



**Manual Técnico**

## Road Luxury



**Aprendices SENA**

**Santiago Urrea**

**Valery Bejarano**

**Carlos Calderon**

**CENTRO INDUSTRIAL Y DESARROLLO EMPRESARIAL – CIDE**

**ANÁLISIS Y DESARROLLO DE SOFTWARE - ADSO**



## Manual Técnico

### SOACHA, CUNDINAMARCA

2025

- 1. Introducción**
- 2. Objetivos del Proyecto**
  - 2.1 Objetivo general**
  - 2.2 Objetivos específicos**
- 3. Alcance del proyecto**
- 4. Personal Involucrado**
  - 4.1 Diagramas de Casos de Uso**
- 5. Características de los usuarios**
- 6. Requerimientos**
  - 6.1 Requerimientos funcionales**
  - 6.2 Requerimientos no funcionales**
- 7. Arquitectura del proyecto**
  - 7.1 Definición metodología**
  - 7.2 Ciclo de vida**
  - 7.2 Diagrama de despliegue**
- 8. Fundamentos y herramientas utilizadas**
- 9. Requisitos del Sistema**
- 10. Procesos del Software**
- 11. Instalación de Aplicaciones**
- 12. Diagrama de Clases**
- 13. Modelo entidad Relación**
- 14. Diagrama entidad relación**



## Manual Técnico

15. Diccionario de Datos

16. Glosario

17. Referencias

### INTRODUCCION

El presente documento tiene como propósito identificar y describir las diferentes técnicas, herramientas y metodologías empleadas durante el proceso de desarrollo del sistema de información para la gestión de alquiler de autos. Su finalidad es facilitar la comprensión de los procesos internos que permiten la administración de usuarios, vehículos, servicios adicionales, reservas y facturación, los cuales constituyen los módulos principales del software.

A lo largo de este manual se presentará una explicación detallada del funcionamiento y diseño del aplicativo, de forma que el personal encargado del mantenimiento, mesa de ayuda o soporte técnico cuente con la información necesaria para brindar acompañamiento oportuno, resolver dudas y atender inquietudes del usuario final.

Cabe resaltar que este manual técnico no pretende ser un curso de formación en desarrollo de software, sino un documento de referencia que expone las técnicas, criterios y decisiones implementadas durante la construcción del sistema.

Por último, se recomienda que quienes consulten este documento posean conocimientos básicos de informática y del uso de componentes esenciales del computador, como el teclado, mouse, monitor y unidad de procesamiento, con el fin de comprender adecuadamente los contenidos aquí descritos.



### Objetivos del Proyecto

Desarrollar un sistema de información para la gestión integral del alquiler de autos que permita administrar de manera eficiente los vehículos, usuarios, reservas, servicios adicionales y procesos de facturación, garantizando un funcionamiento óptimo, confiable y accesible para el personal operativo y administrativo.

### Objetivos específicos

1. **Implementar un módulo de gestión de usuarios** que permita registrar, actualizar y controlar la información de clientes, empleados y administradores del sistema.
2. **Diseñar un módulo de administración de vehículos** que facilite el registro, disponibilidad, características y estado de cada automóvil dentro de la flota.
3. **Desarrollar un módulo de reservas** que permita crear, modificar, consultar y cancelar alquileres de forma rápida y organizada.
4. **Integrar un módulo de servicios adicionales** (como seguros, accesorios, kilometraje extra, entre otros) para complementar y personalizar cada alquiler.
5. **Crear un sistema de facturación** que genere comprobantes precisos y automatizados según los servicios utilizados por cada cliente.
6. **Asegurar una interfaz intuitiva y fácil de usar**, que permita a los operadores navegar por el sistema sin dificultades.
7. **Garantizar la seguridad y consistencia de la información**, mediante controles de acceso, validaciones y manejo adecuado de datos.
8. **Optimizar los procesos internos de la empresa**, reduciendo tiempos de atención y minimizando errores operativos mediante la automatización de tareas.

### Alcance del proyecto

El software **Road Luxury** está diseñado para servir como una herramienta tecnológica de apoyo informativo y formativo para la entidad SENA y para los usuarios directos del sistema, específicamente aprendices e instructores del área de Hotelería y Turismo. Su objetivo es



## Manual Técnico

facilitar la comprensión y práctica de los procesos operativos relacionados con la gestión y administración del alquiler de autos.

### Personal involucrado

<b>Nombre</b>	Santiago Estiven Urrea Torres
<b>Rol</b>	Analista y Desarrollador de Software
<b>Categoría Profesional</b>	Tecnólogo en Análisis y Desarrollo de Software
<b>Responsabilidades</b>	<ul style="list-style-type: none"><li>• Análisis de requerimientos</li><li>• Desarrollo del backend</li><li>• Gestión de bases de datos</li><li>• Documentación técnica</li><li>• Integración de módulos</li></ul>
<b>Información de contacto</b>	<b>Num:</b> 323 2832459 <b>Correo:</b> <b>santiagosteivenurreatorres@gmail.com</b>
<b>Aprobación</b>	<b>Si</b>

<b>Nombre</b>	<b>Valery Daniela Bejarano Rojas</b>
<b>Rol</b>	Analista y Desarrollador de Software
<b>Categoría Profesional</b>	Tecnólogo en Análisis y Desarrollo de Software
<b>Responsabilidades</b>	<ul style="list-style-type: none"><li>• Análisis de requerimientos</li><li>• Desarrollo del backend</li><li>• Gestión de bases de datos</li></ul>



## Manual Técnico

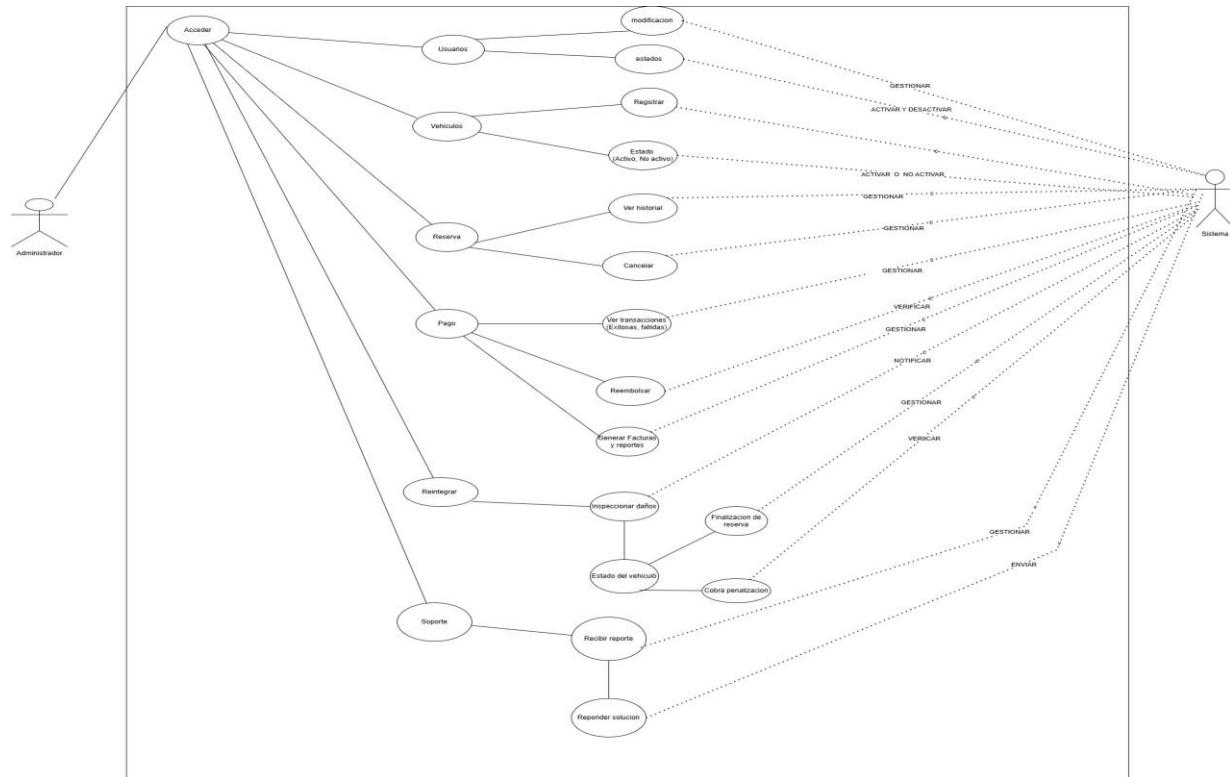
	<ul style="list-style-type: none"><li>• Documentación técnica</li><li>• Integración de módulos</li></ul>
Informacion de contacto	<b>Num: 324 4306979</b> <b>Correo:</b> <b>Valerybejarano429@gmail.com</b>
Aprobacion	Si

