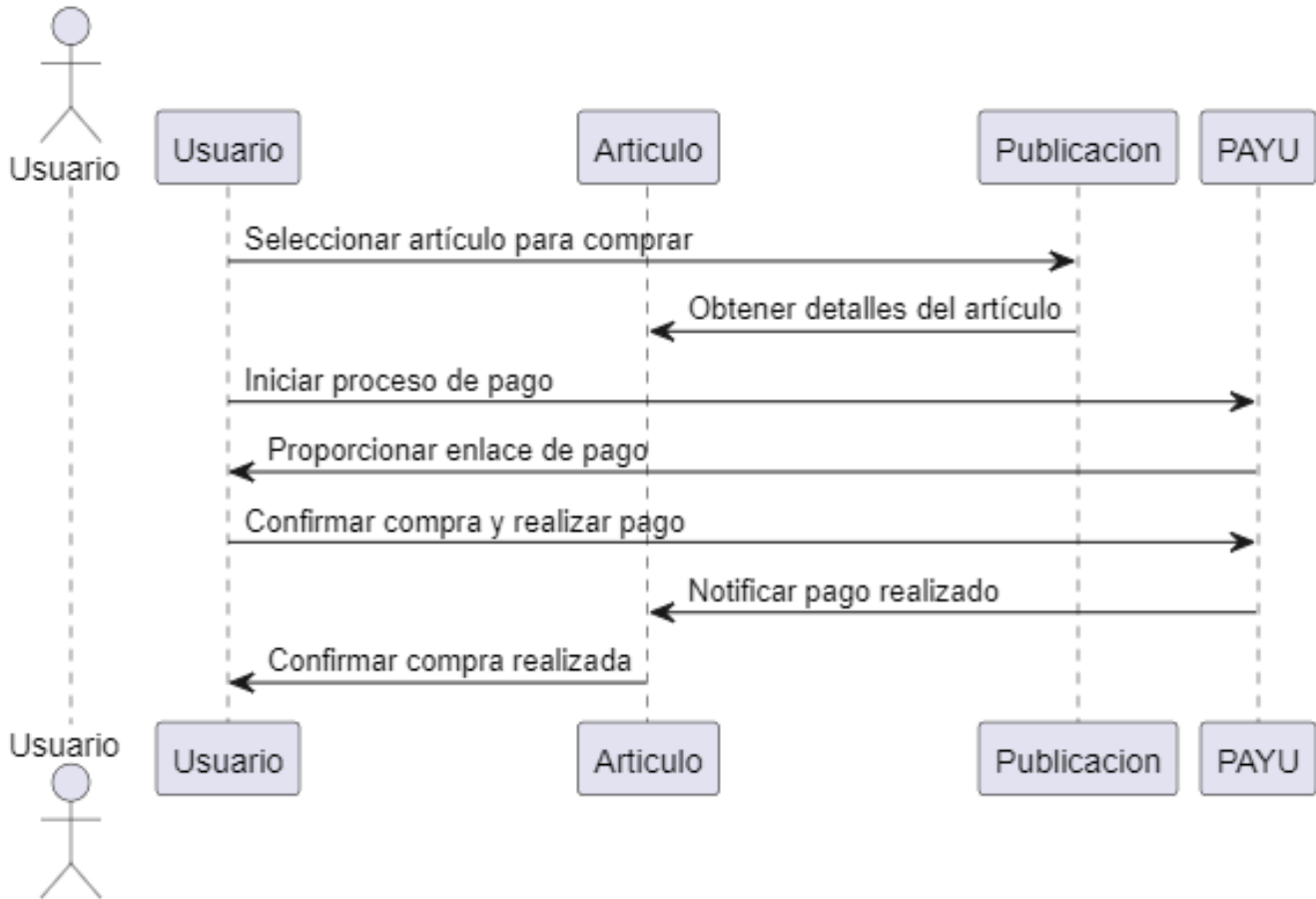
Figura 6 Diagrama de secuencia



Fuente: Autor

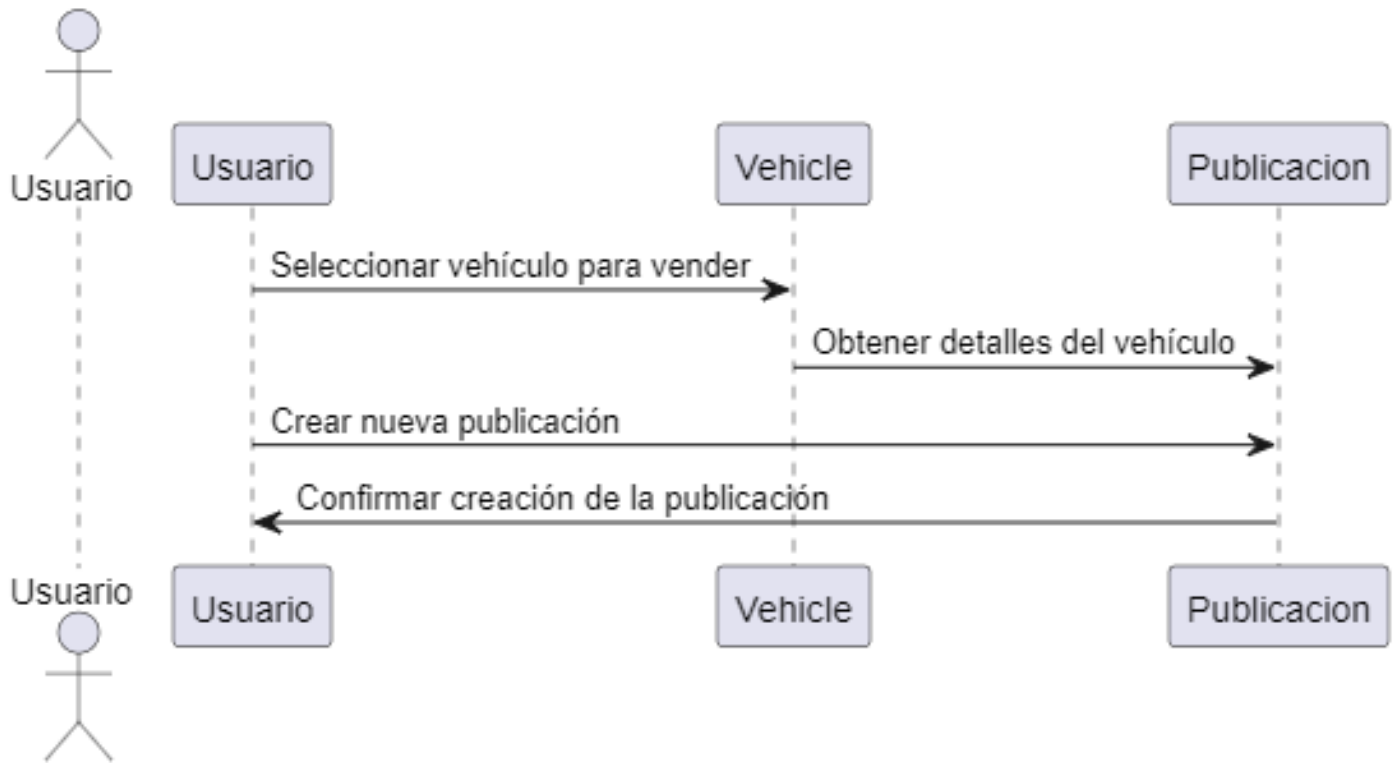
En la Figura 7 es el flujo que un cliente del aplicativo tiene que hacer para la compra de un artículo automovilístico.

Figura 7 Diagrama de secuencia



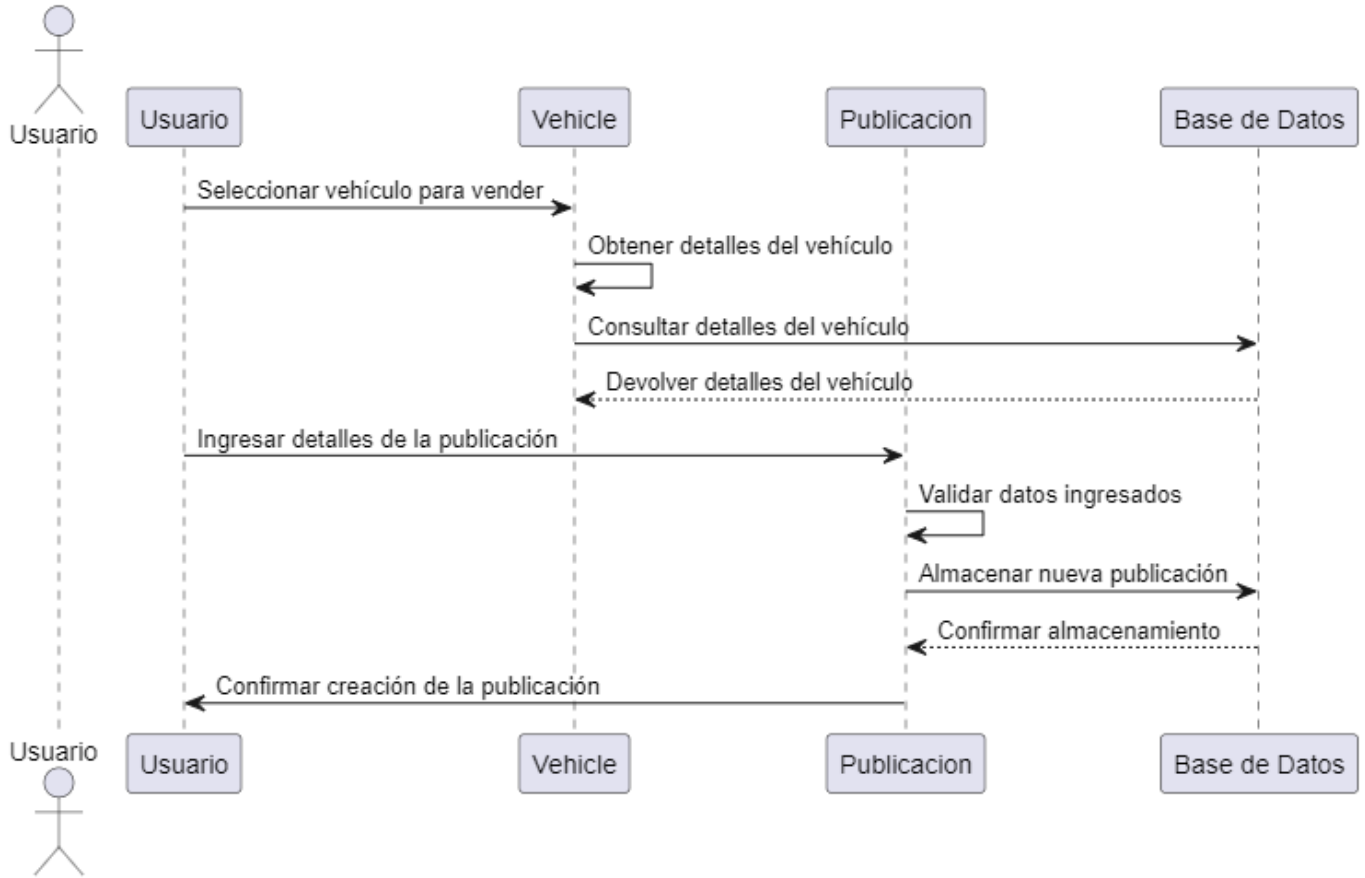
Fuente: Autor

Figura 8 Diagrama de secuencia



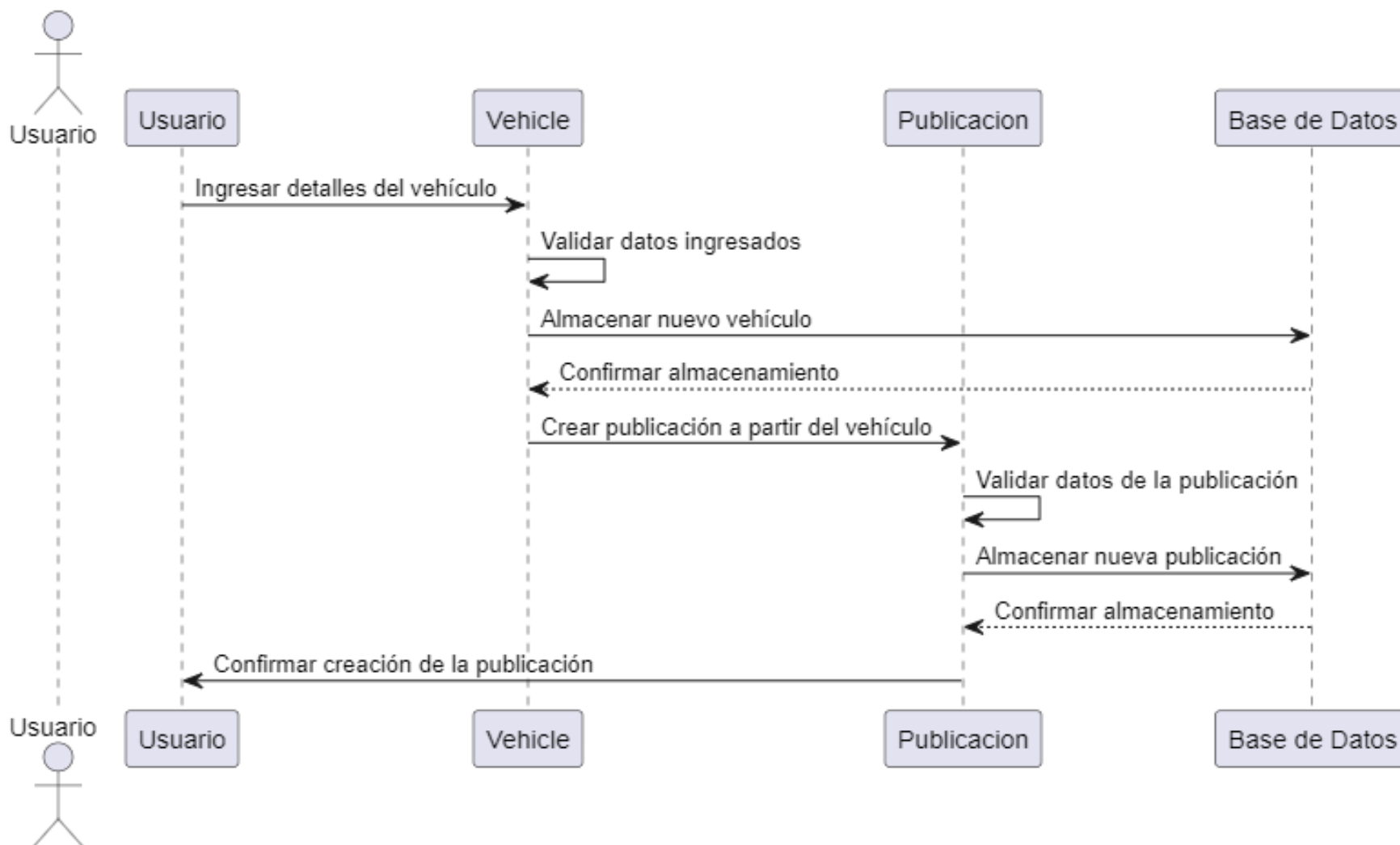
Fuente: Autor

Figura 9 Diagrama de secuencia



Fuente: Autor

Figura 10 Diagrama de secuencia



Fuente: Autor

### 6.3.3. Diagrama de clases

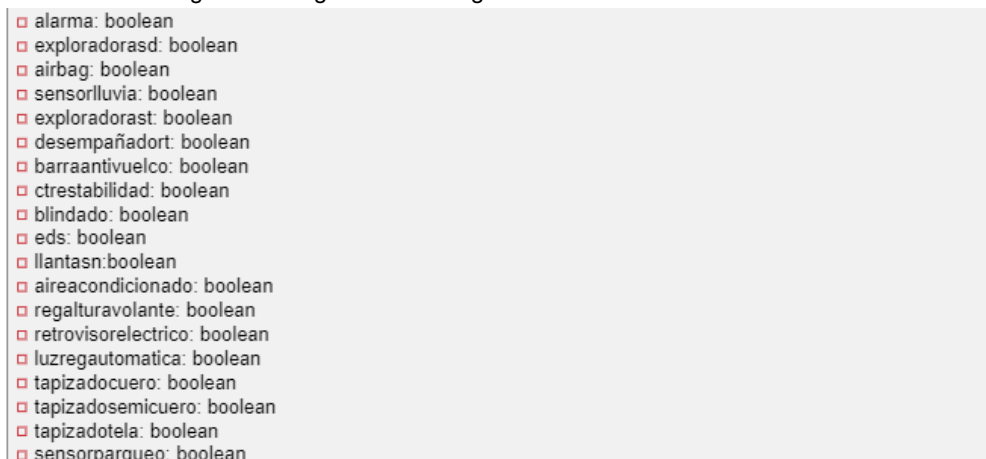
Con el fin de que se entienda mejor la clase de vehículo se incluye aparte para que se pueda leer bien sus atributos y métodos.

Figura 11 Fragmento del diagrama de clase “Clase de Vehicle”



Fuente: Autor

Figura 12 Fragmento del diagrama de clase “Clase de Vehicle”



Fuente: Autor

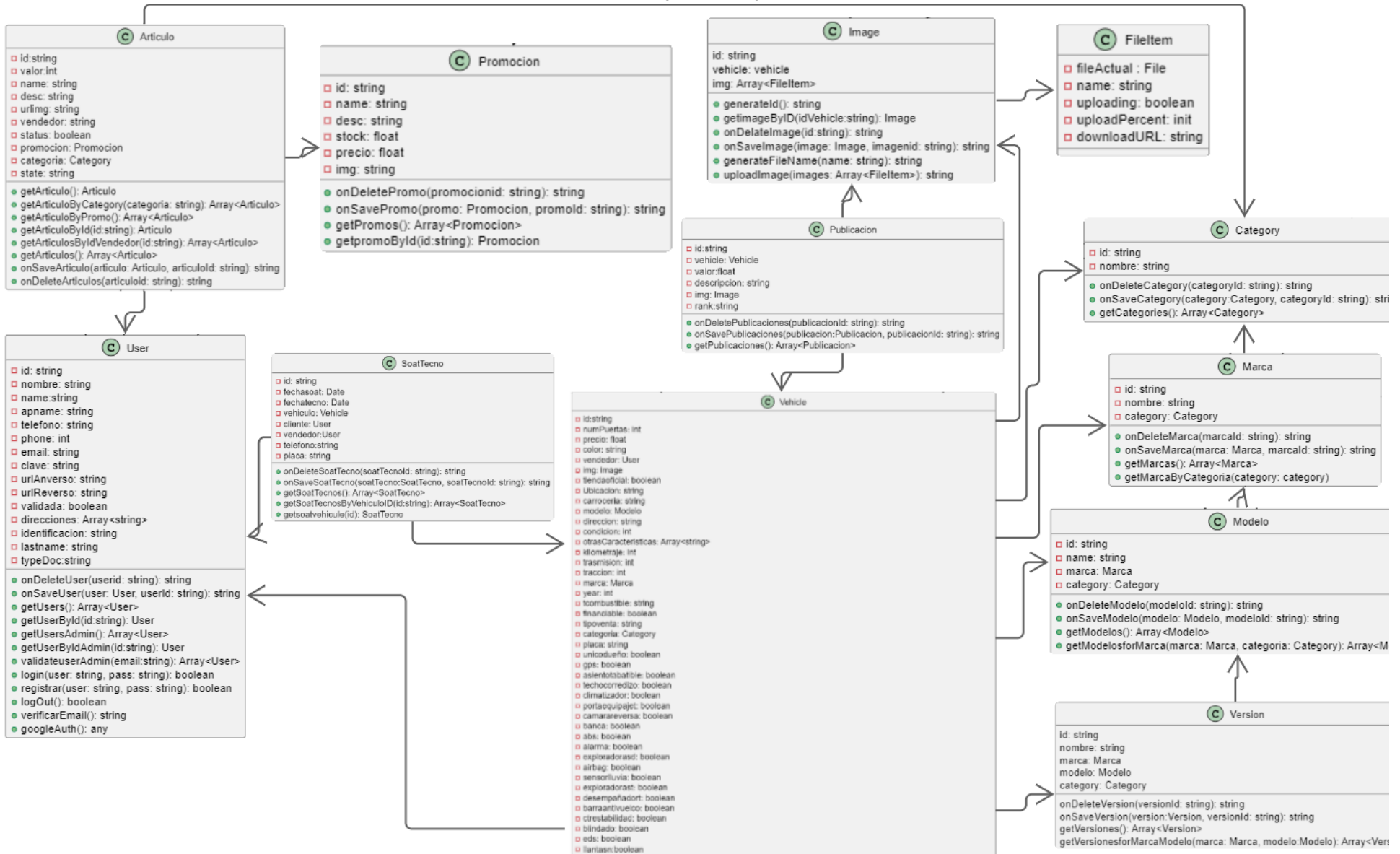
Figura 13 Fragmento del diagrama de clase “Clase de Vehicle”