Nombre	<b>Carlos Angel Calderon Cubillos</b>
Rol	Analista y Desarrollador de Software
Categoría Profesional	Tecnologo en Analisis y Desarrollo de Software
Responsabilidades	<ul style="list-style-type: none"><li>• Análisis de requerimientos</li><li>• Desarrollo del backend</li><li>• Gestión de bases de datos</li><li>• Documentación técnica</li><li>• Integración de módulos</li></ul>
Informe de contacto	<b>Num:3168911937</b> <b>Correo:</b> <b>Motocalderon2196@gmail.com</b>
Aprobacion	Si



## Diagramas de casos de uso

## **CU01 . Actividades Del Administrador**





## 1. IDENTIFICACIÓN DEL CASO DE USO

1.1 Id Caso	CU01
1.2 Nombre	Actividades del Administrador en Road Luxury

## 2. HISTÓRICO DE CASO DE USO

2.1 Autor	Carlos Ángel Calderón Cubillos/ Equipo Road Luxury
2.2 Fecha de Creación	01/12/2025
2.3 Última Actualización	1.0
2.4 Actualizado por	Santiago Estiven Urrea Torres
2.5 Versión	1.0

## 3. DEFINICIÓN DE UN CASO DE USO

### 3.1 DESCRIPCIÓN

Permite evidenciar todas las actividades que el administrador puede realizar dentro del sistema Road Luxury cuando inicia sesión, incluyendo la gestión de usuarios, vehículos, reservas, pagos, reintegros, inspecciones y soporte.

El administrador tiene acceso para realizar todas las tareas de control necesarias para el funcionamiento adecuado del sistema.

### 3.2 ACTORES

- Administrador
- Sistema Road Luxury

### 3.3 PRECONDICIONES

1. El administrador debe estar registrado correctamente en el sistema.
2. Debe haber iniciado sesión para gestionar usuarios.



## Manual Técnico

3. Debe haber iniciado sesión para gestionar vehículos (crear, editar, activar, inactivar).
4. Debe haber iniciado sesión para consultar, actualizar o cancelar reservas.
5. Debe haber iniciado sesión para revisar transacciones, procesar pagos y generar facturas.
6. Debe haber iniciado sesión para realizar inspecciones de reintegro o aplicar penalizaciones.
7. Debe haber iniciado sesión para recibir y responder reportes del módulo de soporte.

### 3.4 FLUJO NORMAL

Paso	Actor	Sistema
1	El administrador ingresa sus credenciales para iniciar sesión.	Valida las credenciales y permite el acceso si son correctas.
2	El administrador gestiona usuarios (crear, actualizar, activar o desactivar).	Verifica la información y permite registrar o modificar usuarios.
3	El administrador gestiona vehículos (crear, consultar, actualizar o cambiar estado).	Ratifica disponibilidad del vehículo y actualiza su estado en la BD.
4	El administrador consulta o cancela reservas.	Confirma el estado de la reserva y permite actualizarla o cancelarla.
5	El administrador revisa pagos y transacciones.	Verifica transacciones exitosas o fallidas y registra la operación.
6	El administrador genera facturas y reportes.	Produce el documento y lo guarda en el sistema.
7	El administrador realiza reintegros.	Valida condiciones de la reserva y procesa el reembolso.
8	El administrador inspecciona daños al finalizar la reserva.	Registra daños, calcula penalizaciones y actualiza el estado del vehículo.
9	El administrador recibe y responde reportes del módulo de soporte.	Envía respuesta al usuario y cierra el reporte.

### 3.5 FLUJO ALTERNATIVO

Paso	Actor	Sistema
1	El administrador ingresa credenciales incorrectas.	Muestra mensaje de error y bloquea el ingreso.
2	El administrador intenta gestionar un usuario inexistente.	Muestra mensaje indicando que el usuario no existe.
3	El administrador intenta cambiar el estado a un vehículo inexistente.	Indica que el vehículo no está registrado.



## Manual Técnico

<b>4</b>	El administrador intenta cancelar una reserva ya cancelada.	Informa que la reserva ya se encuentra cerrada.
<b>5</b>	El administrador intenta generar factura sin datos completos.	Advierte que no se puede generar el documento.
<b>6</b>	El administrador intenta registrar un pago ya procesado.	Notifica transacción duplicada.

### 3.6 FLUJO EXCEPCIONAL

Paso	Actor	Sistema
<b>1</b>	El administrador falla más de 3 veces el inicio de sesión.	El sistema bloquea temporalmente la cuenta.
<b>2</b>	El administrador elimina información crítica accidentalmente.	Genera respaldo automático y muestra alerta.
<b>3</b>	El administrador intenta finalizar una reserva sin inspección del vehículo.	Impide la acción y solicita revisión previa.

### 3.7 POSCONDICIONES

- Todas las acciones quedan registradas en la base de datos.
- Los cambios realizados sobre usuarios, vehículos, reservas, pagos y reportes quedan actualizados.
- Se generan historiales y trazabilidad de cada operación.
- El sistema mantiene integridad y disponibilidad de la información.

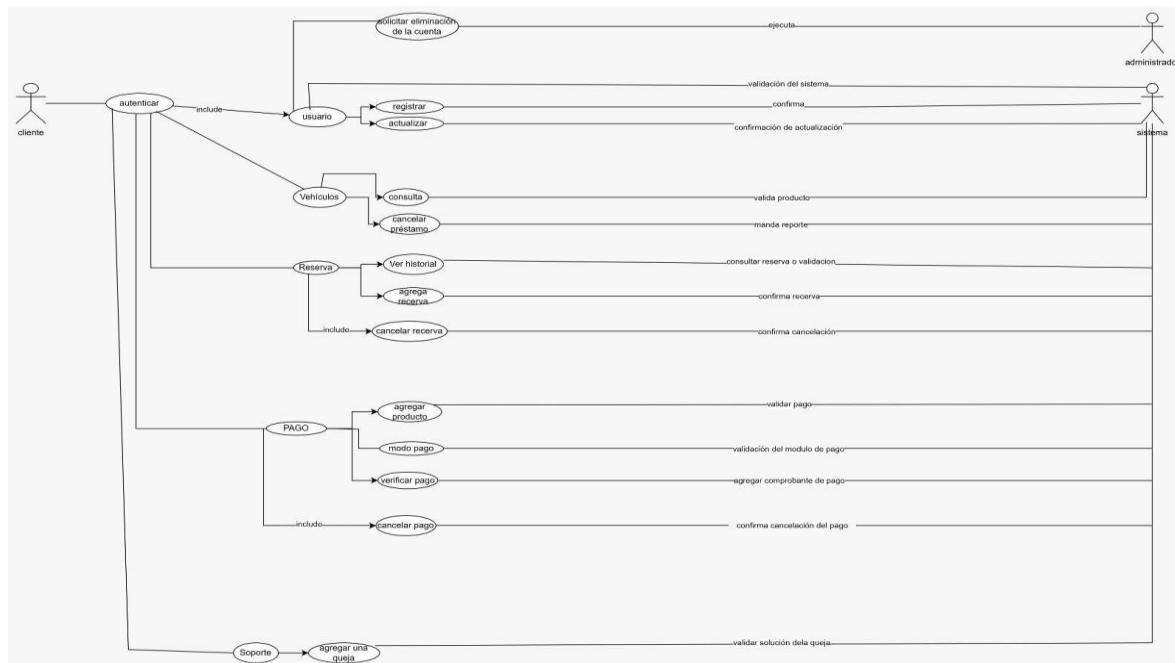
### 3.8 FRECUENCIA

El caso de uso tiene **frecuencia alta**, ya que el administrador realiza continuamente tareas de control y gestión dentro del sistema.



## Manual Técnico

### CU02 – Actividades del Cliente





## 1. IDENTIFICACIÓN DEL CASO DE USO

<b>1.1 Id Caso</b>	CU02
<b>1.2 Nombre</b>	Actividades del Cliente en Road Luxury

## 2. HISTÓRICO DE CASO DE USO

<b>2.1 Autor</b>	Santiago Estiven Urrea Torres / Equipo Road Luxury
<b>2.2 Fecha de Creación</b>	01/12/2025
<b>2.3 Última Actualización</b>	1.0
<b>2.4 Actualizado por</b>	Santiago Estiven Urrea Torres
<b>2.5 Versión</b>	1.0

## 3. DEFINICIÓN DE UN CASO DE USO

### 3.1 DESCRIPCIÓN

Permite evidenciar todas las actividades que el cliente puede realizar dentro del sistema Road Luxury cuando inicia sesión, incluyendo la gestión de su cuenta, la consulta de vehículos, la reserva, la realización de pagos, la cancelación de transacciones, el envío de quejas y el seguimiento de su historial.

El cliente tiene acceso a todas las funciones permitidas para su rol dentro de la plataforma.

### 3.2 ACTORES

- Cliente
- Sistema Road Luxury
- Administrador (*solo cuando interviene en eliminación de cuenta*)



### 3.3 PRECONDICIONES

1. El cliente debe estar registrado en el sistema.
2. Debe haber iniciado sesión para gestionar su información personal.
3. Debe haber iniciado sesión para consultar vehículos disponibles.
4. Debe haber iniciado sesión para crear una reserva.
5. Debe haber iniciado sesión para realizar pagos o verificar transacciones.
6. Debe haber iniciado sesión para cancelar reservas o pagos.
7. Debe haber iniciado sesión para enviar quejas dentro del módulo de soporte.

### 3.4 FLUJO NORMAL

Paso	Actor	Sistema
1	El cliente ingresa sus credenciales para iniciar sesión.	Valida credenciales y permite el acceso al sistema.
2	El cliente registra o actualiza su información personal.	Confirma registro o actualización de datos.
3	El cliente consulta vehículos disponibles.	Valida disponibilidad y muestra información.
4	El cliente agrega una reserva.	Verifica disponibilidad y confirma la reserva.
5	El cliente consulta su historial de reservas.	Muestra historial completo y detalles.
6	El cliente realiza un pago (selección del producto y método).	Valida pago, registra comprobante y confirma transacción.
7	El cliente revisa el estado de su pago.	Muestra validación del módulo de pago.
8	El cliente envía una queja al soporte.	Registra la queja y confirma su envío.
9	El cliente solicita eliminación de su cuenta.	El administrador ejecuta el proceso y el sistema confirma la eliminación.



### 3.5 FLUJO ALTERNATIVO

Paso	Actor	Sistema
1	El cliente ingresa credenciales incorrectas.	Muestra error y no permite el acceso.
2	El cliente intenta registrar datos ya existentes.	Notifica datos duplicados.
3	El cliente intenta reservar un vehículo no disponible.	Informa indisponibilidad del vehículo.
4	El cliente intenta cancelar una reserva ya cancelada.	Notifica que la reserva ya fue cerrada anteriormente.
5	El cliente intenta pagar con un método inválido.	Rechaza pago y solicita corregir método.
6	El cliente intenta cancelar un pago ya procesado.	Indica que ya no es posible cancelar.

### 3.6 FLUJO EXCEPCIONAL

Paso	Actor	Sistema
1	El cliente excede los intentos de acceso.	Bloquea cuenta temporalmente.
2	El cliente intenta actualizar información crítica sin permisos.	Rechaza la acción y protege integridad de datos.
3	El cliente intenta cancelar un pago durante un fallo del proveedor.	Notifica error y solicita intentar nuevamente.

### 3.7 POSCONDICIONES

- Los datos del cliente quedan actualizados correctamente.
- Las reservas, pagos y operaciones realizadas quedan registradas.
- Se actualiza el historial del cliente según sus acciones.
- Las quejas enviadas quedan almacenadas en el módulo de soporte.
- La integridad de la base de datos se mantiene.

### 3.8 FRECUENCIA



## Manual Técnico

Este caso de uso tiene **frecuencia alta**, ya que es el flujo principal y más utilizado en Road Luxury por los clientes.

### Características de los Usuarios

A continuación se darán a conocer los usuarios que manejaran este sistema y también se darán a conocer las funciones que cumplen cada uno

**Tabla 1 .Características del usuario cliente**

Tipo de usuario	Cliente
Formación	Persona natural los conocimientos necesarios para utilizar un aplicativo web tanto en móvil como en computador
Habilidades	Conocimiento básico en el uso de aplicaciones web
Actividades	-Consulta de reservas, autos y generación de contratos - Actualización y eliminación de reservas -Generacion de contratos en la renta de autos

**Tabla 2 .Característica del usuario administrador**



<b>Tipo de usuario</b>	Administrador
<b>Formación</b>	Formación técnica para desarrollar las funciones respectivas de administración del sistema
<b>Habilidades</b>	-Conocimiento amplio frente al manejo de aplicaciones web, móviles y de escritorio -Gran dominio en la resolución rápida de problemas -Trabajo en equipo
<b>Actividades</b>	-Consulta de usuarios, vehículos y facturas -Registro, actualización y eliminación de usuarios y vehículos. -Generación de contrato para el alquiler del vehículo

## 6.5 Requerimientos

### 6.5.1 Requerimientos Funcionales

El sistema Road Luxury debe permitir el registro, autenticación y gestión de usuarios, diferenciando claramente entre clientes y administradores, garantizando acceso controlado mediante credenciales y seguridad basada en tokens. El sistema debe permitir que los usuarios creen una cuenta, inicien sesión, recuperen su contraseña y actualicen sus datos personales, mientras que el administrador debe tener la capacidad de gestionar, modificar o desactivar perfiles según sea necesario.