<ul style="list-style-type: none"><li>❑ vidrioselect: boolean</li><li>❑ baulremoto: boolean</li><li>❑ asientoelectricos: boolean</li><li>❑ puertascentralizado: boolean</li><li>❑ manuales: boolean</li><li>❑ cierrevidriosauto: boolean</li><li>❑ bluetooth: boolean</li><li>❑ dvd: boolean</li><li>❑ vidrioseg: boolean</li><li>❑ repmp3: boolean</li><li>❑ entradausb: boolean</li><li>❑ cubiertaplato: boolean</li><li>❑ plana: boolean</li><li>❑ estribos: boolean</li><li>❑ exploradoras: boolean</li><li>❑ rineslujo: boolean</li><li>❑ spoiler: boolean</li><li>❑ garfabrica: boolean</li><li>❑ garmecanica: boolean</li><li>❑ negociable: boolean</li><li>❑ venpermuta: boolean</li><li>❑ domicilio: boolean</li><li>❑ testdrivD: boolean</li><li>❑ dochome: boolean</li><li>❑ desc: string</li><li>❑ valor: float</li><li>❑ status: boolean</li><li>❑ promocion: Promocion</li><li>❑ descripcion: string</li><li>❑ puertas: string</li><li>❑ motor: string</li></ul>
<ul style="list-style-type: none"><li>❑ seguro: boolean</li><li>❑ tecno: boolean</li><li>❑ version: string</li></ul>
<ul style="list-style-type: none"><li>● onDeleteVehicules(vehicleid: string): string</li><li>● onSaveVehicle(vehicle: Vehicle, vehicled: string): string</li><li>● onUpdateVehicle(vehicle: Vehicle, vehicled: string): string</li><li>● getVehicules(): Array&lt;Vehicle&gt;</li><li>● getVehiculesByldVendor(id: string): Array&lt;Vehicle&gt;</li><li>● getVehicleByld(id: string) Vehicle</li><li>● getVehiculesByldState(id: string, state: string): Array&lt;Vehicle&gt;</li><li>● getVehicleByMarca(marca: Marca): Array&lt;Vehicle&gt;</li><li>● getVehicleByMarcaByCategoria(marca: string, categoria: Category): Array&lt;Vehicle&gt;</li><li>● getVehicleByCategoriaBYMarcaByModelo(categoria: Category, marca: Marca, modelo: Modelo): Array&lt;Vehicle&gt;</li><li>● getVehicleByKilometraje(desde: int, hasta: int, promo: boolean): Array&lt;Vehicle&gt;</li></ul>

Fuente: Autor

### 6.3.3.1 Diagrama de clases completo

Figura 14 Diagrama de Clases



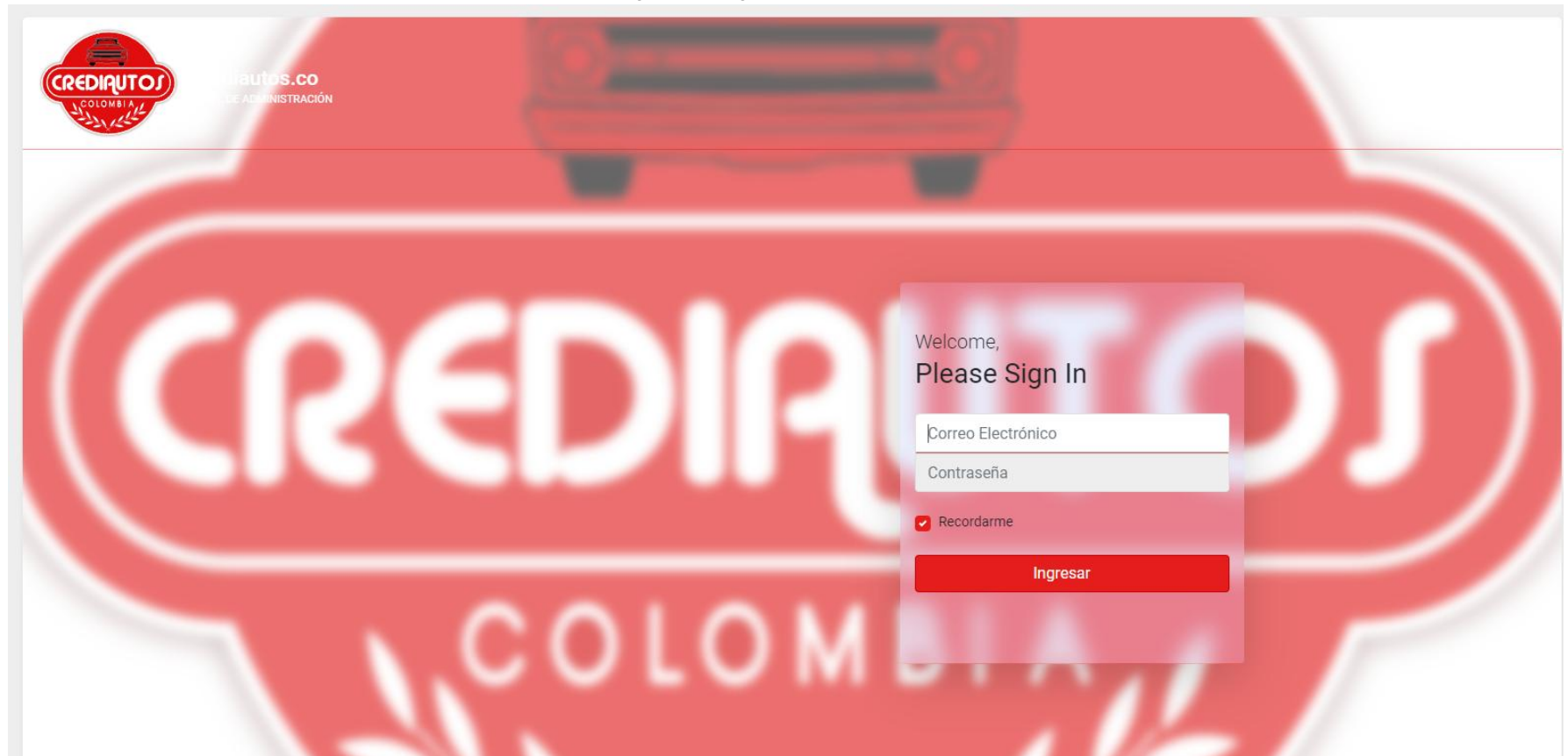
Fuente: Autor

### 6.3.4. Diseño del módulo administrativo

#### 6.3.4.1 *Página de inicio de sesión*

La Figura 15 es la página de inicio de sesión del módulo administrativo el cual es un aplicativo web y permite el ingreso a los colaboradores de la empresa Crediautos.

Figura 15 Página de inicio de sesión

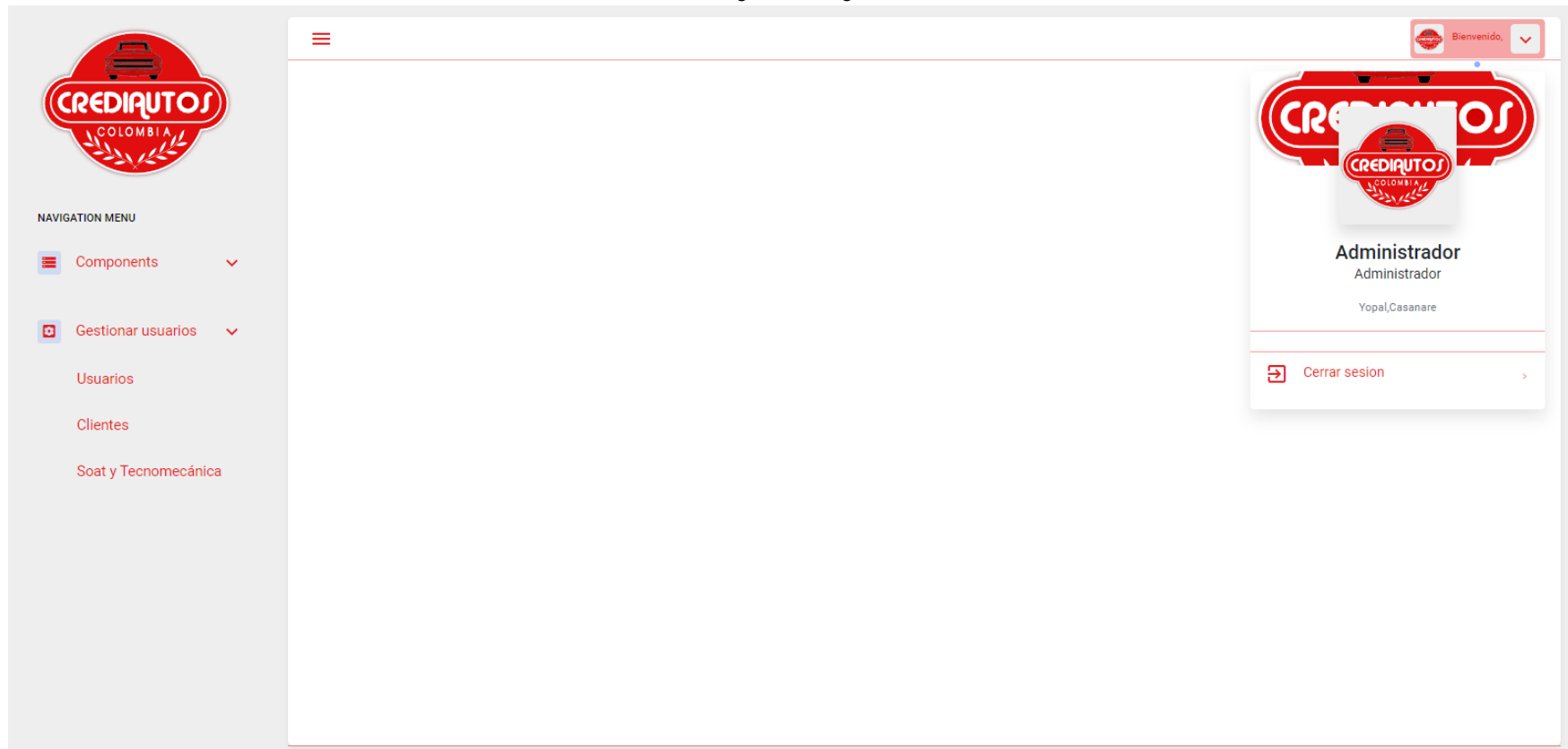


Fuente: Autor

### 6.3.4.2 *Página de inicio*

Este es el landing page del módulo administrativo el cual se muestra después de hacer el inicio de sesión.

Figura 16 Página de inicio



Fuente: Autor

### 6.3.4.3 Menú de navegación

Aquí se puede evidenciar todos los posibles botones que pueda tener el menú de navegación.

Figura 17 Menú de navegación



Fuente: Autor

#### 6.3.4.4 Listado de vehículos

En la lista de vehículos podemos ver una tabla con los automóviles ya creados y sus posibles funcionalidades.

Figura 18 Listado de vehículos

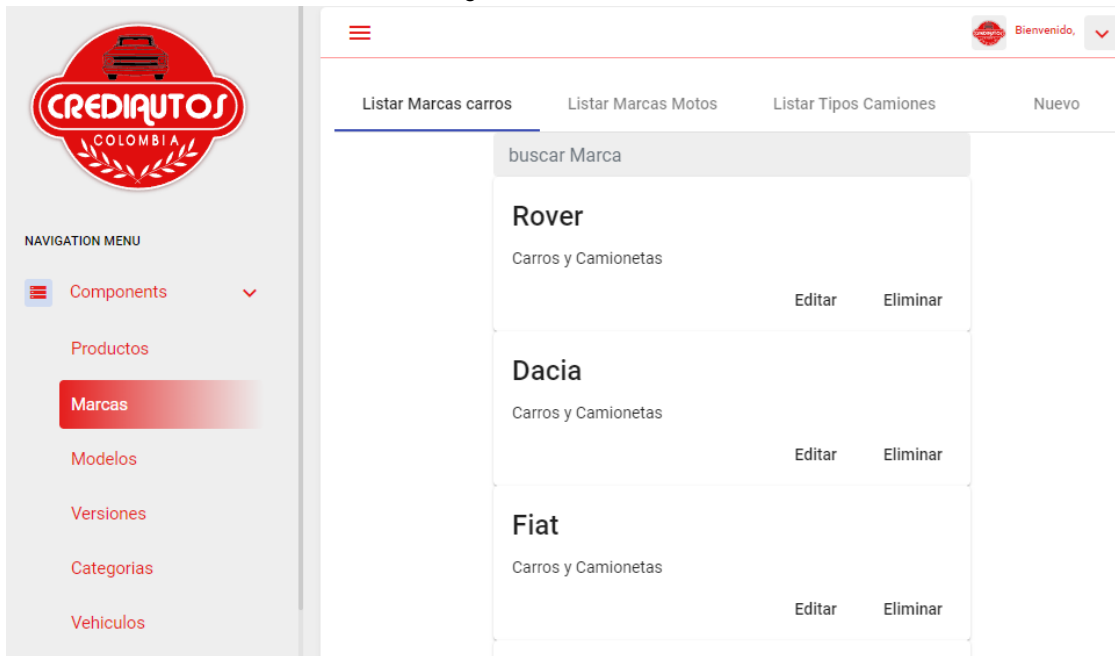
Vendedor	Publicación	Marca	Modelo	Version	Año	Placa	Publicar	Promocion	Valor	Accion
crediautos colombia	Toyota Prado Sumo Modelo 2008	Toyota	T-100	otros	2008	008	<input checked="" type="checkbox"/>	<input type="checkbox"/>	72000000	
crediautos colombia	CHEVROLET NHR 2001	Chevrolet	NKR	otros	2001	250	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	52000000	
Belkis Ibarra	Volkswagen 2008	Volkswagen	Cross Up	Otro	2008	CXL959	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22	
Nelson Blanco	prueba	Fiat	Palio Adventure	Otro	2010	xxx000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	100000	

Fuente: Autor

#### 6.3.4.4 Listado de marcas

En la Figura 19 se pueden listar las marcas de los vehículos y con los botones para crear, editar y eliminar una marca.

Figura 19 Listado de marcas

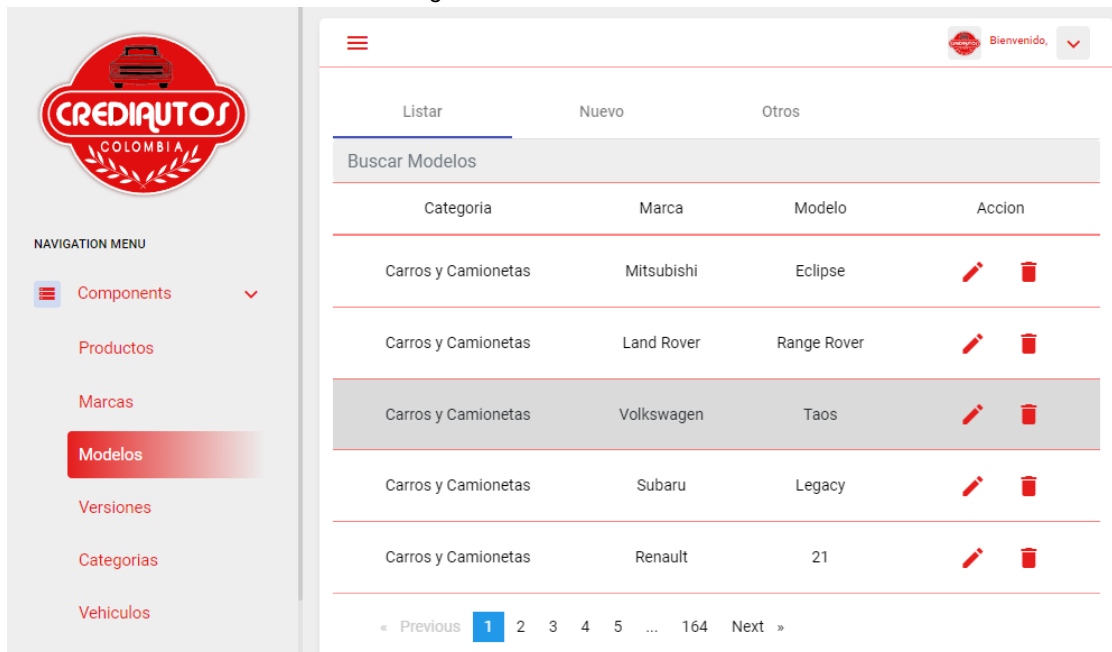


Fuente: Autor

#### 6.3.4.5 Listado de modelos

Al igual que la Figura 20 se tiene una tabla, pero esta vez para los modelos de vehículos donde se pueden crear, editar y eliminar

Figura 20 Listado de modelos



Fuente: Autor

### 6.3.4.6 Listado de versiones

Lista todas las versiones que se tienen de un modelo de vehículo, también tiene la funcionalidad de crear, editar y eliminar versiones.

Figura 21 Listado de versiones

Categoria	Marca	Modelo	Version	Accion
Carros y Camionetas	Kia	Rio	Space	[Edit] [Delete]
Carros y Camionetas	Kia	Rio	1.4	[Edit] [Delete]
Carros y Camionetas	Chevrolet	Spark	Gt	[Edit] [Delete]
Carros y Camionetas	Kia	Rio	1.0	[Edit] [Delete]
Carros y Camionetas	Audi	Turbo	1.8	[Edit] [Delete]
Carros y Camionetas	Renault	Logan	Expression	[Edit] [Delete]

Fuente: Autor

### 6.3.4.7 Listado de categorías

Esta pantalla lista las categorías que se pueden llegar a existir en el proyecto, también puede crear, editar y eliminar.

Figura 22 Listado de categorías

Categoría	Acción
Carros y Camionetas	[Editar] [Eliminar]
Carros de coleccion	[Editar] [Eliminar]
Motos	[Editar] [Eliminar]
Otros vehiculos	[Editar] [Eliminar]

Fuente: Autor

### 6.3.4.8 Listado de usuarios

Aquí es donde se puede administrar los usuarios que se pueden iniciar sesión en el módulo administrativo.

Figura 23 Listados de usuarios del módulo administrativo.

Usuario	Email	Telefono	Role	Detalle	Editar	Eliminar
Crediautos Colombia	root@crediautos.co	3138073045	admin	Detalle	Editar	Eliminar
admin Snader root	harold_parra@yopmail.com	3209210549	admin	Detalle	Editar	Eliminar
usuario prueba prueba@crediautos.co		1111111111	agente	Detalle	Editar	Eliminar
Gerencia Crediautos	gerencia@crediautos.co	3138073045	admin	Detalle	Editar	Eliminar

Fuente: Autor

### 6.3.4.9 Listado de clientes

Aquí es donde se puede administrar los clientes que se registraron en la aplicación web y móvil.