El sistema debe ofrecer un catálogo completo de vehículos, en el cual el administrador pueda registrar nuevos automóviles ingresando su marca, modelo, categoría, precio por día, especificaciones técnicas, imágenes y estado. Asimismo, debe permitir modificar información existente y eliminar o desactivar vehículos cuando sea necesario. Los clientes deben visualizar este catálogo de manera organizada, con posibilidad de consultar detalles, fotos, precios, disponibilidad, y filtrar vehículos según características como categoría, precio o marca.



## Manual Técnico

El sistema debe permitir la creación y gestión de reservas, ofreciendo a los clientes la opción de seleccionar fechas específicas para alquilar un vehículo y validando automáticamente la disponibilidad del mismo dentro de ese rango. Cada reserva debe generar un cálculo automático del costo total basado en los días de alquiler, y debe quedar registrada en el sistema. El administrador debe poder visualizar todas las reservas generadas, revisar su estado, eliminarlas o cancelarlas cuando sea pertinente, mientras que el cliente debe poder consultar su historial de reservas en cualquier momento.

El sistema debe registrar y gestionar pagos asociados a cada reserva, permitiendo almacenar el método de pago, el valor total y la fecha de la transacción. Los clientes deben visualizar en su panel el comprobante del pago correspondiente, y el administrador debe contar con acceso a la lista completa de pagos realizados dentro de la plataforma. El sistema debe generar comprobantes digitales vinculados a cada operación.

El sistema debe incluir un módulo de soporte donde los clientes puedan enviar reclamos, solicitudes o sugerencias, las cuales deben quedar registradas con fecha, descripción y estado. El administrador debe poder visualizar estas solicitudes, responderlas desde su panel y marcar su estado como resuelta, permitiendo que el cliente consulte posteriormente la respuesta y el estado actualizado de su petición.

El sistema debe garantizar la seguridad del acceso, protegiendo las contraseñas mediante encriptación, gestionando sesiones mediante tokens seguros, restringiendo el acceso a funcionalidades internas únicamente a usuarios autenticados y asegurando que ciertas operaciones solo sean permitidas para administradores.

El sistema debe almacenar imágenes y datos relacionados con los vehículos de forma eficiente, permitiendo su visualización rápida dentro del catálogo. Debe registrar un historial detallado de reservas, pagos, modificaciones realizadas por el administrador, solicitudes de soporte y toda actividad relevante para auditoría y seguimiento dentro del sistema.



### 2.5.1 Requerimientos no funcionales

#### Navegabilidad:

El sistema deberá poseer una interfaz gráfica moderna, clara y fácil de usar, permitiendo que tanto clientes como administradores puedan desplazarse por cada módulo (vehículos, reservas, pagos, soporte, perfil) de manera intuitiva sin necesidad de conocimientos técnicos adicionales.

#### Fiabilidad:

El sistema debe garantizar un funcionamiento estable con un margen de operación mínimo del 95% del tiempo, asegurando que los servicios críticos como registro de usuarios, catálogo de vehículos y creación de reservas se mantengan disponibles. Cualquier ventana de mantenimiento deberá ser informada oficialmente con anticipación.

#### Seguridad:

El sistema deberá asegurar la protección de los datos mediante cifrado de contraseñas, validación de accesos y uso de tokens para autenticación. Se deberá verificar cada acción para garantizar que solo los usuarios autorizados (cliente o administrador) puedan ejecutar las funciones correspondientes a su rol.

#### Control de Usuarios:

El administrador deberá poder gestionar todos los usuarios del sistema sin restricciones, pudiendo editar, actualizar o eliminar cuentas según sea necesario, sin importar el rol. El sistema debe asegurar que los clientes solo puedan acceder a sus propios datos, reservas y pagos.

## 7. Arquitectura del proyecto

La arquitectura del sistema Road Luxury está basada en un modelo cliente–servidor con separación clara entre la capa de presentación (front-end), la capa lógica (back-end) y la capa de almacenamiento (base de datos). Esta estructura permite escalabilidad, mantenibilidad y un funcionamiento seguro para la gestión de vehículos, reservas, usuarios, pagos y solicitudes de soporte.

### Definición de la metodología



## Manual Técnico

El desarrollo del proyecto Road Luxury se realizó utilizando la Metodología Incremental, la cual consiste en construir el sistema por partes o módulos independientes, entregando y mejorando cada componente de manera progresiva. Esta metodología permite ir desarrollando las funcionalidades en diferentes etapas, iniciando por los módulos esenciales y agregando nuevas características conforme avanza el proceso.

Con este enfoque, el proyecto se dividió en incrementos como: gestión de usuarios, catálogo de vehículos, reservas, pagos, soporte y panel administrativo. Cada incremento fue analizado, diseñado, implementado y probado antes de continuar con el siguiente, garantizando un avance controlado y una integración segura entre los módulos.

La metodología incremental facilitó el trabajo del equipo, permitiendo incorporar mejoras, corregir errores rápidamente y adaptar funcionalidades según la retroalimentación obtenida durante el desarrollo. Gracias a este enfoque, Road Luxury pudo construirse de manera ordenada, flexible y con entregas parciales completamente operativas.

### Ciclo de vida

El ciclo de vida del proyecto Road Luxury está estructurado en fases secuenciales e iterativas que permiten construir el sistema por partes, garantizando calidad, control y mejora continua. Cada fase se repite en cada incremento del sistema, asegurando que cada módulo se desarrolle completamente antes de pasar al siguiente. El ciclo de vida utilizado comprende las siguientes etapas:

#### 1. Análisis de Requerimientos

En esta fase se recopilaron y documentaron las necesidades del cliente y los usuarios finales. Se identificaron los requerimientos funcionales y no funcionales del sistema, así como las reglas de negocio para la gestión de usuarios, vehículos, reservas, pagos y soporte. Esta fase permitió definir con claridad el alcance inicial del proyecto.

#### 2. Diseño del Sistema

Aquí se estructuró la arquitectura general del proyecto, los diagramas UML, la definición de las vistas, los modelos de la base de datos, el diseño de la interfaz y la organización de los



módulos. Se estableció cómo interactúan las capas del sistema y cómo se conectan entre sí los componentes lógicos y de presentación.

### 3. Desarrollo e Implementación

En esta fase se construyeron los módulos del sistema de manera incremental. Cada incremento incluyó programación del back-end, creación de las vistas del front-end, conexión con la base de datos, desarrollo de la autenticación, manejo de imágenes, lógica de reservas, pagos y panel administrativo. Cada módulo se implementó por separado y luego se integró al sistema general.

### 4. Pruebas y Verificación

Se realizaron pruebas de funcionamiento, validación, seguridad y usabilidad para garantizar que cada módulo operara correctamente. Se verificó la correcta interacción entre módulos, el funcionamiento de las reservas, la autenticación, la disponibilidad de vehículos, la carga de imágenes y la correcta respuesta del sistema. Los errores identificados fueron corregidos antes de continuar con el siguiente incremento.