Figura 24 Listado de clientes

Nombre	Cedula	Telefono	Vehiculos	Documento	Validar	Accion
Nuage Laboratoire			<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input type="radio"/> <input type="text"/> <input type="trash"/>
este este	45678921	3209210549	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input type="radio"/> <input type="text"/> <input type="trash"/>
Nelson Enrique Blanco	1049623634	3118550011	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="radio"/> <input type="text"/> <input type="trash"/>

Fuente: Autor

### 6.3.4.10 Listado de Soat y Tecnomecánica

Figura 25 Listado de Soat y Tecnomecánica

The screenshot shows the 'CREDITOS COLOMBIA' web application interface. On the left is a navigation menu with options: Components, Gestionar usuarios, Usuarios, Clientes, and Soat y Tecnomecánica (highlighted). The main content area is titled 'LISTADO DE VEHICULOS DE CLIENTES' and features a 'Refrescar Tabla' button. Below the title are search filters for dates and SOAT, and a search icon. The table below lists two vehicles:

Nombre	Telefono	Placa	SOAT	Tecnomecánica	Accion
nelson b	456456	xxc123	viernes, 17 de diciembre de 2021	viernes, 24 de diciembre de 2021	 
snader	564	dsa	jueves, 23 de diciembre de 2021	lunes, 20 de diciembre de 2021	 

Fuente: Autor

### 6.3.4.11 Listado de promociones

Figura 26 Listado de promociones

NAVIGATION MENU

- Components
- Productos**
- Marcas
- Modelos
- Versiones
- Categorías
- Vehículos

LISTADO DE PROMOCIONES

buscar promociones

Nombre	Descripcion	Precio		
Prueba Promocion	descripción de la promoción	60000	Modificar	Eliminar

Fuente: Autor

### 6.3.4.12 Listado de artículos

Figura 27 Listado de artículos

NAVIGATION MENU

- Components
- Productos
- Marcas
- Modelos
- Versiones
- Categorías
- Vehículos**

LISTADO DE ARTICULOS

Buscar Producto

Vendedor	Publicación	Categoria	Publicar	Promocion	Valor	Accion
Brayan Snader Sierra Riveros	das	Carros y Camionetas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	123456789	
Brayan Snader Sierra Riveros	dsa	Carros y Camionetas	<input type="checkbox"/>	<input type="checkbox"/>	1234578	

Fuente: Autor

### 6.3.5. Diseño de usuario final

#### 6.3.5.1 Diseño página inicial

Figura 28 Fragmento del diseño página inicial



Publicidad

#### Vehículos Usados Destacados



#### Vehículos Nuevos (OEM) Destacados



Fuente: Autor

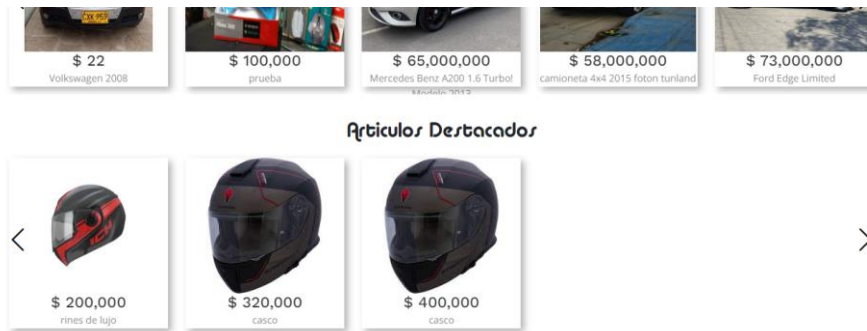
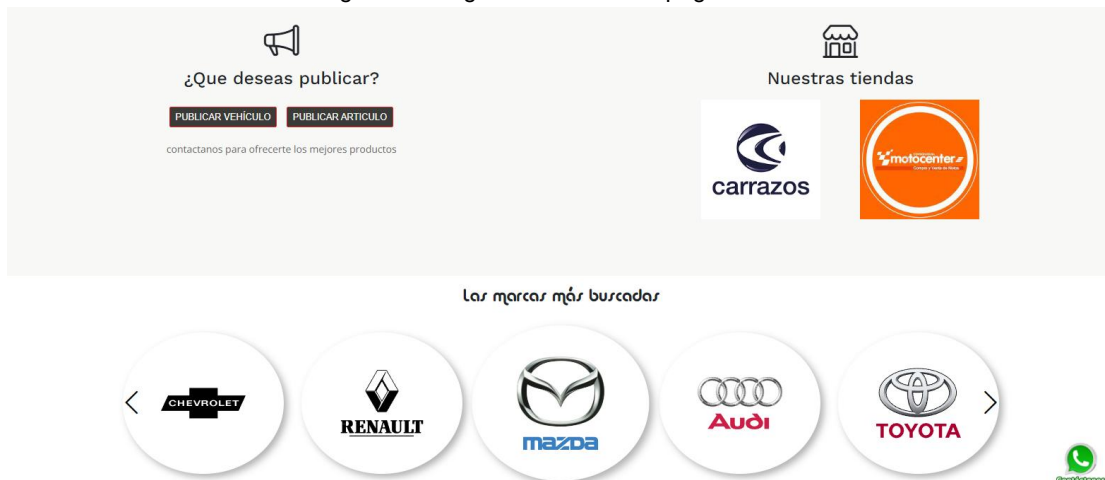


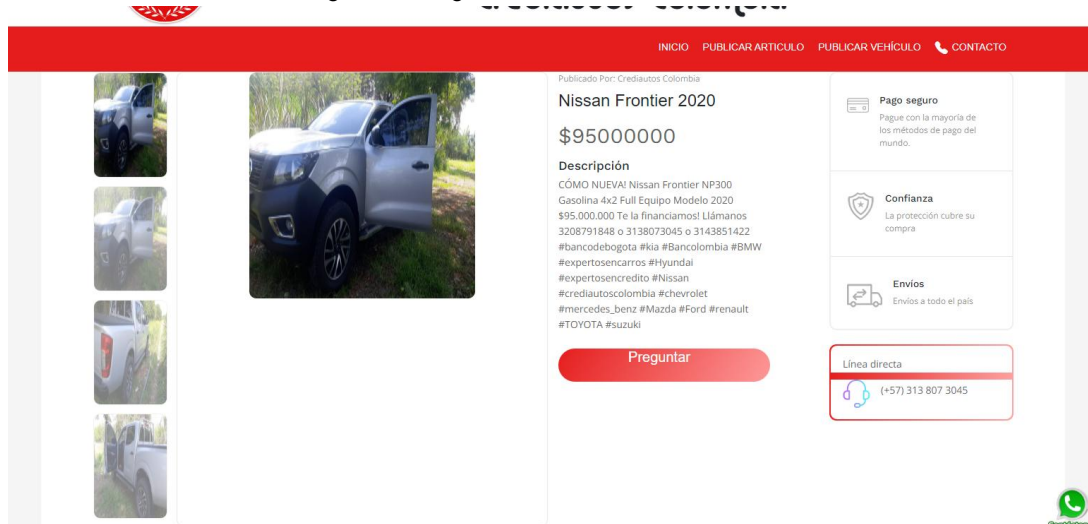
Figura 29 Fragmento del diseño página inicial



Fuente: Autor

### 6.3.5.2 Diseño de vehículo seleccionado


Figura 30 Fragmento del diseño vehículo seleccionado



Fuente: Autor

Figura 31 Fragmento del diseño vehículo seleccionado

Características   Condiciones   Mas Infomación	
Marca	Nissan
Modelo	NP300
Año	2020
Color	Gris
Tipo de combustible	Gasolina
Puertas	4
Transmisión	Mecanica
Motor	2500
Tipo de carroceria	pick Up Doble Cabina
Kilómetros	45000 km



Fuente: Autor

### 6.3.5.3 Diseño de artículo seleccionado

Figura 32 Diseño artículo seleccionado

INICIO PUBLICAR ARTICULO PUBLICAR VEHICULO CONTACTO

Publicado Por: Brayán Snider Sierra Reveros

**Casco**  
\$400000

**Descripción**  
casco talla s color negro

[Preguntar](#)


[Comprar](#)

**Pago seguro**  
Pague con la mayoría de los métodos de pago del mundo.

**Confianza**  
La protección cubre su compra

**Envios**  
Envíos a todo el país.

**Línea directa**  
[\(+57\) 313 807 3045](tel:+573138073045)



Fuente: Autor

### 6.3.5.4 Diseño de filtros de productos

Figura 33 Diseño de filtros

The screenshot displays the Crediautos Colombia website interface. At the top, there is a red navigation bar with social media icons, contact information (313 807 3045, crediautos.colombia@gmail.com), and links for 'Ingresar' and 'Crear cuenta'. Below this is the 'Crediautos Colombia' logo and a secondary navigation bar with links for 'INICIO', 'PUBLICAR ARTICULO', 'PUBLICAR VEHICULO', and 'CONTACTO'.

The main content area is titled 'Carros y Camionetas' and shows 'Visualizando 1-15 de 20 resultados'. On the left, there are three filter panels:

- Marcas:** A list of car brands including Alfa Romeo, Aston Martin, Audi, BMW, BYD, Baic, Brilliance, and Buick.
- Precio:** A range filter from '\$0 Millones' to '\$200 Millones' with a 'Filtrar' button.
- Color:** A list of colors including Amarillo, Azul, Beige, Blanco, Celeste, Dorado, Gris, and Marrón.
- Kilometros:** A range filter from '0 kilometros' to 'Kilometros' with a 'Filtrar' button.

The main grid displays 15 car listings, each with a 'Promoc' badge, a photo, and text details:

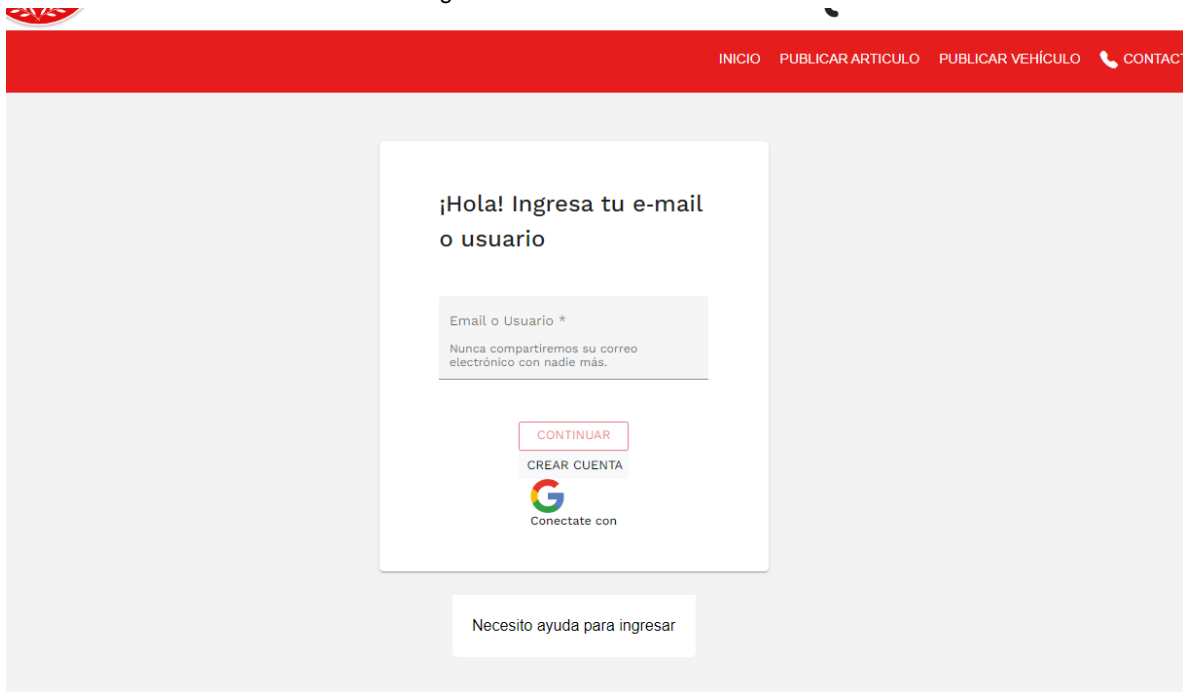
- Listing 1:** CHEVROLET NHR 2001, 500000 Km, \$52,000,000.
- Listing 2:** Volkswagen 2008, 0 Km, \$22.
- Listing 3:** Prueba, 2010 | 0 Km, \$100,000.
- Listing 4:** Toyota Hilux 2.5 Diésel 4x4 Modelo 2014, 2014 | 115000 Km, \$89,000,000.
- Listing 5:** CÓMO NUEVO! Nissan March 2016 Full Equipol, 2016 | 70000 Km, \$39,000,000.
- Listing 6:** Mercedes Benz A200 1.6 Turbo Modelo 2013, 2013 | 0 Km, \$65,000,000.
- Listing 7:** Audi A4 1.8 Turbo Modelo 2011, 2011 | 115000 Km, \$56,000,000.
- Listing 8:** Camioneta 4x4 2015 Foton Tuniang, 2015 | 0 Km, \$58,000,000.
- Listing 9:** CÓMO NUEVA! Renault Duster 4x4, 2018 | 55000 Km, \$63,900,000.
- Listing 10:** Mercedes Benz 2014 ML 250CDI, 2014 | 75000 Km, \$95,800,000.
- Listing 11:** CÓMO NUEVO! Renault Twingo 2l Modelo 2014, 2014 | 96000 Km.
- Listing 12:** Ford Edge Limited, 2014 | 0 Km, \$73,000,000.

On the right side of the page, there are two WhatsApp chat icons.

Fuente: Autor

### 6.3.5.5 *Diseño de inicio de sesión*

Figura 34 Diseño inicio de sesión

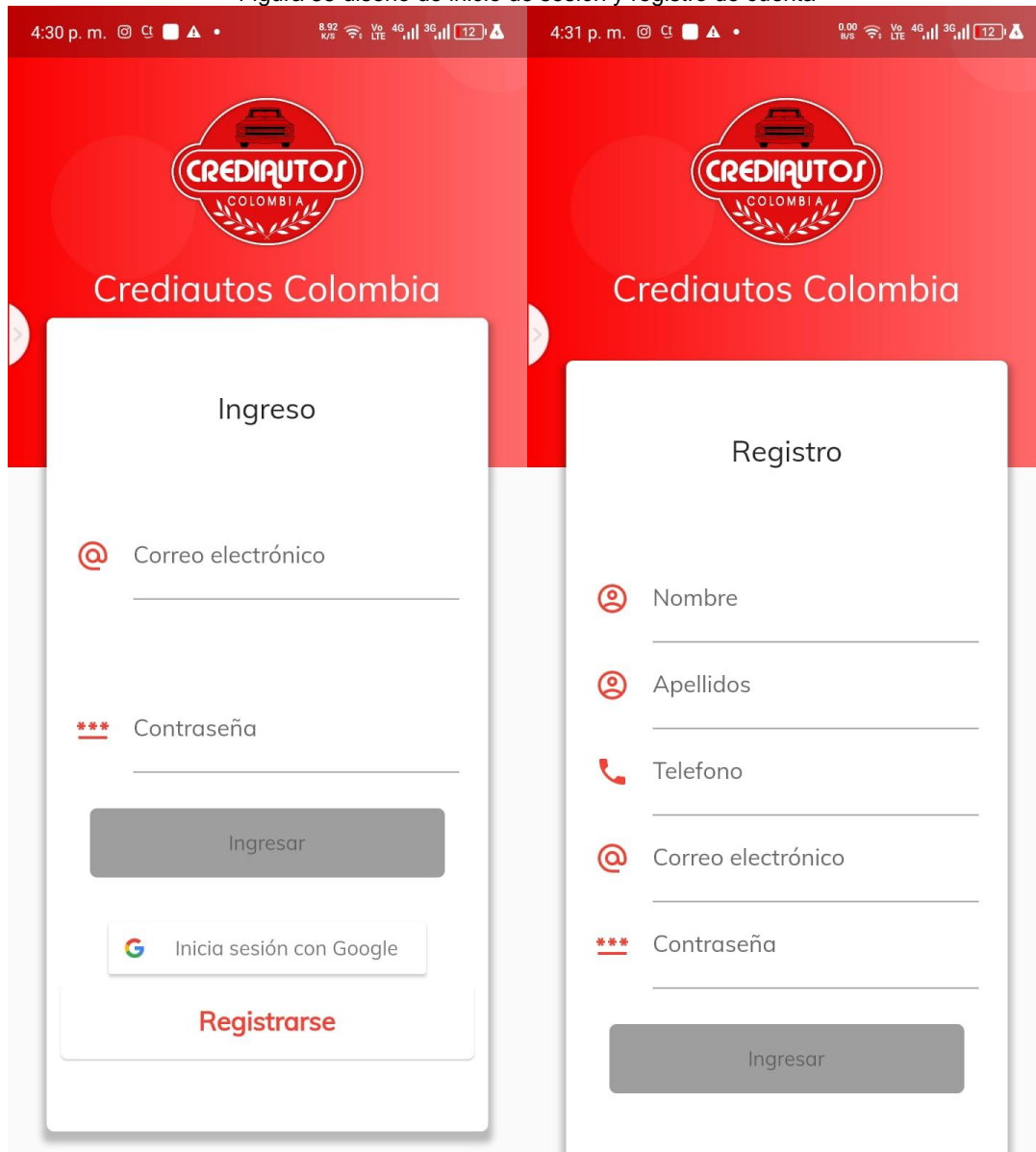


Fuente: Autor

### 6.3.6. Diseño de aplicación móvil

#### 6.3.6.2 Diseño del inicio de sesión y registro de cuenta

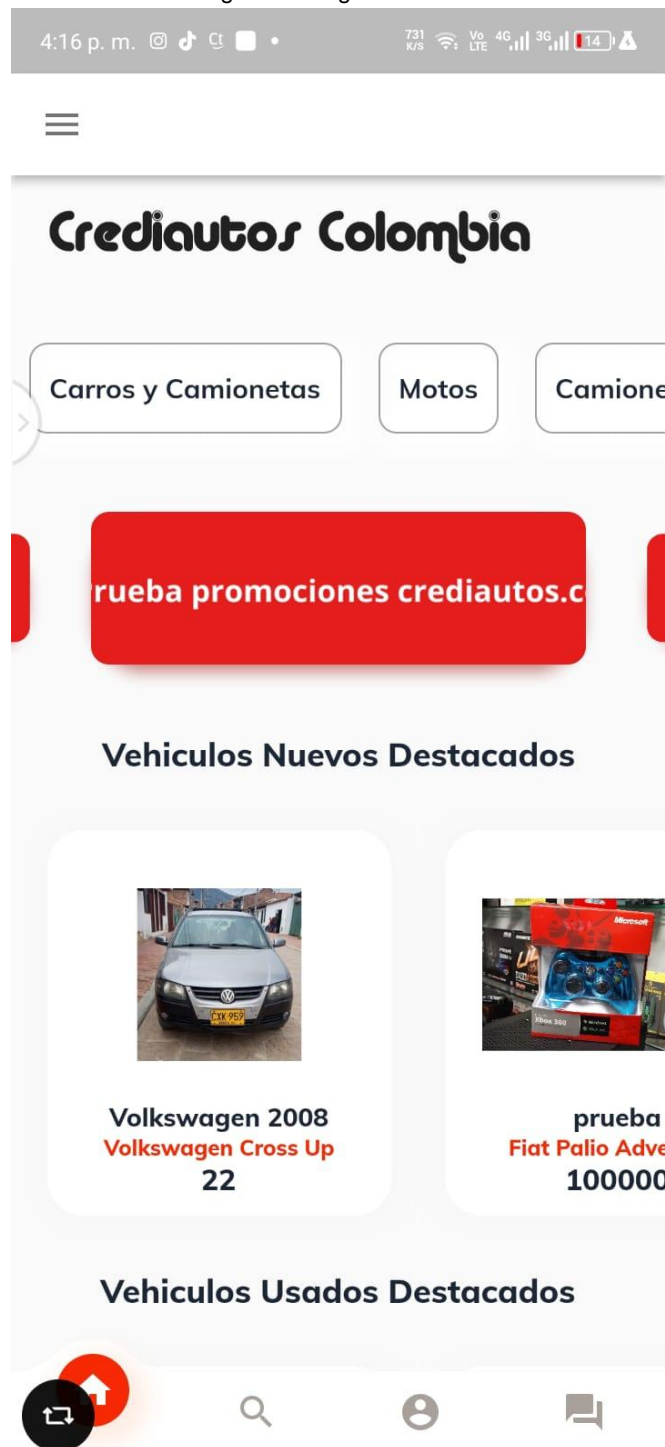
Figura 35 diseño de inicio de sesión y registro de cuenta