### 5. Integración y Ajustes

Después de aprobar cada incremento, se integró al sistema completo. En esta fase se realizaron ajustes de diseño, organización del código, optimización de rendimiento, corrección de inconsistencias y mejoras visuales. Esta etapa permitió refinar el sistema para consolidar una experiencia coherente en todos los módulos.

### 6. Implementación Final

Una vez completados todos los incrementos y validado el funcionamiento global, el sistema Road Luxury fue preparado para su despliegue final. Esto incluyó configuración del servidor,



## Manual Técnico

migración de datos, pruebas finales en entorno real y verificación de la disponibilidad de todos los módulos: usuarios, vehículos, reservas, pagos y soporte.

### 7. Mantenimiento y Mejora Continua

Tras la entrega, el ciclo de vida contempla la corrección de errores residuales, ajustes solicitados por el cliente, actualizaciones de seguridad, optimización del rendimiento y posible incorporación de nuevas funcionalidades. La metodología incremental facilita estas mejoras sin alterar el funcionamiento general del sistema.

#### Diagrama de despliegue

#### Fundamentos y herramientas utilizadas

##### Fundamentos de desarrollo

###### **Ingeniería de Software:**

Se aplicaron conceptos de análisis, diseño, implementación y pruebas para construir un sistema estructurado, documentado y funcional, siguiendo prácticas recomendadas del ciclo de vida del software.

###### **Programación Modular:**

Cada funcionalidad se desarrolló como un módulo independiente (usuarios, vehículos, reservas, pagos, soporte), permitiendo integrar y mejorar el sistema de manera incremental.

###### **Arquitectura Cliente–Servidor:**

El sistema se dividió en una capa de presentación (interfaz web) y una capa lógica en el servidor que gestiona las operaciones y se comunica con la base de datos.

###### **Orientación a Objetos:**

El back-end implementó modelos y clases que representan entidades como Usuario, Vehículo, Reserva y Pago, favoreciendo el orden y la reutilización de código.



## Manual Técnico

### Metodología Incremental:

El desarrollo se realizó por etapas, construyendo un módulo completamente funcional en cada incremento y permitiendo ajustes continuos durante el avance del proyecto.

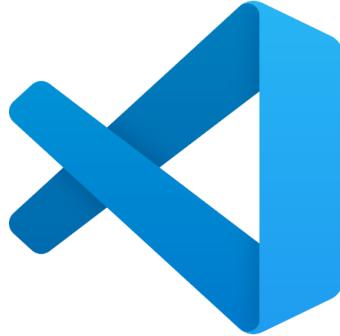
### Seguridad y Autenticación:

Se aplicaron mecanismos de autenticación segura utilizando tokens, encriptación de contraseñas, validación de acceso y control de roles (cliente y administrador).

### Herramientas utilizadas

#### Visual Studio Code

Editor de código utilizado para escribir, organizar y mantener todo el proyecto, tanto el backend en Python/Django como el frontend en HTML, CSS, JavaScript y React. Permite extensiones para depurar, formatear código y conectar con GitHub.



Ilus 1. Logo de Visual Studio extraido de

[https://es.wikipedia.org/wiki/Archivo:Visual\\_Studio\\_Code\\_1.35\\_icon.svg](https://es.wikipedia.org/wiki/Archivo:Visual_Studio_Code_1.35_icon.svg)

#### Python

Lenguaje de programación principal del lado del servidor. Permite desarrollar la lógica del sistema, los modelos, controladores, validaciones, seguridad, autenticación y la interacción con la base de datos.



Ilus.2 Logo de Python Extraido de

<https://es.wikipedia.org/wiki/Archivo:Python-logo-notext.svg>

### Django

Framework backend utilizado para gestionar toda la lógica del negocio. Controla las rutas, la autenticación, el manejo de sesiones con JWT, la comunicación con la base de datos mediante ORM y la lógica de los módulos (usuarios, vehículos, reservas, pagos, soporte).

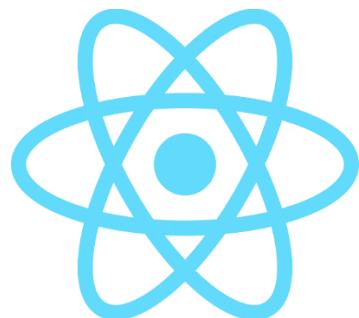


Ilus.3 Logo de django Extraido de

<https://www.fullstackpython.com/django.html>

### React:

React se emplea en el proyecto como la tecnología principal para desarrollar la versión móvil de Road Luxury. Gracias a su naturaleza basada en componentes reutilizables y su integración con **React Native**, permite crear un aplicativo móvil funcional que se conecta directamente al backend desarrollado en Django.



Ilus.4 logo de react extraído de

<https://es.wikipedia.org/wiki/Archivo:React-icon.svg>

### HTML5:

Lenguaje utilizado para estructurar el contenido de las páginas web del proyecto, como formularios, tablas, botones, catálogos y vistas del panel administrador y cliente.



Ilus.5 Logo de HTML5 extraido de

[https://www.flaticon.es/icono-gratis/html-5\\_919827](https://www.flaticon.es/icono-gratis/html-5_919827)

### CSS3:

Lenguaje de estilos usado para diseñar y dar apariencia visual a todas las interfaces, controlar colores, tamaños, posicionamiento, temas modernos y responsividad (vista móvil).



**CSS**



**Ilus.6 Logo de CSS3 extraido de**

[https://www.flaticon.es/icono-gratis/css\\_919826](https://www.flaticon.es/icono-gratis/css_919826)

**JavaScript:**

Lenguaje utilizado para agregar interacción, dinamismo y funciones rápidas en el navegador, como validaciones, menús móviles, botones dinámicos y animaciones.

**JS**



**Ilus.7 Logo de Javascript extraido de**

<https://1000marcas.net/javascript-logo/>

**SQLite:**

Almacena toda la información del sistema en un archivo único. Es ligero, rápido y perfecto para desarrollo local. Funciona totalmente integrado con el ORM de Django.



**Ilus.8 Logo de SQLite extraido de**

<https://1000marcas.net/sqlite-logo/>

**Requisitos del sistema:**

Para el Usuario Final (Cliente y Administrador)

Navegador web actualizado:

- Google Chrome
- Microsoft Edge
- Mozilla Firefox
- Safari (iOS)

Sistema operativo compatible:

- Windows 7, 8, 10, 11
- Android (cualquier versión moderna)
- iOS
- Linux

Conexión a Internet estable.

Soporte para JavaScript habilitado.

Soporte para CSS3 y HTML5 (todos los navegadores modernos lo cumplen).



### Procesos del software

#### . Proceso de Acceso al Sistema

1. El usuario abre un navegador web (Chrome, Edge, Firefox o Safari).
2. Ingresa la URL del sistema Road Luxury.
3. El navegador solicita la página al servidor.
4. El servidor entrega la interfaz inicial, mostrando el menú y las opciones de inicio de sesión o registro.
5. El usuario elige iniciar sesión o registrarse.

#### 2. Proceso de Registro de Usuario

1. El usuario selecciona la opción **Crear Cuenta**.
2. El sistema muestra un formulario donde el usuario ingresa: nombre, correo, teléfono y contraseña.
3. El usuario confirma los datos y envía el formulario.
4. El sistema valida la información e identifica si el correo ya existe.
5. Si todo es correcto, el sistema registra al usuario en SQLite mediante el ORM.
6. El sistema confirma la creación de la cuenta y redirige al inicio de sesión.