Fuente: Autor

### 6.3.6.3 *Página de inicio*

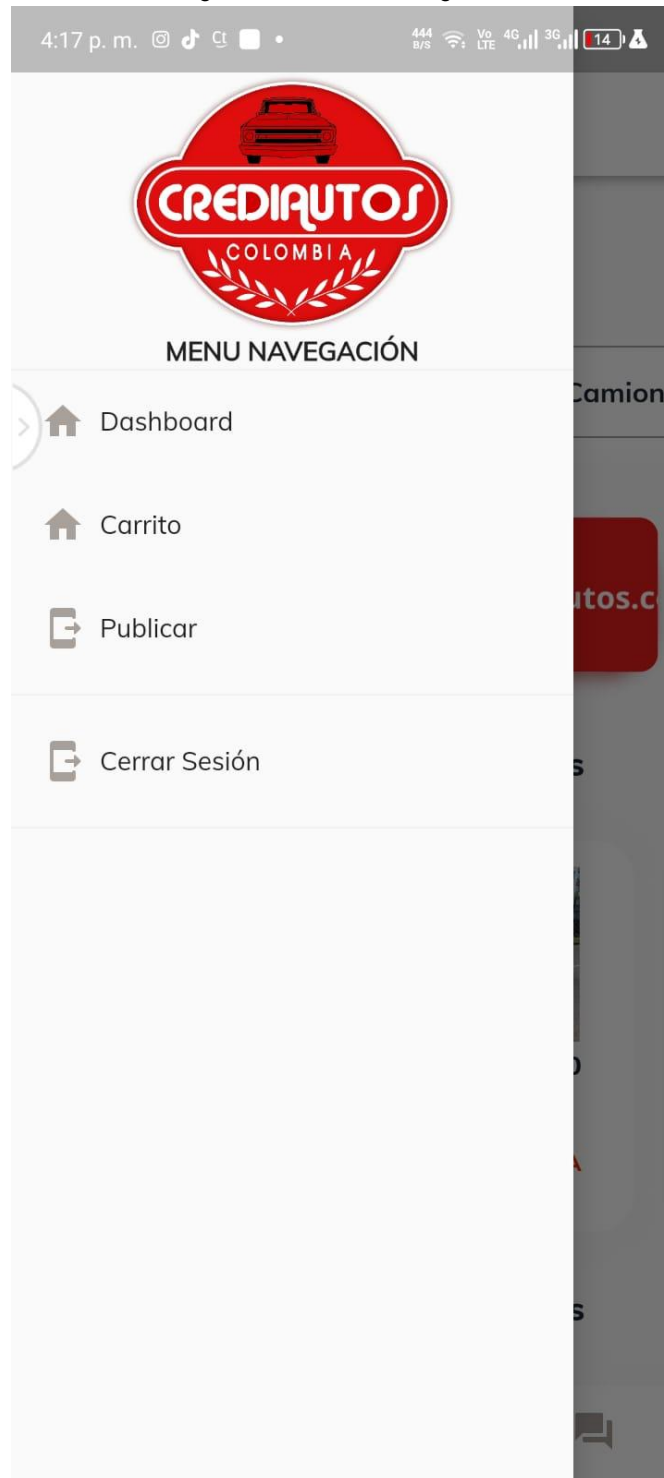
Figura 36 Página de inicio



Fuente: Autor

#### 6.3.6.4 Menú de navegación

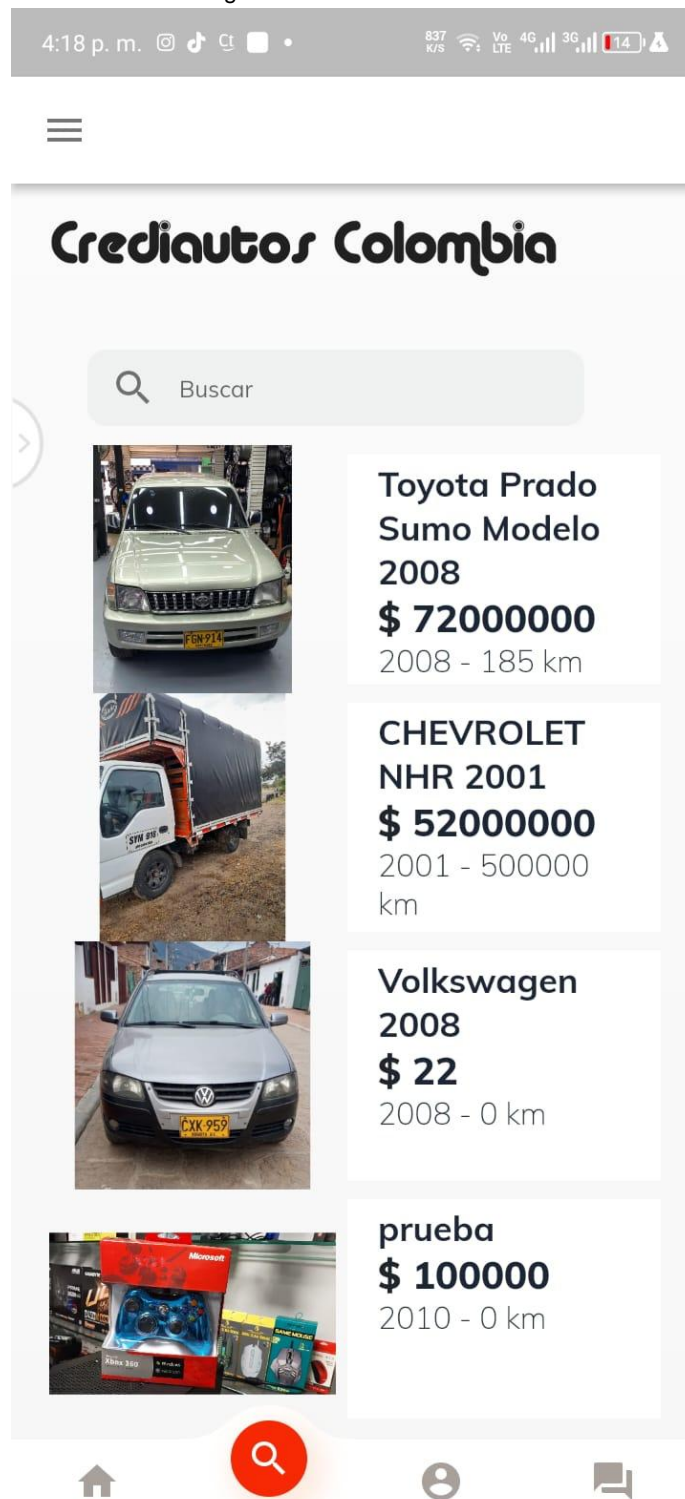
Figura 37 Menú de navegación



Fuente: Autor

### 6.3.6.5 Filtro de vehículos

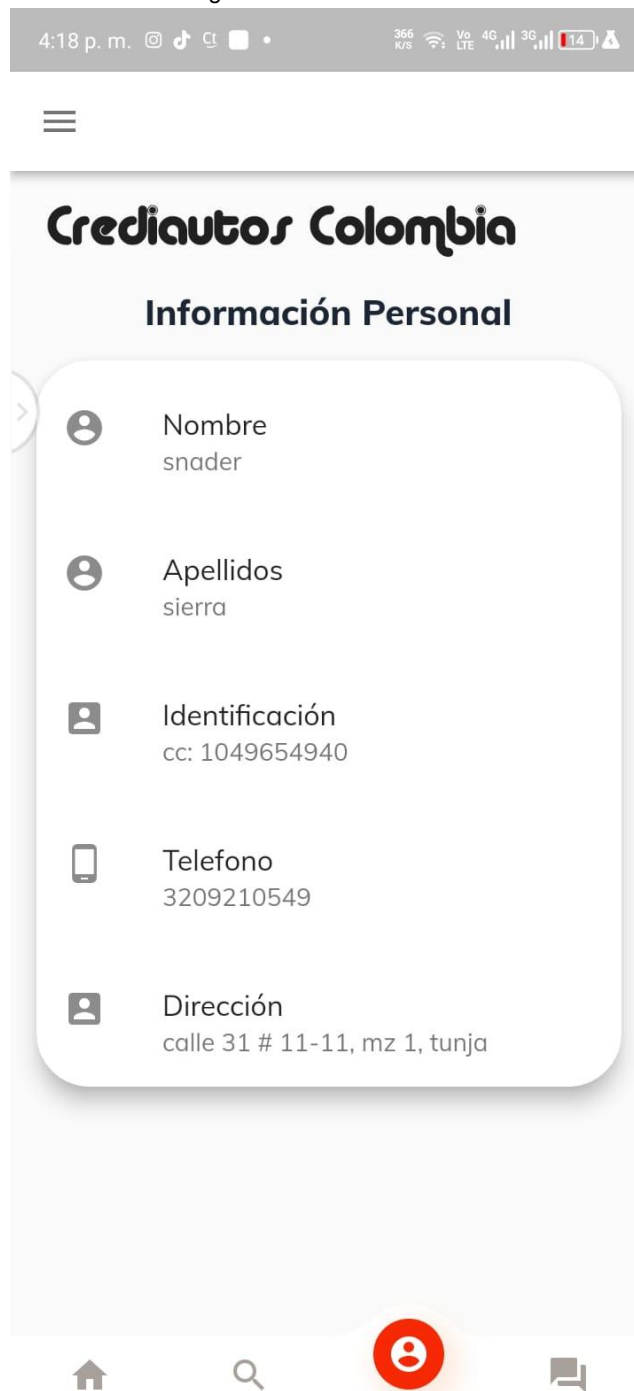
Figura 38 Filtro de vehículos



Fuente: Autor

### 6.3.6.6 Datos de usuario

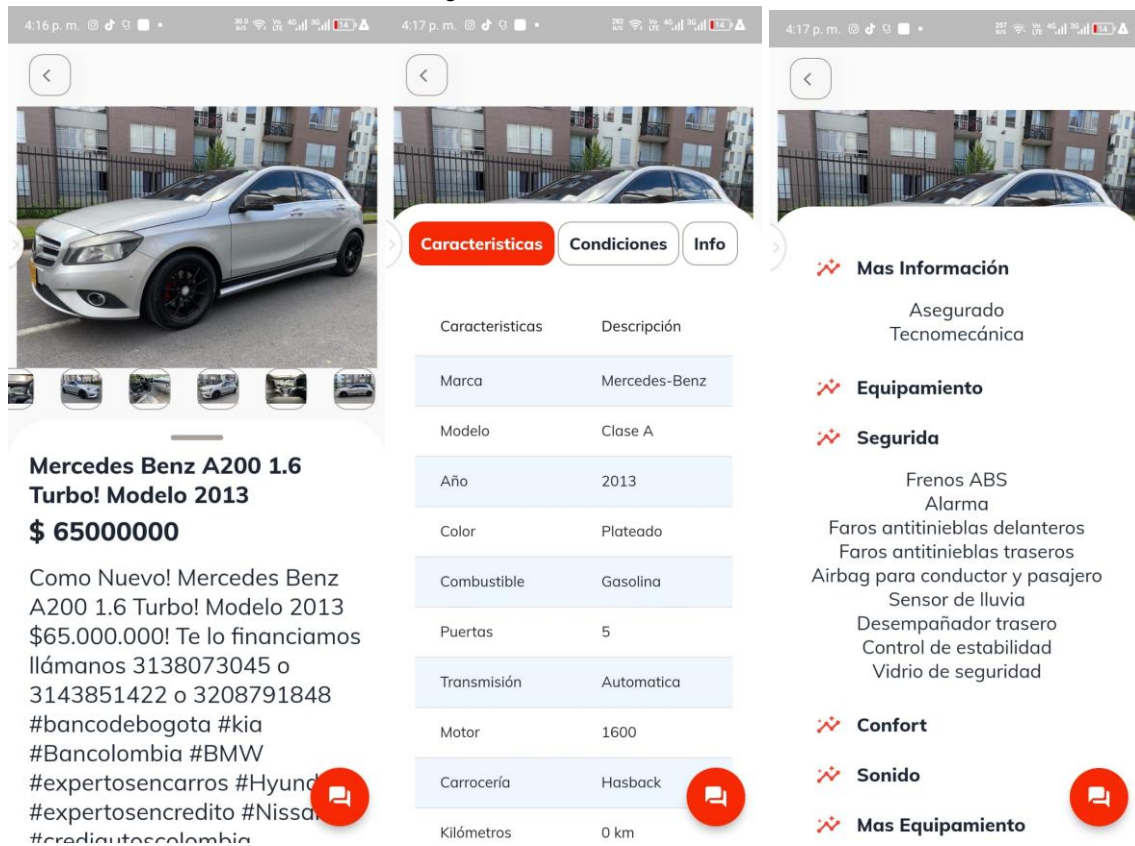
Figura 39 Perfil de usuario



Fuente: Autor

6.3.6.7 Vehículo seleccionado

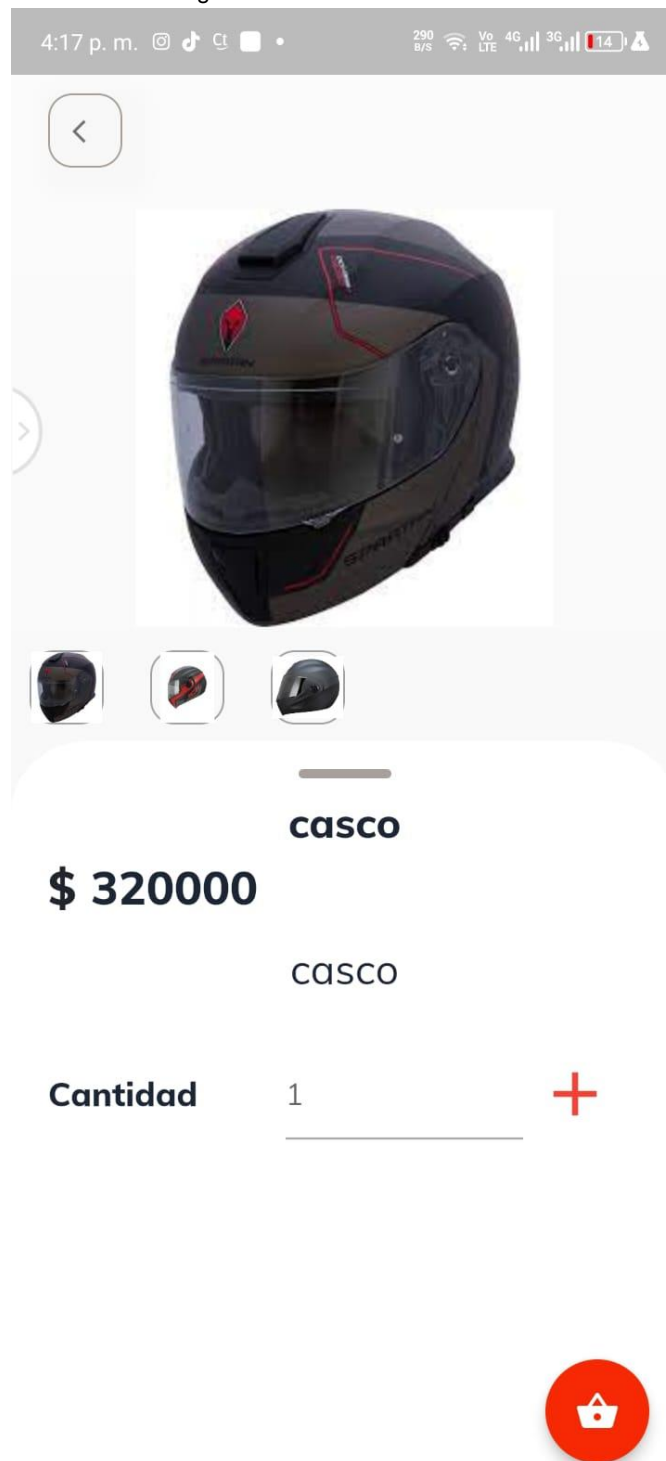
Figura 40 Vehículo seleccionado



Fuente: Autor

### 6.3.6.8 Artículo seleccionado

Figura 41 Artículo seleccionado



Fuente: Autor

### 6.3.6.9 Diseño de publicar

Figura 42 Diseño de publicar

The screenshot shows a mobile application interface for publishing. At the top, the status bar displays the time as 4:40 p. m. and various system icons. Below the status bar, the word 'Publicar' is centered. The main content area is titled 'Paso 1 de 4'. It contains a text input field for 'Titulo para tu publicacion', a larger text input field for 'Descripción', and four dropdown menus labeled 'Selecciona la Categoria', 'Seleccione la Marca', 'Seleccione el Modelo', and 'Seleccione el Version de tu'. Below these is a text input field for '¿De que año es tu?'. At the bottom right of the form area, there is a '0/4' indicator. A prominent red button with the text 'Siguiete' is located at the bottom center of the screen.

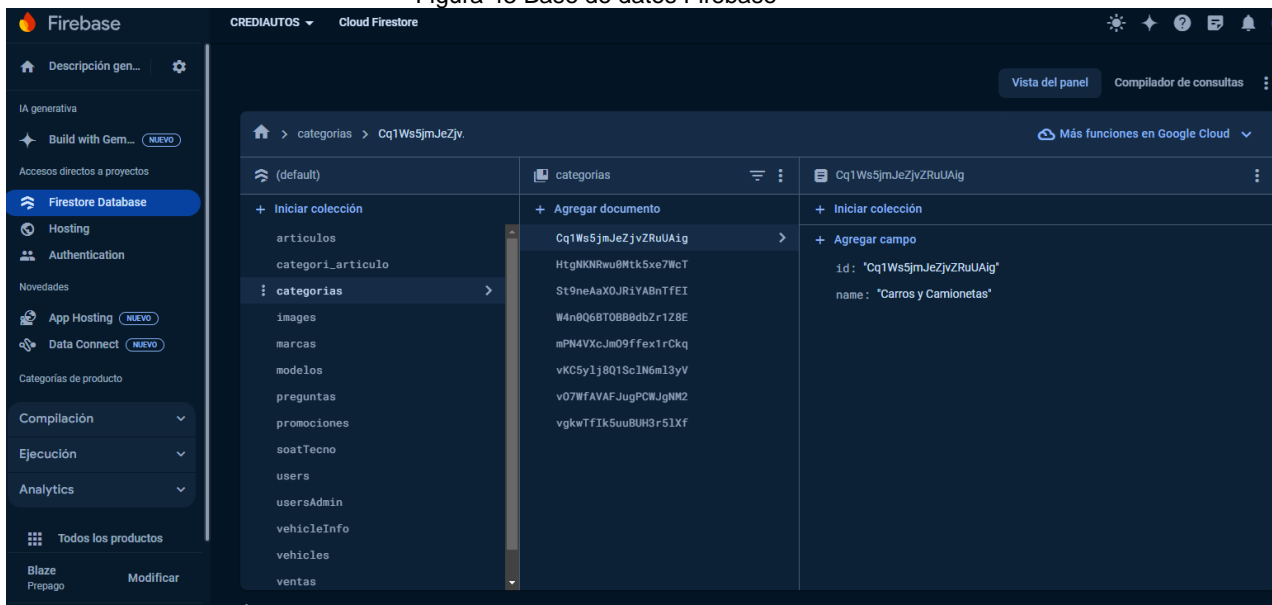
Fuente: Autor

## 6.4. Fase 4. Desarrollo de la aplicación

### 6.4.1. Desarrollo de la base de datos

Teniendo en cuenta la el diagrama de entidad-relación y el patrón de diseño MVC se crea una base de datos utilizando Firebase, ya que facilita la autenticación por medio de email y contraseña, como también, la autenticación por Google, además que es muy económico para la empresa Crediautos.

Figura 43 Base de datos Firebase



Fuente: Autor

#### 6.4.2. Creación de proyecto

Para la creación de proyecto se utiliza Visual Code como entorno de trabajo, utilizando Node con TypeScript para la realizar el mapeo de todas las clases que están representadas en el diagrama de clases

#### 6.4.3. Conexión con Firebase

Para hacer la conexión con la base de datos se instaló un paquete de npm con el siguiente código en la terminal de trabajo: **“npm install firebase”** teniendo ya el paquete instalado se creó un archivo que se llama **“environment.ts”**, donde está la configuración la cual Firebase la facilita al crea la aplicación web.

Figura 44 Conexión Firebase

```
export const environment = {
  firebase: {
    projectId: 'crediautos-53fae',
    appId: 'xxxxxxxxxxxxxxxxxxxxx',
    storageBucket: 'crediautos-53fae.appspot.com',
    locationId: 'us-central',
    apiKey: 'xxxxxxxxxxxxxxxxxxxxx',
    authDomain: 'crediautos-53fae.firebaseio.com',
    messagingSenderId: 'xxxxxxxxxxxxxxxxxxxxx',
  },
  production: false,
  firebaseConfig: {
    apiKey: "xxxxxxxxxxxxxxxxxxxxx",
    authDomain: "crediautos.firebaseio.com",
    projectId: "crediautos",
    storageBucket: "crediautos.appspot.com",
    messagingSenderId: "xxxxxxxxxxxxxxxxxxxxx",
    appId: "xxxxxxxxxxxxxxxxxxxxx"
  }
}
```

Fuente: Autor

#### 6.4.4. Creación de servicios

Los servicios son los que hacen las consultas a la base de datos y para esto se creó para cada modelo su propio CRUD como se muestra en las siguientes Figuras

Figura 45 Servicio de Artículo

```
10 export class ArticuloService {
11
12   articulos: Observable<Articulo[]>;
13
14   private vehicleCollection: AngularFirestoreCollection<Articulo>;
15
16   constructor(private readonly afs: AngularFirestore) {
17     this.vehicleCollection = afs.collection<Articulo>('articulos');
18     this.getArticulos();
19   }
20
21   onDeleteArticulos(vehicleId: string): Promise<void>{
22     return new Promise(async (resolve, reject)->{
23       try {
24         const result = await this.vehicleCollection.doc(vehicleId).delete();
25         resolve(result);
26       } catch (error) {
27         reject(error.message);
28       }
29     });
30   }
31 > onSaveArticulo(vehiculo: Articulo, vehiculoId: string): Promise<void>{...
43   }
44 > private getArticulos(): void{...
48   }
49 > getArticulosById(id:string){...
51   }
52 > getArticuloById(id:string){...
56   }
57 > getArticuloByPromo(){...
61   }
62 > getArticuloByCategoria(categoria: string){...
66   }
67 > getArticulo(){...
71   }
72 }
73
```