#### 3. Proceso de Inicio de Sesión

1. El usuario ingresa su correo y contraseña en el formulario.
2. El servidor valida las credenciales.
3. Si son correctas:
  - Se genera una sesión o token JWT.
  - Se identifica el rol (Cliente o Administrador).
4. El sistema redirige al usuario a su panel correspondiente.

#### 4. Proceso de Exploración del Catálogo de Vehículos

1. El usuario autenticado accede al catálogo.
2. El sistema consulta en SQLite la lista de vehículos disponibles.



## Manual Técnico

3. Se muestran imágenes, marca, categoría, precio, modelo y disponibilidad.
4. El usuario puede filtrar o seleccionar un vehículo.
5. El sistema carga la descripción completa del vehículo seleccionado.

### 5. Proceso de Gestión de Vehículos (Administrador)

1. El administrador ingresa al módulo de vehículos desde el panel.
2. Puede elegir **agregar, editar o eliminar** un vehículo.
3. Si agrega uno:
  - Completa el formulario con fotos y especificaciones.
  - El sistema guarda los datos en SQLite.
4. Si edita:
  - Modifica los campos deseados.
  - El sistema actualiza la información en la base.
5. Si elimina:
  - El sistema retira el vehículo del catálogo.

### 6. Proceso de Creación de Reserva

1. El usuario selecciona un vehículo disponible.
2. Ingresa las fechas de alquiler.
3. El sistema verifica disponibilidad en ese rango.
4. El sistema calcula automáticamente el costo total por días.
5. El usuario confirma la reserva.
6. El sistema registra la reserva en la base de datos y bloquea el vehículo para esas fechas.



## Manual Técnico

### 7. Proceso de Pago

1. El usuario ingresa al módulo de pagos asociado a su reserva.
2. El sistema muestra el valor total calculado.
3. El usuario ingresa los datos del pago (según el tipo implementado).
4. El sistema registra el pago en la base de datos.
5. Se genera un comprobante digital visible para el usuario y el administrador.

### 8. Proceso de Cancelación de Reservas

1. El usuario o administrador accede al módulo de reservas.
2. Selecciona la reserva que desea cancelar.
3. El sistema verifica si la reserva cumple condiciones para cancelación.
4. Si es válida, el sistema cambia su estado y libera el vehículo.
5. El usuario recibe confirmación.

### 9. Proceso de Soporte (PQRS)

1. El usuario ingresa al módulo de soporte.
2. Crea una nueva solicitud escribiendo su queja, reclamo o sugerencia.
3. El sistema registra la solicitud con fecha y estado “Pendiente”.
4. El administrador entra a la bandeja de PQRS.
5. Lee la solicitud y redacta la respuesta.
6. El sistema marca la solicitud como “Respondida”.
7. El usuario puede ver la respuesta desde su panel.

### 10. Proceso del Panel Administrativo

1. El administrador inicia sesión.
2. Accede al menú lateral donde encuentra las secciones:
  - o Usuarios
  - o Vehículos
  - o Reservas



## Manual Técnico

- Pagos
  - Soporte
3. Selecciona uno de los módulos para gestionar la información.
  4. El sistema realiza las operaciones mediante el ORM sobre SQLite.
  5. El administrador visualiza los resultados en tiempo real.

### 11. Proceso de Cierre de Sesión

1. El usuario selecciona la opción **Cerrar Sesión**.
2. El sistema elimina la sesión o token JWT.
3. Lo redirige a la página principal o login.

### Instalacion de aplicaciones

**Paso nro1:** Para iniciar la instalación de la aplicación, el primer paso consiste en **descargar el archivo comprimido del proyecto Road Luxury**, el cual contiene toda la estructura del sistema desarrollada en Django.

La ilustración siguiente muestra el archivo comprimido:

Road_Luxury	5/11/2025 10:24 p. m.	Carpeta comprimi...	1.805 KB
-------------	-----------------------	---------------------	----------

**Ilus9.Aplicacion de escritorio comprimida. Autoría propia**

**Paso nro2:** Una vez descargado el archivo, se debe **descomprimir la carpeta** para acceder a los módulos del sistema, las vistas, los modelos y el archivo de base de datos SQLite.

Road_Luxury	5/11/2025 10:25 p. m.	Carpeta de archivos
-------------	-----------------------	---------------------

**Ilus10.Aplicacion de escritorio descomprimida.Autoria propia**

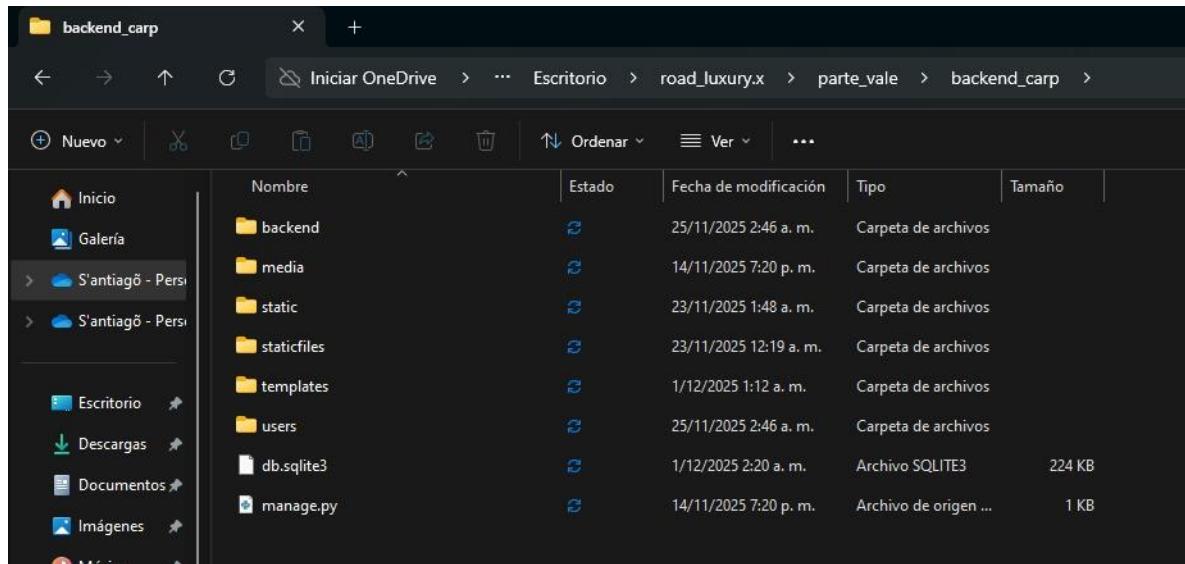


## Manual Técnico

**Paso nro3:** En este paso, se debe ingresar a la carpeta principal resultante de la descompresión.

Dentro de ella se encuentra toda la estructura base del proyecto, incluyendo:

- El archivo **db.sqlite3**, correspondiente a la base de datos.
- El archivo **manage.py**, encargado de ejecutar el servidor y gestionar comandos administrativos.
- Las carpetas de la aplicación, tales como:
  - usuarios
  - vehiculos
  - reservas
  - pagos
  - soporte
- Las carpetas media y static, utilizadas para almacenar imágenes y recursos del proyecto.