Fuente: Autor

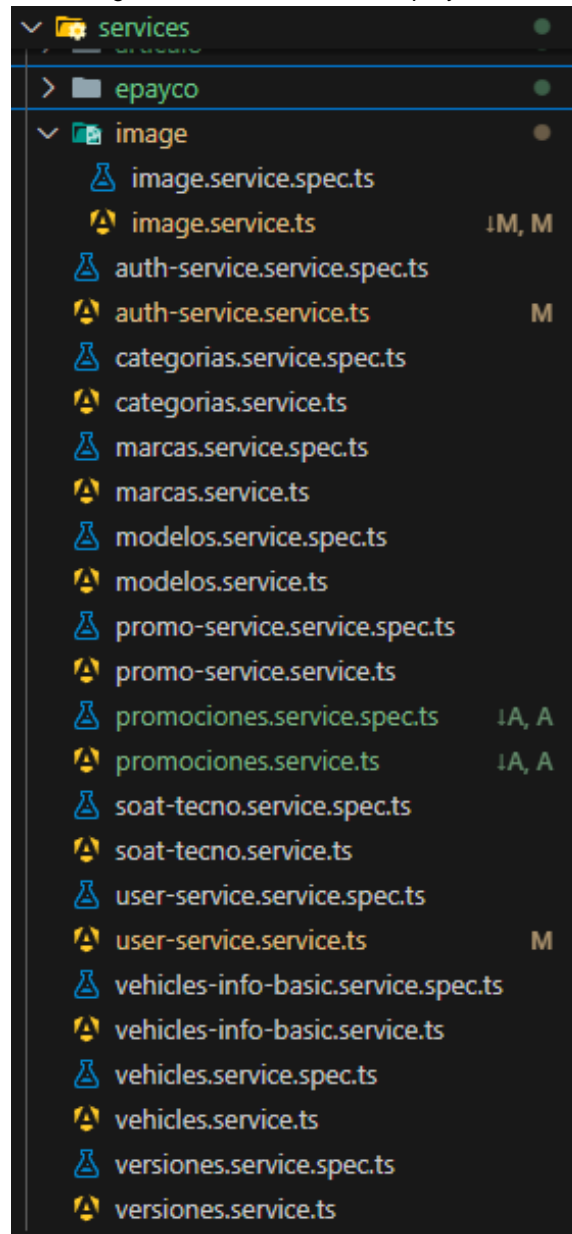
Figura 46 Servicio de auth

```
10 export class AuthServiceService {
14   constructor(private auth: AngularFireAuth) {
15     auth.authState.subscribe( user => {
18     })
19   }
20   getUser(){
21     |
22     |   return this.auth.authState;
23     |
24   }
25   login(user: string, pass: string): Promise<any>{
26     |   return this.auth.auth.signInWithEmailAndPassword(user, pass);
27     |
28   }
29   registrar(user: string, pass: string){
30     |   return this.auth.auth.createUserWithEmailAndPassword(user, pass);
31     |
32   }
33   logout(){
34     |   window.localStorage.setItem("isLoggedIn", "false");
35     |
36     |   return this.auth.auth.signOut();
37     |
38   }
39
40   verificarEmail() {
41     |   You, hace 1 segundo * Uncommitted changes
42     |   if(this.auth.auth.currentUser){
43     |     |   this.auth.auth.currentUser.sendEmailVerification();
44     |     |
45     |     |
46     |     |
47     |     |
48     |   googleAuth(){
49     |     |   this.auth.auth.signInWithPopup(new firebase.auth.GoogleAuthProvider());
50     |     |   firebase.auth.Auth.Persistence.LOCAL
51     |     |
52     |   }
```

Fuente: Autor

Toda la creación de servicio se realizó con las demás entidades como se puede observar en los siguientes archivos:

Figura 47 servicios de todo el proyecto

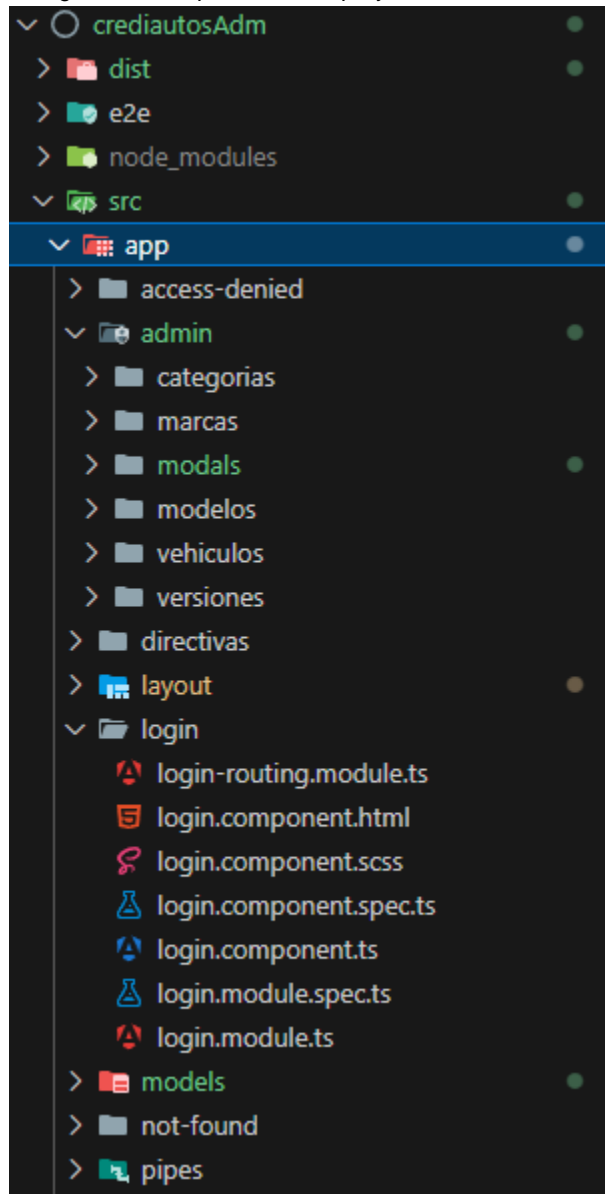


Fuente: Autor

#### 6.4.5. Creación del módulo administrativo

Para la creación del módulo administrativo se utilizó el framework Angular ya facilita todo el tema visual o de diseño utilizando Bootstrap, como Angular se divide por componentes podemos reutilizar código en varios segmentos del proyecto. Para esto se empezó con crear el proyecto y se empezó a dividir por componentes teniendo en cuenta los UML ya realizados.

Figura 48 Componentes del proyecto administrativo



Fuente: Autor

Se crearon los modelos según los diagramas UML ya realizados teniendo en cuenta los atributos como se muestra en la siguiente Figura 49

```
crediautosAdm > src > app > models > vehicle.interface.ts > Vehicle
You, hace 1 segundo | 3 authors (Snards and others)
1 export interface Vehicle {
2   id?: string;
3   numPuertas: number;
4   precio: number;
5   color: string;
6   vendedor: String;
7   urlimg: String;
8   tiendaoficial: Boolean;
9   Ubicacion: String;
10  carroceria: String;
11  modelo: String;
12  direccion: String;
13  condicion: number;
14  otrasCaracteristicas?: String[];
15  kilometraje: number;
16  transmision: number;
17  traccion: number;
18  marca: String;
19  year: number;
20  tcombustible: string;
21  financiable: Boolean;
22  tipoventa?: String;
23  categoria: String;
24  placa: String;
25  unicondueño: boolean;
26  gps?: boolean;
27  asientotabatible?: boolean;
28  techocorredizo?: boolean;
29  climatizador?: boolean;
30  portaequipajet?: boolean;
31  camarareversa?: boolean;
32  banca?: boolean;
33  abs?: boolean;
34  alarma?: boolean;
35  exploradorasd?: boolean;
36  airbag?: boolean;
37  sensorlluvia?: boolean;
38  exploradorast?: boolean;
```

Figura 49 Modelo de vehículo  
Fuente: Autor

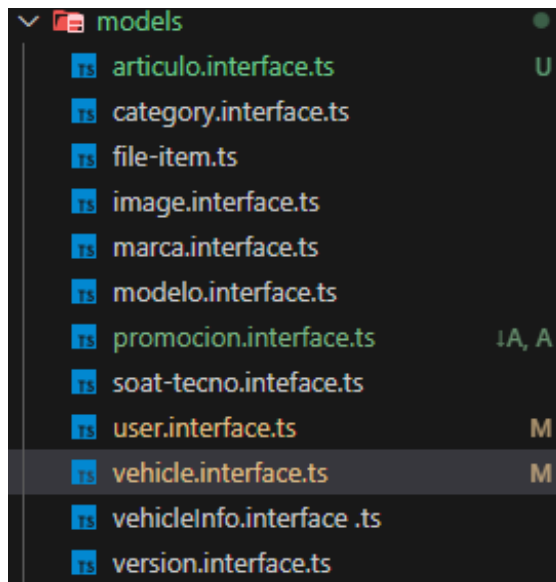


Figura 50 Modelos del entorno administrativo  
Fuente: Autor

Dentro de cada componente se divide en archivos los cuales son: componente.html el cual es el encargado de hacer la maqueta para el sitio web, el siguiente es el componete.css en este va todo el diseño que pueda llevar la plantilla del .html el otro archivo es el encargado de llevar la lógica, es donde va el código que llama a los controladores o para visualizar datos en la plantilla, todo esto se puede visualizar mejor en la Figura 51 a la Figura 54.

Figura 51 Archivo html del componente list-marca

```
crediautosAdm > src > app > admin > marcas > list-marcas > list-marcas.component.html > ng-template
Go to component | Snards, hace 2 años | 2 authors (Harold.parra and others)
1 <div class="row">
2   <div class="col-12">
3     <div class="info">
4       <input class="form-control" type="text" name="filterMarcas" placeholder="buscar Marca" [(ngModel)]="filterMarcas">
5       <mat-card *ngFor="let marca of marcas | name:filterMarcas | paginate: {id:'listar', itemsPerPage: 10, currentPage: pageActual } ">
6         <mat-card-title>{{ marca.name }}</mat-card-title>
7         <mat-card-description>{{ marca.category }}</mat-card-description>
8         <mat-card-actions align="end">
9           <button mat-button (click)="onEditar(marca.id, editmodal, marca.name, marca.category)">Editar</button>
10          <button mat-button (click)="openDeleteModal(deleteModal, marca.id, marca.name)">Eliminar</button>
11        </mat-card-actions>
12      </mat-card>
13      <pagination-controls id="listar" (pageChange)="pageActual = $event" responsive="true">
14    </pagination-controls>
15  </div>
16 </div>
17 </div>
18 > <ng-template #editmodal>...
46 </ng-template>
47 > <ng-template #deleteModal>...
64 </ng-template>
65 > <ng-template #confirModal>...
69 </ng-template> Harold.parra, hace 3 años • marcas ok
```

Fuente: Autor

Figura 52 Archivo CSS del componente list-marca

```
crediautosAdm > src > app > admin > marcas > list-marcas > list-marcas.component.css
You, hace 3 años | 2 authors (Harold.parra and others)
1 .info{
2   display: flex;
3   flex-direction: column;
4   justify-content: center;
5   width: 60%;
6   margin-left: auto;
7   margin-right: auto;
8 }
9 /* estilos modal */
```

Fuente: Autor

Figura 53 Archivo ts del componente list-marca

```
crediautosAdm > src > app > admin > marcas > list-marcas > list-marcas.component.ts > ...
12 export class ListMarcasComponent implements OnInit {
13     categories: any[];
14     marcaForm: FormGroup;
15     deletemarca: string;
16     marcaDeleteid: string;
17     marcaeditid: string;
18     isloading=false;
19     icon=false;
20     pageActual: number = 1;
21     filterMarcas='';
22
23
24     constructor(private marcaService: MarcasService, private dialog: NgbModal,
25                 private fb: FormBuilder,
26                 private categoriaService: CategoriasService) {
27         this.marcaService.getMarcaByCategoria('Carros y Camionetas').subscribe(resp => {
28             this.marcas = resp;
29         })
30         this.categoriaService.categories.subscribe(resp => {
31             this.categories = resp;
32         })
33     }
34
35     ngOnInit(): void {
36     }
37
38     onEditar(id:string, editmodal, marca:string, categoria:string){
39         this.marcaeditid = id;
40         this.initForm();
41         marca != null ?this.marcaForm.get('name').setValue(marca): "";
42         categoria != null ? this.marcaForm.get('category').setValue(categoria): "";
43
44         this.dialog.open(editmodal)
45     }
46 }
47
48 > openDeleteModal(deleteModal, marcaId:string, marca:string){...
52 }
```

Fuente: Autor

Con la creación de todos los componentes y módulos del proyecto del aplicativo web para administrador, según los requerimientos se tienen que configurar las rutas para llamar cada componente hijo con su respectivo módulo padre.

Figura 54 Rutas de navegación

```

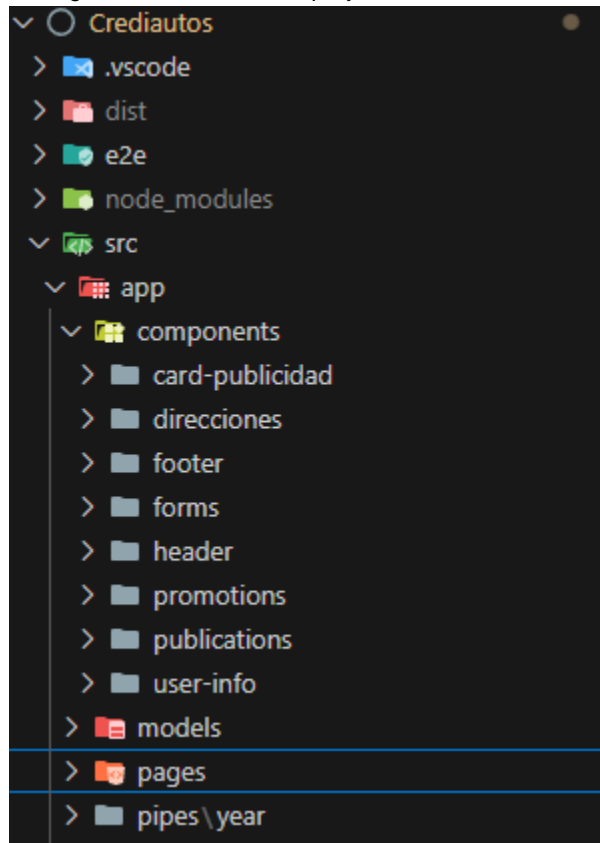
crediautosAdm > src > app > layout > layout-routing.module.ts > ...
3 import { LayoutComponent } from './layout.component';
4
5 const routes: Routes = [
6   {
7     path: '',
8     component: LayoutComponent,
9     children: [
10      { path: '', redirectTo: 'dashboard', pathMatch: 'prefix' },
11      { path: 'dashboard', loadChildren: () => import('./dashboard/dashboard.module').then(m => m.DashboardModule) },
12      {
13        path: 'dashboardnetworking',
14        loadChildren: () => import('./dashboardnetworking/dashboardnetworking.module').then(m => m.DashboardNetworkingModule)
15      },
16      { path: 'charts', loadChildren: () => import('./charts/charts.module').then(m => m.ChartsModule) },
17      { path: 'tables', loadChildren: () => import('./tables/tables.module').then(m => m.TablesModule) },
18      { path: 'forms', loadChildren: () => import('./form/form.module').then(m => m.FormModule) },
19      { path: 'bs-element', loadChildren: () => import('./bs-element/bs-element.module').then(m => m.BsElementModule) },
20      { path: 'grid', loadChildren: () => import('./grid/grid.module').then(m => m.GridModule) },
21      { path: 'components', loadChildren: () => import('./bs-component/bs-component.module').then(m => m.BsComponentModule) },
22      { path: 'blank-page', loadChildren: () => import('./blank-page/blank-page.module').then(m => m.BlankPageModule) },
23      { path: 'usuario', loadChildren: () => import('./components/modules/usuario/usuario.module').then(m => m.UsuarioModule)},
24      { path: 'clientes', loadChildren: () => import('./components/modules/clientes/clientes.module').then(m => m.ClientesModule)},
25      { path: 'soatTecno', loadChildren: () => import('./components/modules/soat-tecno-clientes/soat-tecno-clientes.module').then(m => m.SoaTModule)},
26      { path: 'ventas', loadChildren: () => import('./components/modules/ventas/ventas.module').then(m => m.VentasModule)},
27      { path: 'productos', loadChildren: () => import('./components/modules/productos/productos.module').then(m => m.ProductosModule)},
28      { path: 'category', loadChildren: () => import('./admin/categorias/categorias.module').then(m => m.CategoriasModule)},
29      { path: 'marcas', loadChildren: () => import('./admin/marcas/marcas.module').then(m => m.MarcasModule)},
30      { path: 'modelos', loadChildren: () => import('./admin/modelos/modelos.module').then(m => m.ModelosModule)},
31      { path: 'versiones', loadChildren: () => import('./admin/versiones/versiones.module').then(m => m.VersionesModule)},
32      { path: 'vehiculos', loadChildren: () => import('./admin/vehiculos/vehiculos.module').then(m => m.VehiculosModule)},
33      { path: "editar/vehicle", loadChildren: () => import('./admin/vehiculos/new-vehicles/new-vehicles.module').then(m => m.NewVehi)},
34      { path: "ver/vehicle", loadChildren: () => import('./admin/vehiculos/new-vehicles/new-vehicles.module').then(m => m.NewVehi)}
35    ]
36  }
37 ];
    
```

Fuente: Autor

#### 6.4.6. Creación del módulo de usuario final

Como se utilizó Angular para el entorno administrativo, también se usó en el entorno de usuario final, por lo cual, se realizaron los mismos pasos del punto anterior, iniciando con la creación del proyecto en Angular y la estructuración de los componentes.

Figura 55 Estructura del proyecto de usuario final



Fuente: Autor

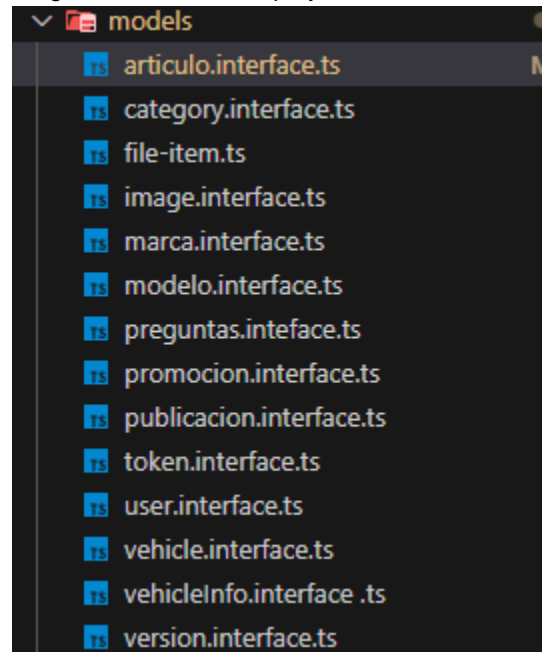
Teniendo ya la estructura del proyecto según el patrón MVC se prosiguió a crear los modelos de cada entidad necesaria para el entorno de usuario final.

Figura 56 Modelo de artículo

```
Crediautos > src > app > models > articulo.interface.ts > Artículo
Snards, hace 20 meses | 1 author (Snards)
1  export interface Artículo {
2      id:string;
3      valor:number;
4      name: string;
5      desc: string;
6      urlimg: string;
7      vendedor: string;
8      status: boolean;
9      promocion: boolean;
10     categoria: string;
11     state: string;
12 }
```

Fuente: Autor

Figura 57 Modelos del proyecto de usuario final



Fuente: Autor

Ya teniendo los modelos necesarios se realiza la parte visual. Esto comprende la creación de componentes y el desarrollo de la parte visual de la página web como se puede ver en las Figura 58 a la Figura 65.

Figura 58 Plantilla del Inicio

```
Crediautos > src > app > pages > inicio > inicio > inicio.component.html > div.row.publicidad > div.col-12
Go to component | ...
1 <app-header></app-header>
2 <app-banner></app-banner>
3 <app-search></app-search>
4 <app-banner></app-banner>
5 <div class="row publicidad">
6   <div class="col-12">
7     <h4>Publicidad</h4>
8   </div>
9 </div>
10 </div>
11 <div>
12   <app-featured></app-featured>
13   <app-featurednew></app-featurednew>
14   <app-featuredarticle></app-featuredarticle>
15 </div>
16
17 <app-stors></app-stors>
18   <app-wanted></app-wanted>
19
20 <div class="row publicidad">
21   <div class="col-12">
22   </div>
23 </div>
24 <app-wanted-motos></app-wanted-motos>
25 <div class="row publicidad">
26   <div class="col-12">
27   </div>
28 </div>
29 <app-wanted-camiones></app-wanted-camiones>
30 <app-footer-p></app-footer-p>
31
```

Fuente: Autor

Figura 59 Estilos de la página de inicio

```
Crediautos > src > app > pages > inicio > inicio > inicio.component.css > ..
...
1  .publicidad{
2      align-items: center;
3      height: 50px;
4      margin-top: 40px;
5      margin-bottom: 40px;
6      text-align: center;
7
8  }
9  .fondoRedBlack{
10     background-color: #8c0303
11 }
12 .h6{
13     font-size: 1.5rem !important;
14 }
```

Fuente: Autor

Figura 60 Lógica de la página de inicio

```
Crediautos > src > app > pages > inicio > inicio > inicio.component.ts > InicioComponent
...
1  import { Component, OnInit } from '@angular/core';
2
3  ...
4  @Component({
5      selector: 'app-inicio',
6      templateUrl: './inicio.component.html',
7      styleUrls: ['./inicio.component.css']
8  })
9  export class InicioComponent implements OnInit {
10
11     constructor() { }
12
13     ngOnInit(): void {
14     }
15 }
16
```

Fuente: Autor

Se sigue con la creación de componentes y módulos padres de cada sección que se necesita en el proyecto. A continuación, se aprecian algunos fragmentos de código de dichos componentes.

Figura 61 Plantilla de la creación de un vehículo

```
Crediautos > src > app > pages > admin > vehicle-info > vehicle-info.component.html > div.body > div.body-page > div.title
Go to component | ...
1 <app-header></app-header>
2 <div class="body">
3   <div class="body-page">
4     <div class="title">
5       <h5>Agregar informacion de un vehiculo</h5>
6     </div>
7     <div class="form">
8       <form [formGroup]="vehicleForm">
9         <div class="row">
10          <div class="col-6">
11            <div class="form-floating mb-3">
12              <input type="text" class="form-control" id="floatingInput" placeholder="Marca" formControlName="marca">
13              <label for="floatingInput">Marca</label>
14            </div>
15          </div>
16          <div class="col-6">
17            <div class="form-floating mb-3">
18              <input type="text" class="form-control" id="floatingInput" placeholder="Modelo" formControlName="modelo">
19              <label for="floatingInput">Modelo</label>
20            </div>
21          </div>
22        </div>
23        <div class="row">
24          <div class="col-6">
25            <div class="form-floating mb-3">
26              <input type="text" class="form-control" id="floatingInput" placeholder="Version" formControlName="version">
27              <label for="floatingInput">Version</label>
28            </div>
29          </div>
30          <div class="col-6">
31            <div class="form-floating mb-3">
32              <input type="number" class="form-control" id="floatingInput" placeholder="Año de lanzamiento" formControlName="año">
33              <label for="floatingInput">Año</label>
34            </div>
35          </div>
36        </div>
37        <div class="row">
38          <div class="col-6">
```

Fuente: Autor

Figura 62 Estilo para la creación de un vehículo

```
Crediautos > src > app > pages > admin > vehicle-info > vehicle-info.component.css > .title
...
1 .title{
2   margin-left: auto;
3   margin-right: auto;
4   width: 50%;
5   margin: 20px;
6   display: flex;
7   flex-direction: row;
8   justify-content: center;
9   align-items: center;
10  padding: 20px;
11 }
12 .info-title{
13   width: 100%;
14 }
15 .info-title a{
16   display: flex !important;
17   flex-direction: row;
18   justify-content: space-between;
19 }
20 .body{
21   background-color: #F4F5FF;
22 }
23 .body-page{
24   justify-content: center;
25   width: 80%;
26   margin-left: auto;
27   margin-right: auto;
28 }
29 .form{
30   background-color: white;
31   width: 60%;
32   padding: 40px;
33   border-radius: 8px;
34   display: flex;
35   flex-direction: column;
36   justify-content: center;
37   margin-left: auto;
38 }
```

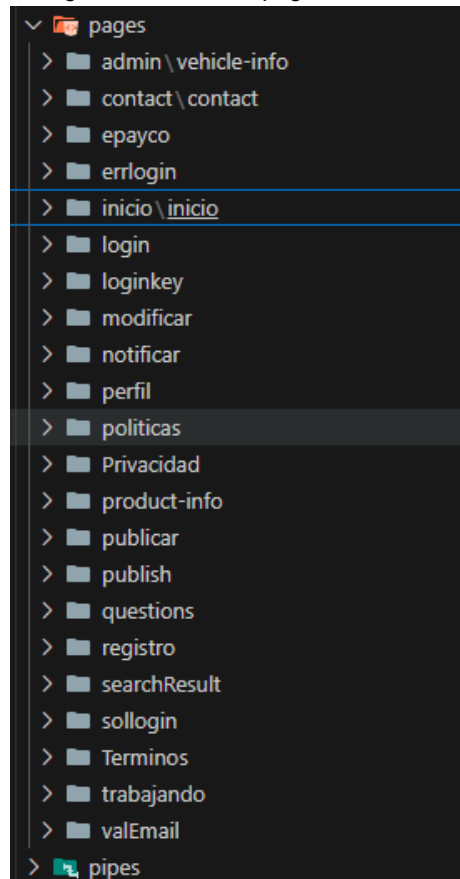
Fuente: Autor

Figura 63 Lógica para la creación de un vehículo

```
10 }
11 export class VehicleInfoComponent implements OnInit {
12
13     vehicleForm: FormGroup;
14
15     categorias:any[];
16     constructor(private fb: FormBuilder, private categoriaService: CategoriesService,
17                 private vehicle: VehicleInfoService) {
18         this.initForm();
19         this.getCategorias();
20     }
21
22     ngOnInit(): void {
23     }
24
25     initForm(){
26         this.vehicleForm= this.fb.group({
27             marca: new FormControl('', [Validators.required]),
28             modelo: new FormControl('', [Validators.required]),
29             puertas: new FormControl('', [Validators.required]),
30             carroceria: new FormControl('',[Validators.required]),
31             direccion: new FormControl('', [Validators.required]),
32             trasmision: new FormControl('', [Validators.required]),
33             traccion: new FormControl('', [Validators.required]),
34             year: new FormControl('', [Validators.required]),
35             combustible: new FormControl('', [Validators.required]),
36             categoria: new FormControl('', [Validators.required]),
37             version: new FormControl('', [Validators.required]),
38             gps: new FormControl(''),
39             asientotabitable: new FormControl(''),
40             techocorredizo: new FormControl(''),
41             climatizador: new FormControl(''),
42             portaequipajet: new FormControl(''),
43             camarareversa: new FormControl(''),
44             banca: new FormControl(''),
45             abs: new FormControl(''),
46             alarma: new FormControl(''),
47             exploradorasd: new FormControl(''),
48             alphan: new FormControl('')
```

Fuente: Autor

Figura 64 Todas las páginas creadas



Fuente: Autor

Habiendo desarrollado todas las secciones del aplicativo web, se pueden configurar o codificar las rutas para poder consumir todos los módulos padres con sus respectivos componentes hijos.

Figura 65 Rutas del proyecto usuario final

```
Crediautos > src > app > app-routing.module.ts > ...
3 import { PagoEpaycoComponent } from './pages/epayco/pago-epayco/pago-epayco.component';
4 import { ResumenComponent } from './pages/perfil/perfil/resumen/resumen.component';
5 import { ArticuloInfoComponent } from './pages/product-info/articulo-info/articulo-info.component';
6 import { ProductInfoComponent } from './pages/product-info/product-info.component';
7 import { SearchResultComponent } from './pages/searchResult/search-result/search-result.component';
8
9 const routes: Routes = [
10
11   Haga clic para agregar un punto de interrupción en: () => import('./pages/registro/registro.module').then(m => m.RegistroModule) },
12   { path: 'inicio', loadChildren: () => import('./pages/inicio/inicio/inicio.module').then(m => m.InicioModule) },
13   { path: 'login', loadChildren: () => import('./pages/login/login/login.module').then(m => m.LoginModule) },
14   { path: 'questions', loadChildren: () => import('./pages/questions/questios/questios.module').then(m => m.QuestiosModule) },
15   { path: 'publish', loadChildren: () => import('./pages/publish/publish-car/publish-car.module').then(m => m.PublishCarModule) },
16   { path: 'contact', loadChildren: () => import('./pages/contact/contact/contact.module').then(m => m.ContactModule) },
17   { path: 'modificar', loadChildren: () => import('./pages/modificar/anuncio/anuncio.module').then(m => m.AnuncioModule) },
18   { path: 'notificar', loadChildren: () => import('./pages/notificar/anuncio/anuncio.module').then(m => m.AnuncioModule) },
19   { path: 'terminos', loadChildren: () => import('./pages/Terminos/condiciones/condiciones.module').then(m => m.CondicionesModule) },
20   { path: 'politicas', loadChildren: () => import('./pages/politicas/politicas.module').then(m => m.PoliticasModule) },
21   { path: 'privacidad', loadChildren: () => import('./pages/Privacidad/privacidad/privacidad.module').then(m => m.PrivacidadModule) },
22   { path: 'errlog', loadChildren: () => import('./pages/errlogin/errlogin/errlogin.module').then(m => m.ErrloginModule) },
23   { path: 'sollog', loadChildren: () => import('./pages/sollogin/sollogin/sollogin.module').then(m => m.SolloginModule) },
24   { path: 'buscar', component: SearchResultComponent },
25   { path: 'trabajando', loadChildren: () => import('./pages/trabajando/trabajando/trabajando.module').then(m => m.TrabajandoModule) },
26   { path: 'loginkey', loadChildren: () => import('./pages/loginkey/loginkey/loginkey.module').then(m => m.LoginkeyModule) },
27   { path: 'valemil', loadChildren: () => import('./pages/valEmail/val-email/val-email.module').then(m => m.ValEmailModule) },
28   { path: 'perfil', loadChildren: () => import('./pages/perfil/perfil/perfil.module').then(m => m.PerfilModule) },
29   { path: 'cardpublicidad', loadChildren: () => import('./components/card-publicidad/card-publicidad.module').then(m => m.CardPublicid
30   { path: 'publicarForm', loadChildren: () => import('./pages/publicar/form/publicarform/publicarform.module').then(m => m.Publicar-for
31   { path: 'producto/:id/:tipo', component: ProductInfoComponent },
32   { path: 'articulo/:id/:tipo', component: ArticuloInfoComponent },
33   { path: 'admVehicle', loadChildren: () => import('./pages/admin/vehicle-info/vehicle-info.module').then(m => m.VehicleInfoModule) },
34   { path: 'categoryform', loadChildren: () => import('./components/forms/category-form/category-form.module').then(m => m.CategoryForm
35   { path: 'marcaform', loadChildren: () => import('./components/forms/marca-form/marca-form.module').then(m => m.MarcaFormModule) },
36   { path: 'modeloform', loadChildren: () => import('./components/forms/modelos-form/modelos-form.module').then(m => m.ModelosFormModul
37   { path: 'versionform', loadChildren: () => import('./components/forms/version-form/version-form.module').then(m => m.VersionFormModu
38   { path: 'userInfo', loadChildren: () => import('./components/user-info/user-info.module').then(m => m.UserInfoModule) },
39   { path: 'pago', component: PagoEpaycoComponent },
40   { path: '', pathMatch: 'full', redirectTo: '/inicio' },
```

Fuente: Autor

#### 6.4.7. Creación de la aplicación móvil

Cumpliendo con uno de los objetivos propuestos la creación del aplicativo móvil se decidió que se desarrollarlo utilizando Flutter como framework, ya que hace el desarrollo más ágil y se pueden ver los cambios en tiempo real en un dispositivo o emulador de Android.

```
crediautoscolombia > lib > main.dart > MyApp > build
Snards, hace 2 años | 1 author (Snards)
1 import 'package:crediautoscolombia/src/bloc/Provaider.dart';
2 import 'package:crediautoscolombia/src/config/routes.dart';
3 import 'package:crediautoscolombia/src/preferencias_usuario/preferencias_usuario.dart';
4 import 'package:flutter/material.dart';
5 import 'package:crediautoscolombia/src/themes/theme.dart';
6 import 'package:google_fonts/google_fonts.dart';
7 import 'package:firebase_core/firebase_core.dart';
8
Run | Debug | Profile
9 void main() async {
10
11   WidgetsFlutterBinding.ensureInitialized();
12   final prefs = new PreferenciasUsuario();
13   await Firebase.initializeApp();
14   await prefs.initPrefs();
15   runApp(MyApp());
16 }
17
Snards, hace 2 años | 1 author (Snards)
18 class MyApp extends StatelessWidget {
19   // This widget is the root of your application.
20   @override
21   Widget build(BuildContext context) {
22     return Provider(child: MaterialApp(
23       title: 'Crediautos',
24       theme: AppTheme.lightTheme.copyWith(
25         textTheme: GoogleFonts.muliTextTheme(
26           Theme.of(context).textTheme,
27         )),
28       debugShowCheckedModeBanner: false,
29       routes: Routes.getRoute(),
30       initialRoute: "MainPage",
31     )); // MaterialApp // Provider
```

Figura 66 Widget raíz archivo main.dart  
Fuente: Autor

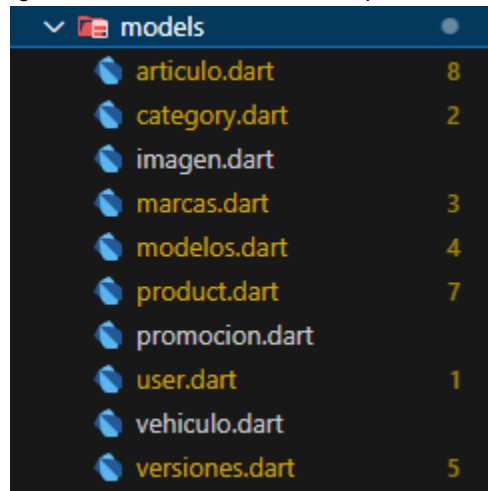
Con la creación del proyecto en flutter, se crean los modelos necesarios para el aplicativo móvil como se puede observar en la Figura 67 y Figura 68.

Figura 67 Modelo de Category

```
crediautoscolombia > lib > src > models > category.dart > ...  
Snards, hace 2 años | 1 author (Snards)  
1 import 'dart:convert'; Snards, hace 2 años • publicar  
2   
Snards, hace 2 años | 1 author (Snards)  
3 class Category {  
4     Category({  
5         required this.id,  
6         required this.name,  
7     });  
8  
9     String id;  
10    String name;  
11  
12    factory Category.fromJson(String str) => Category.fromMap(json.decode(str));  
13  
14    String toJson() => json.encode(toMap());  
15  
16    factory Category.fromMap(Map<String, dynamic> json) => Category(  
17        id: json["id"] == null ? null : json["id"],  
18        name: json["name"] == null ? null : json["name"],  
19    );  
20  
21    Map<String, dynamic> toMap() => {  
22        "id": id == null ? null : id,  
23        "name": name == null ? null : name,  
24    };  
25 }
```

Fuente: Autor

Figura 68 Todos los modelos del aplicativo móvil

A screenshot of a mobile application's interface showing a list of models under the heading 'models'. The list contains ten entries, each with a blue icon, a filename, and a number. The entries are: 'articulo.dart' (8), 'category.dart' (2), 'imagen.dart', 'marcas.dart' (3), 'modelos.dart' (4), 'product.dart' (7), 'promocion.dart', 'user.dart' (1), 'vehiculo.dart', and 'versiones.dart' (5).

Model Name	Count
articulo.dart	8
category.dart	2
imagen.dart	
marcas.dart	3
modelos.dart	4
product.dart	7
promocion.dart	
user.dart	1
vehiculo.dart	
versiones.dart	5

Fuente: Autor

Posteriormente, se sigue con la creación de proveedores necesarios para el aplicativo. En la siguiente Figura 69 se puede observar cómo se crearon los proveedores.

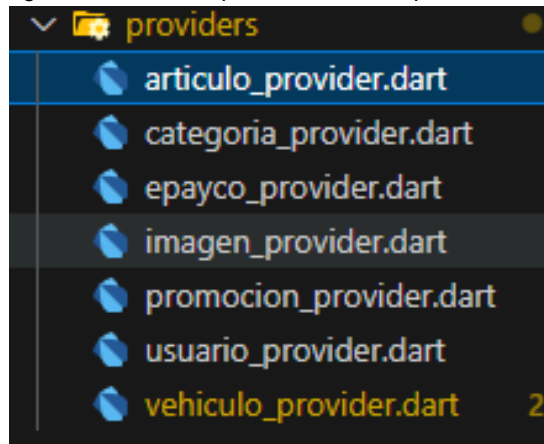
Figura 69 Proveedor de artículos

```
crediautoscolombia > lib > src > providers > articulo_provider.dart > ...
5
Snards, hace 2 años | 1 author (Snards)
6 class ArticuloProvider {
7   ArticuloProvider() {
8     print("inicializa Articulo");
9     listArticulos = FirebaseFirestore.instance.collection('articulos');
10  }
11  late final CollectionReference listArticulos;
12
13  Future<Articulo> getArticulo() async {
14    final DocumentSnapshot snapshot =
15      await listArticulos.doc('iQLSVu9ZtDCg2HtTLTQK').get();
16    final Articulo articulo =
17      Articulo.fromMap(snapshot.data() as Map<String, dynamic>);
18    return articulo;
19  }
20
21  Future<List<Articulo>> getArticulos() async {
22    final QuerySnapshot<Object?> snapshot =
23      await listArticulos.where('status', isEqualTo: true).get();
24    List<Articulo> articulos = [];
25    for (var i = 0; i < snapshot.docs.length; i++) {
26      articulos.add(
27        Articulo.fromMap(snapshot.docs[i].data() as Map<String, dynamic>));
28    }
29    print("///// listArticulos: ${articulos.toString()}");
30    return articulos;
31  }
32
33  Future<List<Articulo>> getArticulosPrmocion() async {
34    final QuerySnapshot<Object?> snapshot = await listArticulos
35      .where('status', isEqualTo: true)
36      .where('promocion', isEqualTo: true)
37      .get();
38    List<Articulo> articulos = [];
39    for (var i = 0; i < snapshot.docs.length; i++) {
40      articulos.add(
41        Articulo.fromMap(snapshot.docs[i].data() as Map<String, dynamic>));
42    }

```

Fuente: Autor

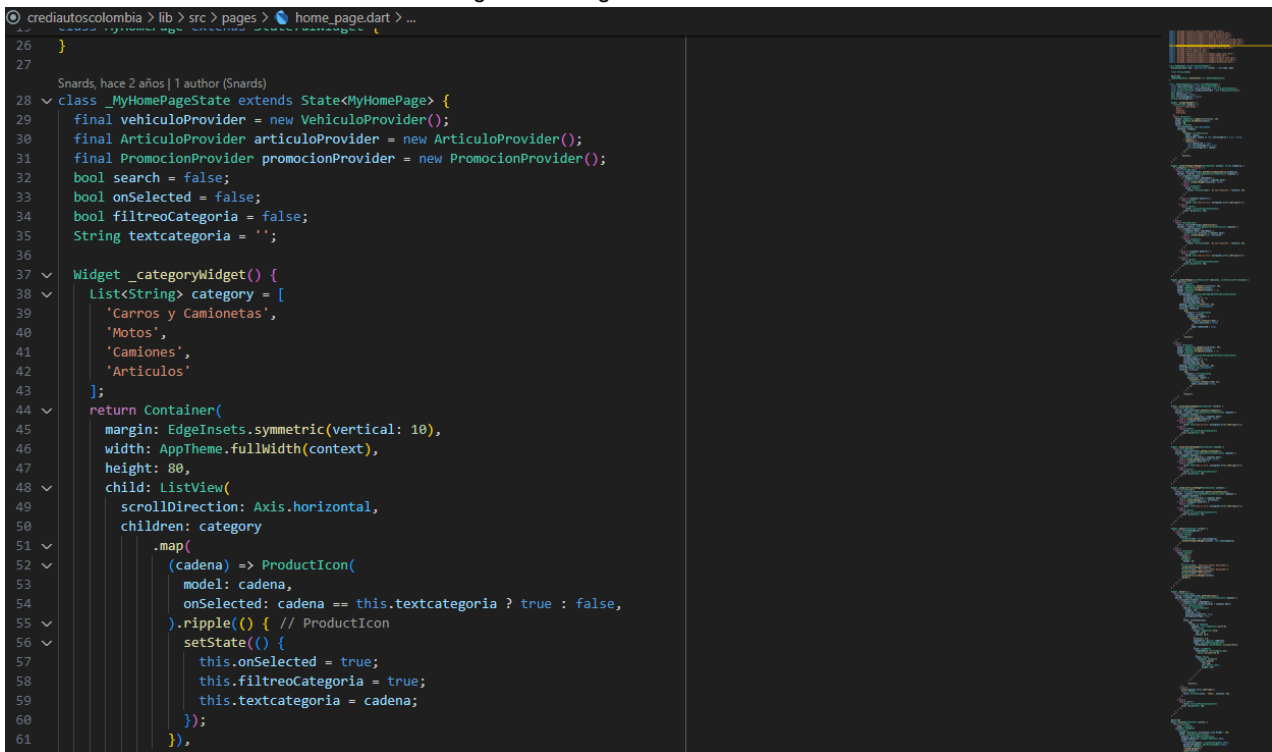
Figura 70 Todos los proveedores del aplicativo móvil



Fuente: Autor

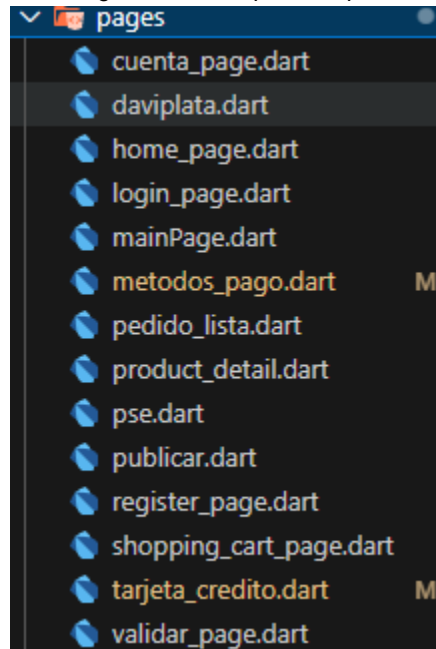
Se desarrollaron las vistas o páginas que tendría el aplicativo. A su vez se realizaron nuevos Widgets para la implementación en dichas páginas.