Nombre	Estado	Fecha de modificación	Tipo	Tamaño
backend	🕒	25/11/2025 2:46 a.m.	Carpeta de archivos	
media	🕒	14/11/2025 7:20 p.m.	Carpeta de archivos	
static	🕒	23/11/2025 1:48 a.m.	Carpeta de archivos	
staticfiles	🕒	23/11/2025 12:19 a.m.	Carpeta de archivos	
templates	🕒	1/12/2025 1:12 a.m.	Carpeta de archivos	
users	🕒	25/11/2025 2:46 a.m.	Carpeta de archivos	
db.sqlite3	🕒	1/12/2025 2:20 a.m.	Archivo SQLITE3	224 KB
manage.py	🕒	14/11/2025 7:20 p.m.	Archivo de origen ...	1 KB

Ilus.11. Contenido de la carpeta principal. Autoria Propia

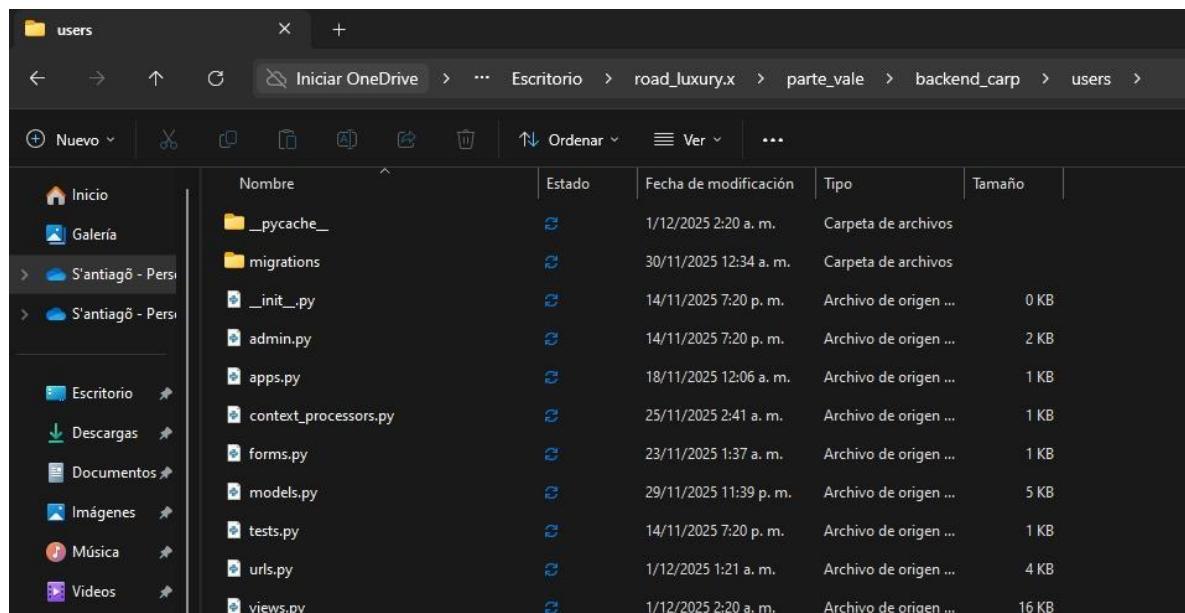


## Manual Técnico

**Paso nro4:** Dentro de la carpeta principal se encuentra todo el contenido del proyecto del lado del servidor (backend), gestionado por Django.

Aquí es posible visualizar:

- Archivos .py que manejan las vistas, modelos y rutas.
- El archivo de configuración **settings.py**.
- El archivo **urls.py**, encargado de definir las rutas del sistema.
- El archivo **db.sqlite3** donde se almacena toda la información.



Nombre	Estado	Fecha de modificación	Tipo	Tamaño
__pycache__		1/12/2025 2:20 a. m.	Carpeta de archivos	
migrations		30/11/2025 12:34 a. m.	Carpeta de archivos	
__init__.py		14/11/2025 7:20 p. m.	Archivo de origen ...	0 KB
admin.py		14/11/2025 7:20 p. m.	Archivo de origen ...	2 KB
apps.py		18/11/2025 12:06 a. m.	Archivo de origen ...	1 KB
context_processors.py		25/11/2025 2:41 a. m.	Archivo de origen ...	1 KB
forms.py		23/11/2025 1:37 a. m.	Archivo de origen ...	1 KB
models.py		29/11/2025 11:39 p. m.	Archivo de origen ...	5 KB
tests.py		14/11/2025 7:20 p. m.	Archivo de origen ...	1 KB
urls.py		1/12/2025 1:21 a. m.	Archivo de origen ...	4 KB
views.py		1/12/2025 2:20 a. m.	Archivo de origen ...	16 KB

**Ilus.12.Contenido de la Carpeta. Autoria propia**



## Manual Técnico

**Paso nro5:** Para ejecutar el sistema, se debe iniciar el servidor Django.

Los pasos son:

1. Abrir una terminal o ventana de PowerShell dentro de la carpeta principal del proyecto.
2. (Opcional) Activar el entorno virtual si el proyecto lo utiliza.
3. Ejecutar el siguiente comando:
4. `python manage.py runserver`

Al ejecutarlo, Django iniciará el servidor local y mostrará la dirección de acceso, generalmente:

```
(venv) PS C:\Users\Usuario\OneDrive\Escritorio\road_luxury.x\parte_vale\backend_carp> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 30, 2025 - 19:41:34
Django version 5.2.8, using settings 'backend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/
```

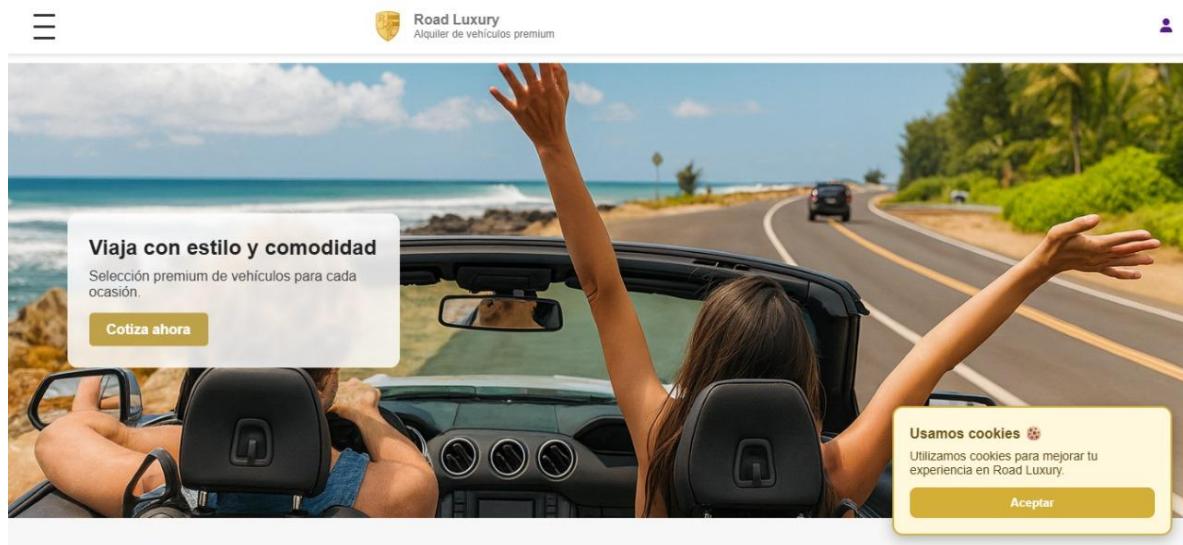
**Ilus.13 Ejecución del proyecto de escritorio o web. Autoría propia**



## Manual Técnico

**Paso nro6:** Para confirmar que la aplicación se ha ejecutado correctamente, el usuario debe ingresar en su navegador al enlace proporcionado por el servidor, donde podrá visualizar la **interfaz principal del sistema Road Luxury**, que generalmente corresponde a la pantalla de inicio de sesión.