Figura 71 Página de inicio



Fuente: Autor

Figura 72 Páginas creadas para el aplicativo móvil



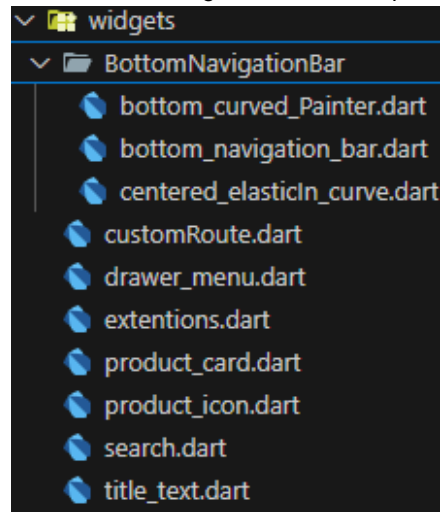
Fuente: Autor

Figura 73 Widget del menú

```
crediautoscolombia > lib > src > widgets > drawer_menu.dart > ...
3 import 'package:crediautoscolombia/src/utils/alerta.dart';
4 import 'package:flutter/material.dart';
5
6 // ignore: must_be_immutable
7 Snards, hace 24 meses | 1 author (Snards)
8 class DrawerMenu extends StatelessWidget {
9   final _pref = PreferenciasUsuario();
10  UsuarioProvider usuario = new UsuarioProvider();
11  @override
12  Widget build(BuildContext context) {
13    return Drawer(
14      child: ListView(
15        padding: EdgeInsets.zero,
16        children: <Widget>[
17          _buildDrawerHeader(),
18          _buildDrawerItem(Icons.home, 'Dashboard',
19            () => {Navigator.of(context).popAndPushNamed('/')},
20          _buildDrawerItem(Icons.home, 'Carrito',
21            () => {Navigator.of(context).popAndPushNamed('/card')}),
22          _buildDrawerItem(
23            Icons.send_to_mobile_rounded,
24            'Publicar',
25            () => {
26              if (this._pref.token.length > 3)
27                {Navigator.pushNamed(context, '/publicar')}
28              else
29                {mostrarAlertaLogin(context,
30                  'Para publicar un vehiculo debes iniciar sesión')}
31            }
32          ),
33          Divider(),
34          if (_pref.token == '')
35            _buildDrawerItem(Icons.account_circle, 'Inicia Sesión',
36              () => {Navigator.pushNamed(context, '/login')}),
37          if (_pref.token == '')
38            _buildDrawerItem(Icons.account_circle, 'Regístranse',
39              () => {Navigator.pushNamed(context, '/register')}),
```

Fuente: Autor

Figura 74 Todos los Widget desarrollados para la app



Fuente: Autor

Luego, se continuó con el enrutamiento de las páginas dentro de la app. Para esto se dejó aparte en la carpeta de configuración del proyecto.

Figura 75 Rutas de la app

```
crediautoscolombia > lib > src > config > routes.dart
11 import 'package:crediautoscolombia/src/widgets/customRoute.dart';
12 import 'package:flutter/material.dart';
13
14 class Routes {
15   static Map<String, WidgetBuilder> getRoute() {
16     return <String, WidgetBuilder>{
17       '/': (BuildContext context) => MainPage(title: 'main'),
18       '/detail': (BuildContext context) => ProductDetailPage(),
19       '/login': (BuildContext context) => LoginPage(),
20       '/search': (BuildContext context) => ShoppingCartPage(),
21       '/publicar': (BuildContext context) => PublicarPage(),
22       '/register': (BuildContext context) => RegisterPage(),
23       '/perfil': (BuildContext context) => CuentaPage(),
24       '/card': (BuildContext context) => Card(),
25       '/metopago': (BuildContext context) => MetodosPagoPage(),
26       '/validation': (BuildContext context) => ValidarPage ();
27     };
28   }
29
30   static Route<dynamic> onGenerateRoute(RouteSettings settings) {
31     if (settings.name == '/detail') {
32       return CustomRoute<bool>{
33         builder: (BuildContext context) => ProductDetailPage(),
34         settings: settings;
35       } else if (settings.name == '/login') {
36         return CustomRoute<bool>{
37           builder: (BuildContext context) => LoginPage(), settings: settings;
38         } else {
39           return CustomRoute<bool>{
40             builder: (BuildContext context) => MainPage(title: 'main'),
41             settings: settings;
42           }
43         }
44     }
45 }
```

Fuente: Autor

## 6.5. Fase 5. Pruebas

Tabla 18 Prueba 1

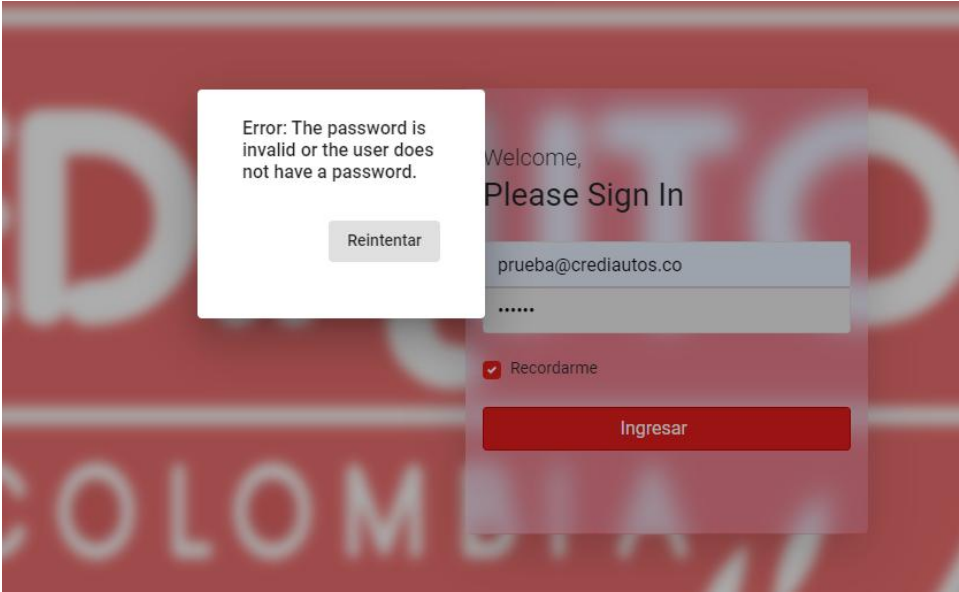
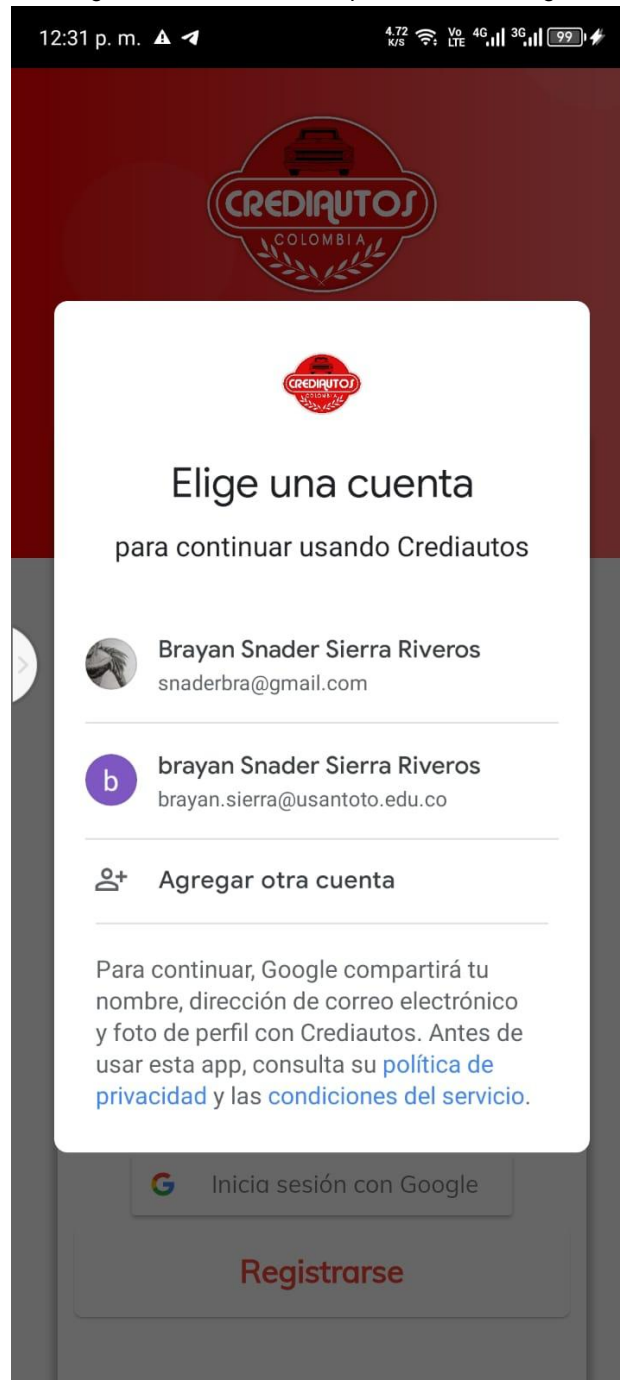
Prueba	
Id de Prueba	P1
Descripción	Se hacen pruebas de inicio de sesión de las aplicaciones web y móviles por medio de email y contraseña, como también por medio de inicio de sesión por Google
Resultado	Al querer iniciar sesión con un email o contraseña incorrecta el software valida bien y genera un mensaje de email o contraseña equivocada. Cuando se ingresa el email y la contraseña correcta el software direcciona al inicio del sistema ya con los datos del usuario. En el proceso de inicio de sesión por medio de Google el software obtiene la información de la cuenta de Google y se inicia sesión exitosamente.
Anexo	<p>Figura 76 Prueba de inicio de sesión con credenciales incorrectas</p>  <p>Fuente: Autor</p>

Figura 77 Inicio de sesión por medio de Google

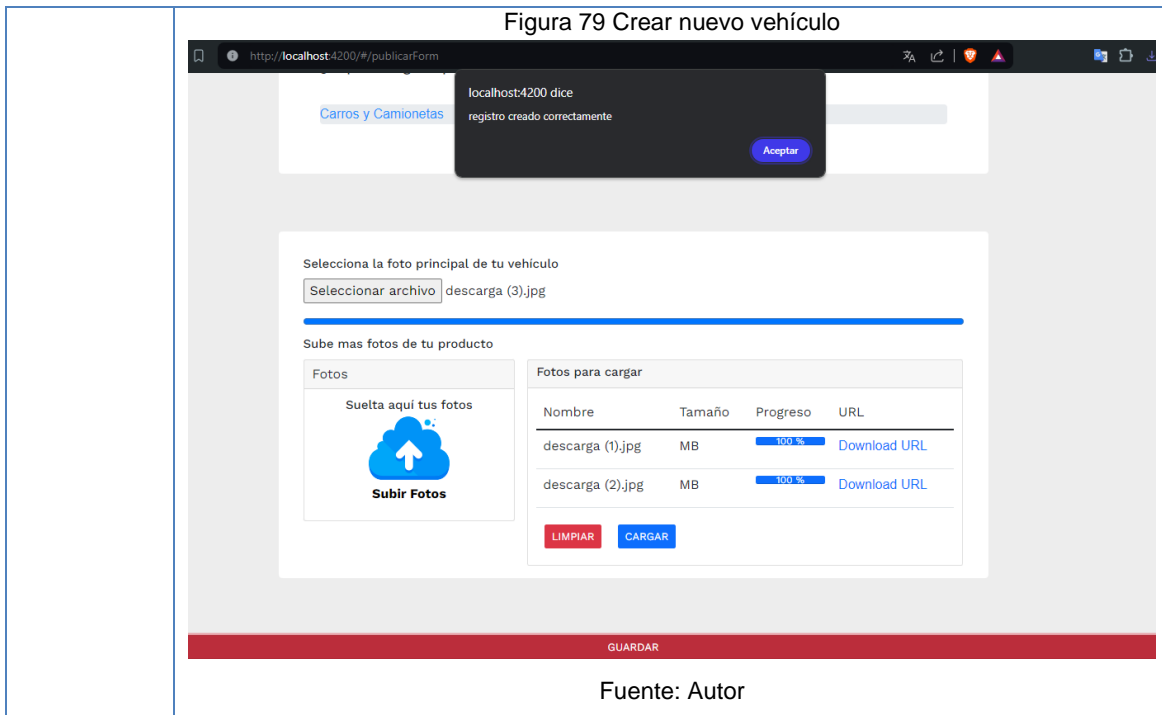


Fuente: Autor

Fuente: Autor

Tabla 19 Prueba P2

Prueba	
Id de Prueba	P2
Descripción	Se prueba la creación de publicaciones nuevas “vehículos y artículos”
Resultado	En la creación de una nueva publicación sea un vehículo o un artículo, primero valida que el usuario esté logueado, después de confirmar los campos requeridos u obligados para la creación de dicha publicación, lo siguiente es dar clic en el botón de guarda y muestra un mensaje de guardado con éxito
Anexo	<p>Figura 78 Prueba de publicar si estar logueado</p>  <p>Fuente: Autor</p>





Fuente: Autor

Fuente: Autor

Tabla 20 Prueba P3




Prueba	
Id de Prueba	P3
Descripción	Creación de nuevas Categorías, Marcas, Modelos y Versiones, esto es en el módulo administrativo
Resultado	Al agregar categorías, marca, modelos y/o Versiones las hace eficazmente sin ningún problema
Anexo	<p>Figura 80 Creación de una nueva versión</p> <p><b>Agregar nueva version</b></p> <p>Seleccionar categoria: Carros y Camionetas</p> <p>Seleccionar marca: Dacia</p> <p>Seleccionar modelo: 1410</p> <p>Nombre de la version: 2024</p> <p><b>Guardar</b></p>
	Fuente: Autor

Figura 81 La nueva versión creada					
Listar		Nuevo			
20					
Categoría	Marca	Modelo	Version	Accion	
Carros y Camionetas	Dacia	1410	2024	 	

Fuente: Autor

Fuente: Autor

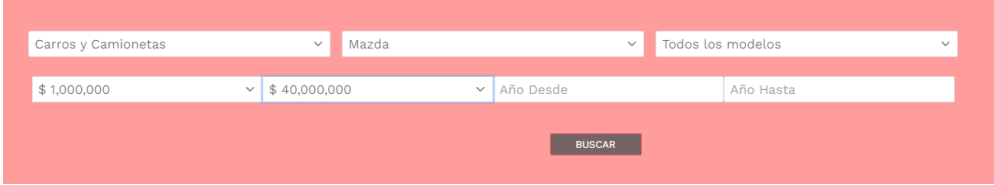
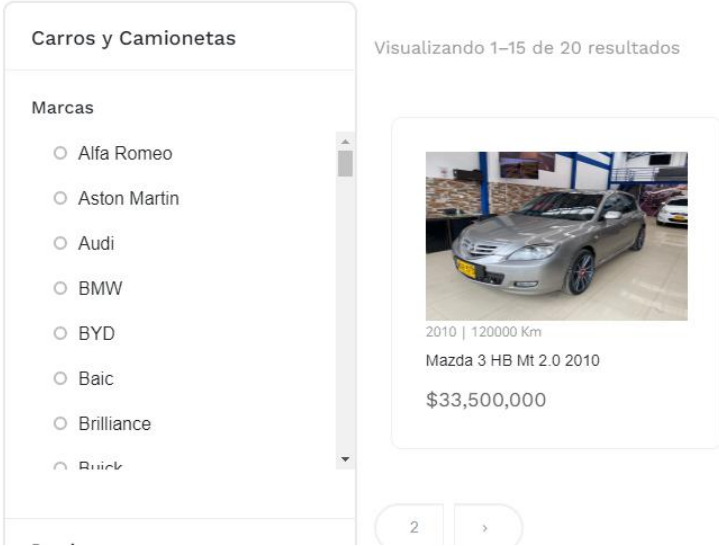
Tabla 21 Prueba P4

Prueba															
Id de Prueba	P4														
Descripción	Creación de usuarios en el módulo administrativo														
Resultado	Al digitar todos los campos requeridos y darle clic al botón guardar se crea el usuario nuevo														
Anexo	<p>Figura 82 Formulario de creación de usuarios</p> <p><b>Completa los datos del usuario a crear</b></p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Nombre * P4</p> <p>Apellidos * prueba 4</p> <p>Telefono * +57 3333333333 </p> <p>Rol * Agente </p> <p>Email * brayan.sierra@usantoto.edu.co</p> <p>Contraseña ..... </p> </div> <p>Fuente: Autor</p> <p>Figura 83 Lista de usuarios con el nuevo usuario</p> <table border="1"> <thead> <tr> <th>Gerencia Crediautos</th> <th>gerencia@crediautos.co</th> <th>3138073045</th> <th>admin</th> <th>Detalle</th> <th>Editar</th> <th>Eliminar</th> </tr> </thead> <tbody> <tr> <td>P4 prueba 4</td> <td>brayan.sierra@usantoto.edu.co</td> <td>3333333333</td> <td>Agente</td> <td>Detalle</td> <td>Editar</td> <td>Eliminar</td> </tr> </tbody> </table> <p>Fuente: Autor</p>	Gerencia Crediautos	gerencia@crediautos.co	3138073045	admin	Detalle	Editar	Eliminar	P4 prueba 4	brayan.sierra@usantoto.edu.co	3333333333	Agente	Detalle	Editar	Eliminar
Gerencia Crediautos	gerencia@crediautos.co	3138073045	admin	Detalle	Editar	Eliminar									
P4 prueba 4	brayan.sierra@usantoto.edu.co	3333333333	Agente	Detalle	Editar	Eliminar									

Fuente: Autor

Tabla 22 Prueba P5

Prueba	
Id de Prueba	P5
Descripción	Filtro de búsqueda de la página inicial en el módulo de usuario final

Resultado	Se realizaron varias pruebas con los filtros de la página inicial, teniendo en cuenta diferentes combinaciones que se puedan llegar a dar. En todos los casos la aplicación hace bien el filtro de las publicaciones
Anexo	<p style="text-align: center;">Figura 84 Filtro con marca Mazda y ajuste de precio</p>  <p style="text-align: center;">Fuente: Autor</p> <p style="text-align: center;">Figura 85 Resultado del filtro de la figura 81</p>  <p style="text-align: center;">Fuente: Autor</p>

Fuente: Autor

Tabla 23 Prueba P6

Prueba	
Id de Prueba	P6
Descripción	Filtros laterales en el módulo de usuario final
Resultado	Se realizaron diferentes combinaciones de estos filtros el sistema responde bien al aplicar eficazmente los filtros por ejemplo buscar por marca y precio

Anexo

### Figura 86 Filtro lateral Chevrolet y precio

Visualizando 1-15 de 20 resultados Ver 12 ▾

**Carros y Camionetas**

**Marcas**

- Changan
- Changhe
- Chery
- Chevrolet
- Chrysler
- Citroën
- DFM/DFSK
- Ds


**Precio:**

-


\$0 Millones — \$200 Millones

**Filtrar**


**Color**




2001 | 500000 Km  
CHEVROLET NHR 2001  
\$52,000,000



2014 | 0 Km  
Chevrolet Sonic Modelo 2014  
\$38,000,000



2007 | 190000 Km  
GRAN VITARA SPORT 2007 4X4  
\$39,000,000



2012 | 0 Km  
Captiva Sport 2012  
\$37,000,000

Fuente: Autor

Fuente: Autor

## 7. CONCLUSIONES Y ANÁLISIS DE RESULTADOS

Según los objetivos propuestos en este desarrollo se puede afirmar que:

- Para el levantamiento de requisitos fue un poco ambiguo el cliente de Crediautos ya que no tenía una idea clara que cómo quería el software. A parte, al no tener una idea clara durante el desarrollo agregaba más requisitos haciendo que el software fuera más complicado y más complejo, para superar esta dificultad se limitó el alcance del software con el cliente en varias reuniones.
- Durante la fase de diseño del software se realizaron varios prototipos esperando la aprobación de Crediautos, al igual que con los requisitos el diseño sufrió varias modificaciones al transcurso de desarrollo del software, ya que, Crediautos cambia de opinión con respecto a temas visuales.
- Durante el desarrollo se consideró la implementación de una pasarela de pago. El inconveniente era que Crediautos debía pagar la tarifa de suscripción, por eso cambiaron de parecer. Sin embargo, se desarrollaron los servicios y los componentes asociados con la pasarela de pago.

## 8. ANEXOS

### 8.1. Manual de usuarios

Este manual de usuario es una guía para los usuarios que interactúen con los tres diferentes módulos que se desarrollaron en este proyecto.

#### 8.1.1. Aplicativo móvil

En el aplicativo móvil se tiene las mismas funcionalidades que en el módulo usuario final con la diferencia en el diseño, que está de una forma más agradable para la vista de un smartphone que tenga sistema operativo Android.

##### 8.1.1.1 Registro de cliente

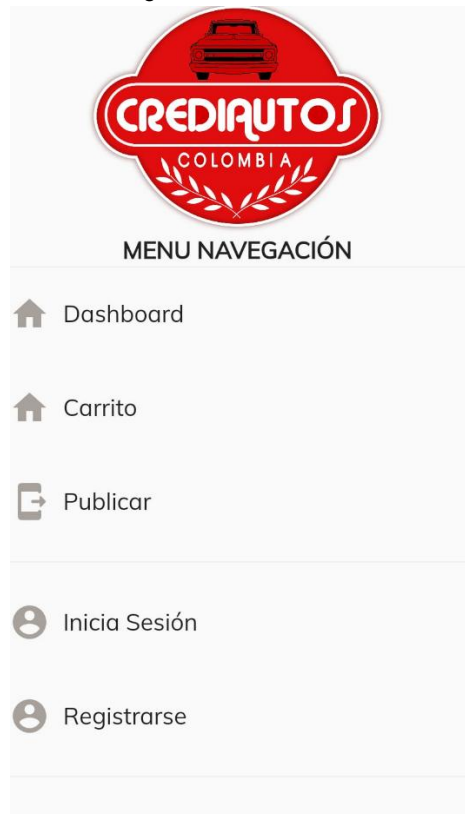
Para que un cliente se registre debe ingresar a la aplicación donde mostrara el dashboard de la APP, seguido debe dar click en el menú que se observa en la Figura 87, para poder desplegarlo como se muestra en la Figura 88.

Figura 87 Header aplicativo movil



Fuente: Autor

Figura 88 Menú lateral





Fuente: Autor


Al dar click en el submenú de Registrarse que sale en la Figura 88 lo redireccionará a la pantalla de registro de usuario como se puede evidenciar en la Figura 89 donde tendrá que diligenciar el pequeño formulario y luego dar click en el botón de ingresar


Figura 89 Registro de usuario


Registro

 Nombre

 Apellidos

 Telefono

 Correo electrónico

 Contraseña

Fuente: Autor

### ***8.1.1.2 Inicio de sesión del cliente***

Para iniciar sesión el usuario tiene dos opciones las cuales son, por medio de nuestra aplicación con un correo y una contraseña y la otra es por medio de autenticación de Google como se enseña en la Figura 90.

Figura 90 Inicio de sesión



The image shows a mobile application login screen. At the top, there is a red header with a white logo that says 'Credimotos COLOMBIA'. Below the header, the word 'Ingreso' is centered in a large, dark font. There are two input fields: the first is labeled 'Correo electrónico' with an '@' icon and a horizontal line below it; the second is labeled 'Contraseña' with three asterisks and a horizontal line below it. Below these fields is a grey button with the text 'Ingresar'. Underneath that is a white button with the Google logo and the text 'Inicia sesión con Google'. At the bottom, there is a white button with the text 'Registrarse' in red.

Fuente: Autor

### 8.1.1.3 *Publicación de un producto*

En la publicación de un vehículo o artículo el usuario debe estar previamente logueado en el aplicativo móvil. Para publicar el usuario debe desplegar el menú lateral como se realizó

en la Figura 90 del paso anterior, al tener ya el menú el usuario debe dar click en publicar como se visualiza en la Figura 88, el cual lo redirecciona a la pantalla de publicar que contiene 4 pasos para hacer una nueva publicación.

El paso 1 tiene un formulario general de la aplicación como se visualiza en la Figura 91.

Figura 91 Paso 1 de publicar

## Publicar

**Paso 1 de 4**

Titulo para tu publicacion  
carro full

Descripción  
hermoso vehículo

Selecciona la Categoría  
Carros y Camionetas

Seleccione la Marca  
Fiat

Seleccione el Modelo  
Argo

Seleccione el Version  
Otro

¿De que año es tu Fiat?  
2012

4/4

**Siguiente**

Fuente: Autor

El paso 2 contiene un formulario más detallado sobre el vehículo a publicar ver Figura 92.

Figura 92 Paso 2

## Publicar

### características de tu Vehículo

Tipo de combustible  
**gasolina**

Tipo de carrocería  
**sedan**

Dirección  
**mecánica**

Trasmisión  
**manual**

Control de tracción  
**delantera**

Número de puertas  
**5**

Cilindrada  
**1.6**

Motor  
**v8**

Placa  
**xxx000**

Unico dueño

**Siguiente**

Fuente: Autor

En el paso 3 se tiene unas opciones que no son obligatorias esta son características más detallada del vehículo presenta en la Figura 93.

Figura 93 Paso 3

## Publicar

**Selecciona las características de tu Vehículo**

Mas Informacion	^
Asegurado	<input checked="" type="checkbox"/>
Tecnomecánica	<input type="checkbox"/>

Equipamiento

Seguridad

Confort

Sonido

Mas Equipamiento

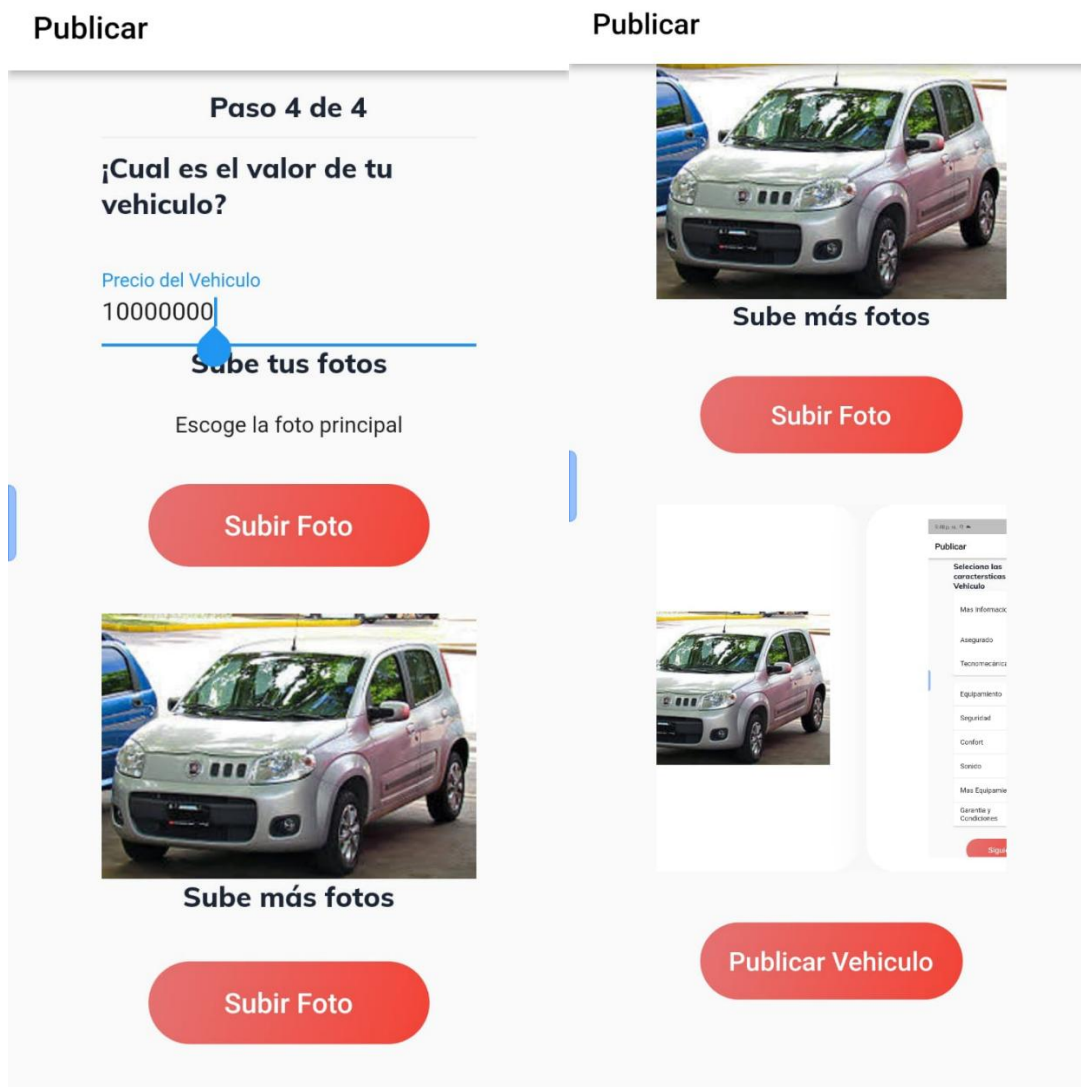
Garantía y Condiciones

**Siguiente**

Fuente: Autor

En el último paso se puede diligenciar el precio del vehículo, la foto de porta y las fotos que se mostraran en la galería de la publicación como se visualiza a continuación en la Figura 94.

Figura 94 Paso 4



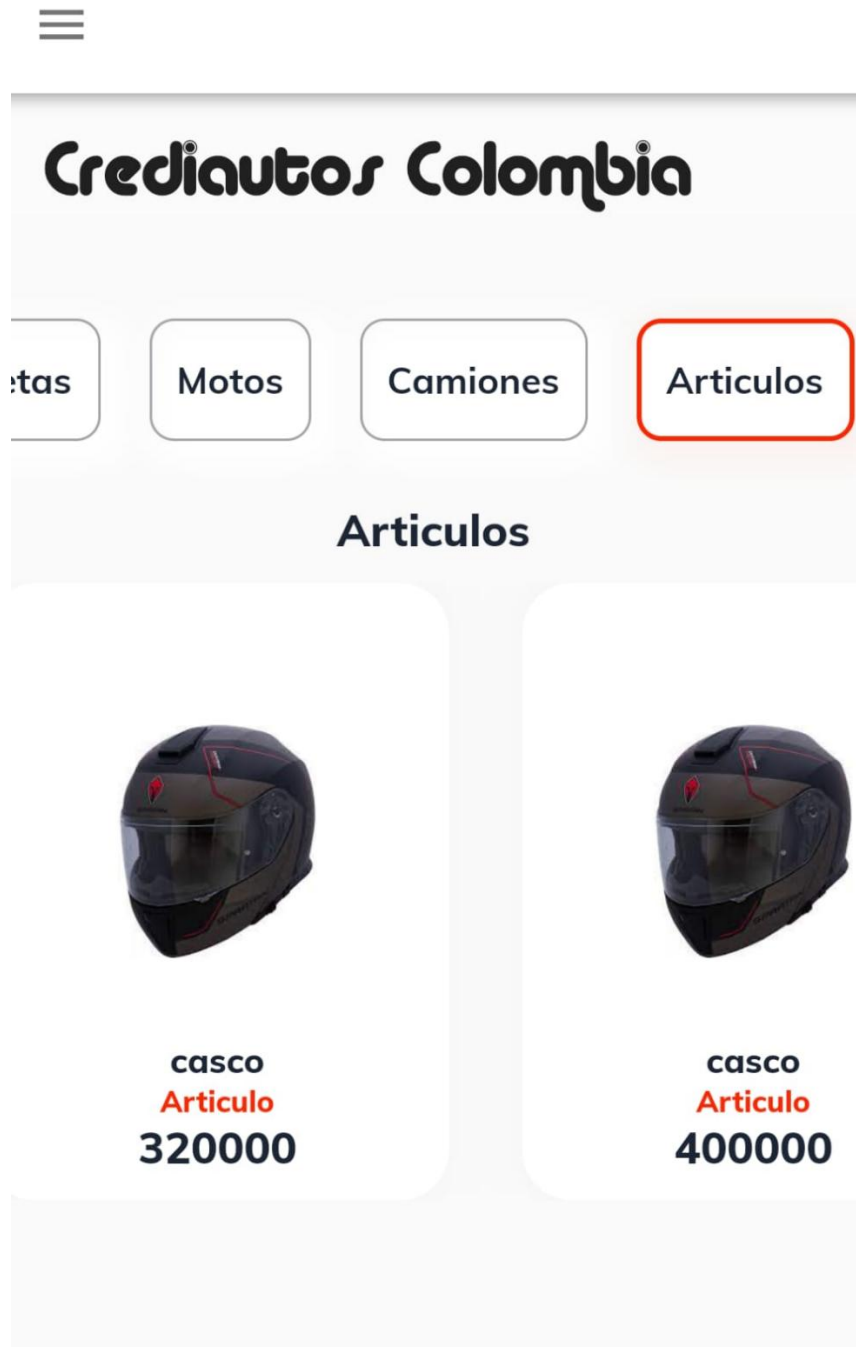
Fuente: Autor

Al dar click en el botón "publicar vehículo" termina el proceso de publicación de producto.

#### 8.1.1.4 Compra de un artículo automovilístico

Para la compra de un artículo lo primero que tiene que hacer el usuario es estar logueado y seleccionar un producto como se ve en la Figura 95.

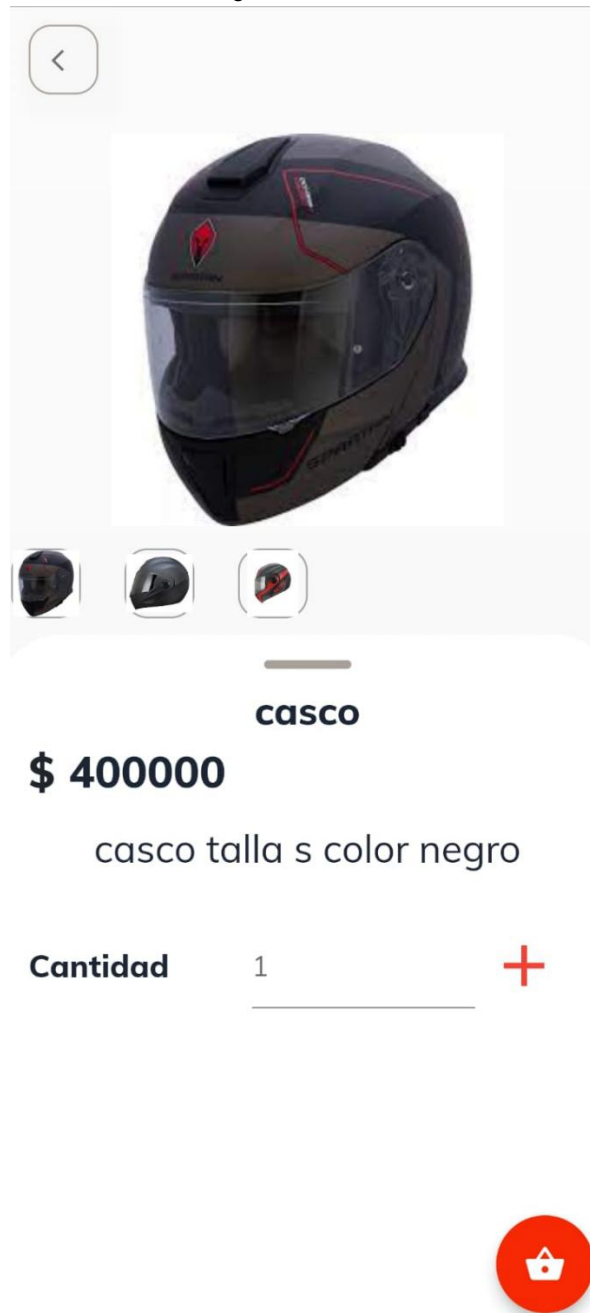
Figura 95 Lista de artículos



Fuente: Autor

Después de tener el artículo seleccionado se agrega al carrito de compras dando click en el botón rojo que tiene un icono de canasta y está situado en la parte inferior derecha de la pantalla como se puede observar en la Figura 96.

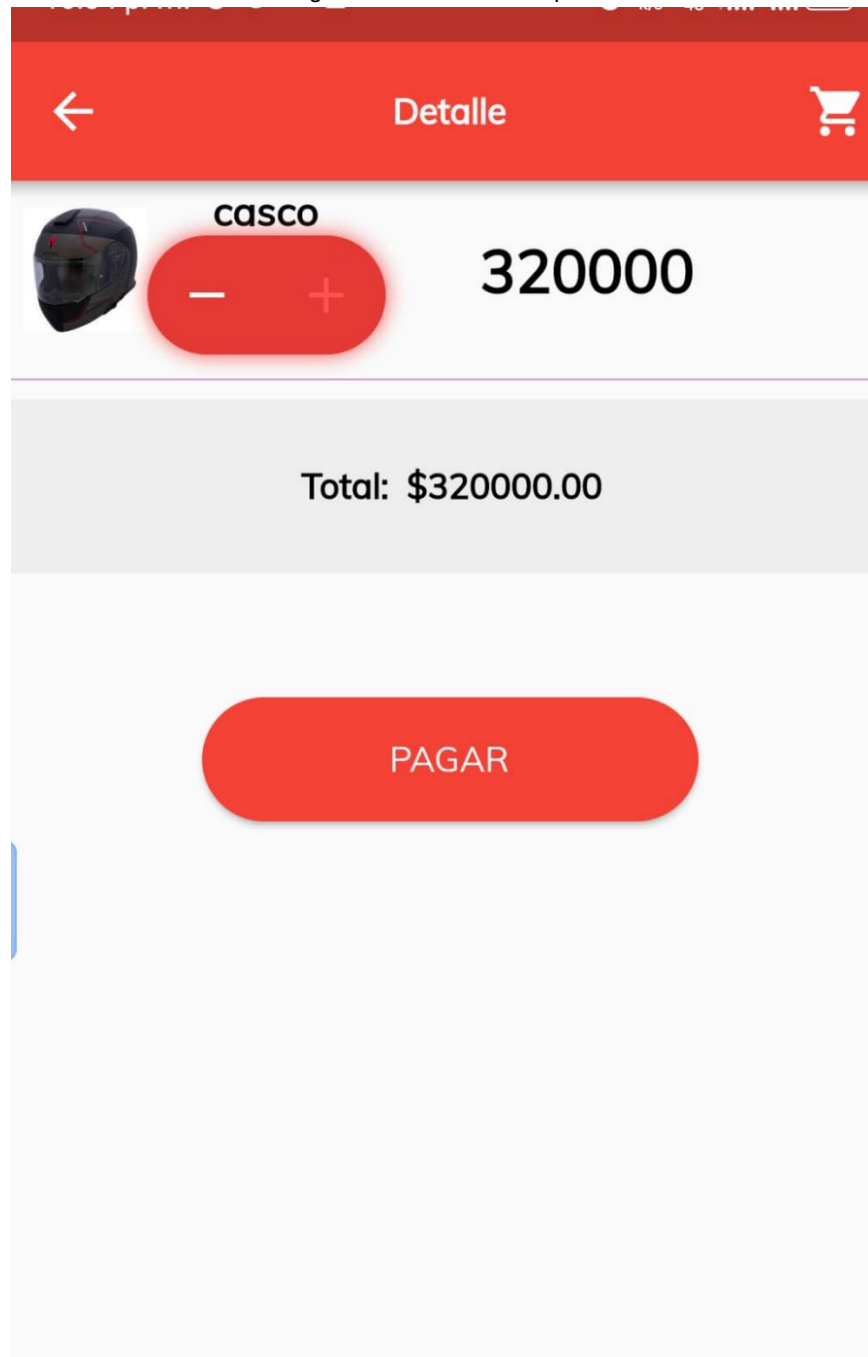
Figura 96 Artículo



Fuente: Autor

Al dar click en el botón de la canasta lo redirecciona al carrito de compras donde se puede ver la cantidad de artículos que desea comprar, como también por unidad del producto como se enseña en la Figura 97.

Figura 97 Carrito de compras



Fuente: Autor

Si se le da click en el botón de “pagar” de la Figura 97 nos direcciona a la pasarela de pago con sus diferentes opciones de pago como se puede evidenciar en la Figura 98.

Figura 98 Métodos de pago

The image displays two sequential screens of a payment gateway. Both screens feature a top navigation bar with a back arrow and three payment method buttons: 'Tarjeta Credito' (highlighted in red), 'PSE', and 'Daviplata'. The left screen is titled 'Tarjeta de Credito' and contains a section for 'Información Básica' with input fields for 'Nombre', 'Apellidos', 'Tipo de Documento' (with a dropdown arrow), 'Número de Documento', 'Número de Telefono', and 'Email'. The right screen is also titled 'Tarjeta de Credito' and contains a section for 'Informacion de Tarjeta' with input fields for 'Número de la tarjeta', 'CVV', 'Año de expedición', 'Mes de expedición', and 'Número de cuotas'. Below these fields, the total amount 'Total: \$320000.00' is displayed in bold. At the bottom of the right screen is a large red button labeled 'Pagar'.

Fuente: Autor

## 9. REFERENCIAS

Digite. (2021). [Imagen]. Obtenido de digite: <https://www.digite.com/es/agile/que-es-scrum/>

Gallego, M. T. (s.f.). Metodología SCRUM. En M. T. Gallego, *Metodología SCRUM*.

Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. (s.f.).

En P. L. Sonia I. Mariño, *Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación*.

Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. USA: McGraw-Hill.