Una vez cargada la página, se valida que el sistema está funcionando correctamente y que las rutas, vistas y base de datos responden de manera adecuada.

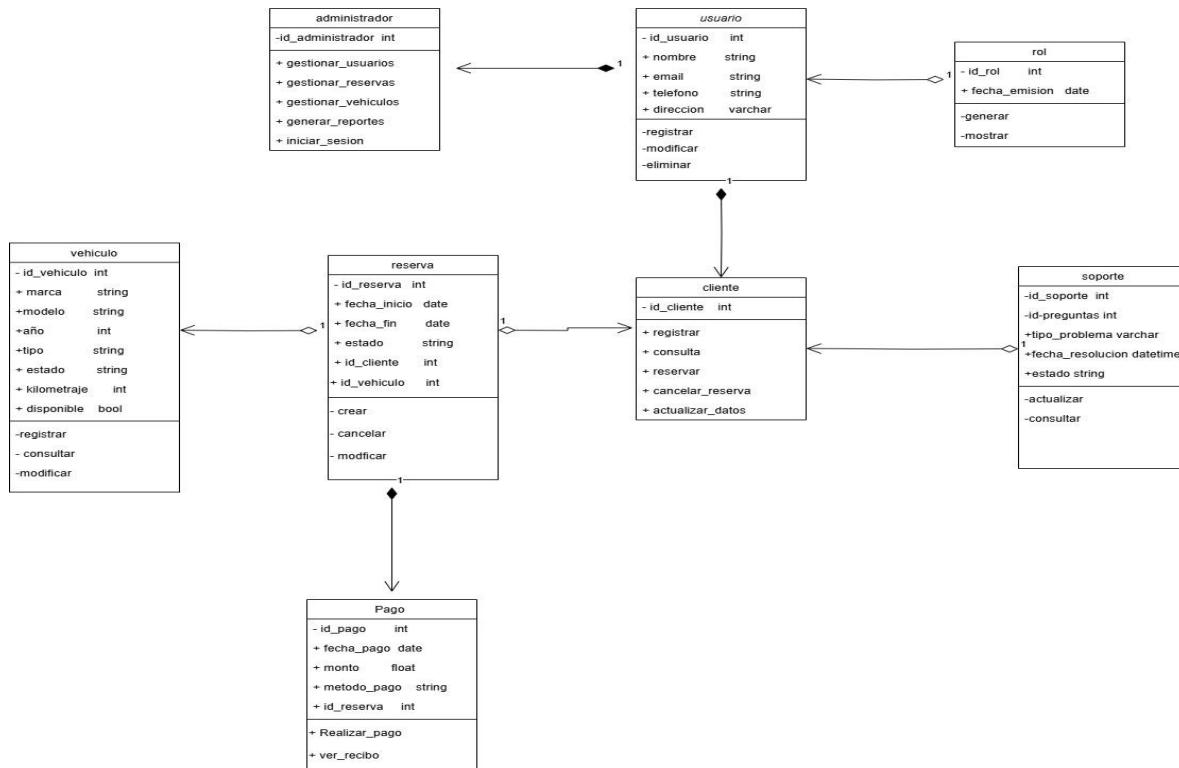


**Ilus.14 Interfaz principal cuando se realiza la ejecución del proyecto.Autoria propia.**



## Manual Técnico

### Diagramas de clases





## Manual Técnico

### Clase Rol

La clase *rol* representa los diferentes tipos de permisos que pueden asignarse a un usuario dentro del sistema. Contiene información como el identificador y la fecha de emisión del rol, los cuales permiten definir si un usuario actuará como cliente o como administrador. Esta clase se relaciona directamente con *usuario*, ya que un rol puede estar asociado a varios usuarios, mientras que cada usuario únicamente posee un rol específico.

### Clase Usuario

La clase *usuario* representa a cualquier persona registrada dentro del sistema Road Luxury. Contiene información personal como el nombre, el correo electrónico, el teléfono y la dirección del usuario, así como un identificador único que permite gestionar su registro, modificación o eliminación. La relación principal de esta clase se da con *rol*, indicando que cada usuario pertenece a un rol específico. Además, de esta clase derivan dos especializaciones: *administrador* y *cliente*, que heredan su información básica pero cuentan con comportamientos particulares.

### Clase Administrador

La clase *administrador* es una especialización de *usuario* que representa a las personas encargadas de gestionar completamente el sistema. Un administrador puede crear, modificar o eliminar usuarios, gestionar reservas, administrar vehículos y generar reportes, lo que lo convierte en la figura con mayor nivel de control. Su relación con el usuario es de herencia, ya que un administrador “es un usuario” pero con funciones adicionales. También se relaciona indirectamente con las demás clases, ya que supervisa y controla todas las operaciones del sistema.



## Manual Técnico

### Clase Cliente

La clase *cliente* es otra especialización de *usuario* que representa al usuario final que utiliza el sistema para reservar vehículos. Un cliente puede registrarse, consultar el catálogo, realizar y cancelar reservas, actualizar sus datos y generar solicitudes de soporte. Su relación principal es con *reserva*, dado que un cliente puede tener varias reservas asociadas, y con *soporte*, ya que puede generar múltiples solicitudes o reclamos dentro del sistema. La herencia desde *usuario* indica que también comparte su información personal básica.

### Clase Vehículo

La clase *vehiculo* representa a cada uno de los automóviles disponibles dentro del sistema de alquiler. Incluye información como la marca, el modelo, el año, el tipo, el estado del vehículo, el kilometraje y su disponibilidad. Cada vehículo puede estar vinculado a varias reservas, representando el proceso natural de que un mismo automóvil puede ser alquilado por diferentes clientes en diferentes fechas. La relación con *reserva* es de uno a muchos, ya que un vehículo puede aparecer en múltiples reservas, pero cada reserva solamente tiene un vehículo asociado.

### Clase Reserva

La clase *reserva* representa la acción mediante la cual un cliente solicita el alquiler de un vehículo durante un rango de fechas. Contiene información sobre el identificador de la reserva, la fecha de inicio, la fecha de finalización y el estado en el que se encuentra. Esta clase posee dos relaciones fundamentales: una con *cliente*, indicando que un cliente puede tener varias reservas a su nombre, y otra con *vehiculo*, indicando que un vehículo puede estar presente en varias reservas. Además, la reserva mantiene una relación uno a uno con *pago*, ya que cada reserva genera un único pago asociado.



## Manual Técnico

### Clase Pago

La clase *pago* representa el registro del pago que un cliente efectúa por una reserva. Incluye información como el identificador del pago, la fecha en que se realizó, el monto y el método utilizado. Su relación principal es con la clase *reserva*, ya que cada pago corresponde exactamente a una única reserva. Esta relación establece un vínculo directo entre ambas clases, definiendo que no puede existir un pago sin una reserva previamente creada.

### Clase Soporte

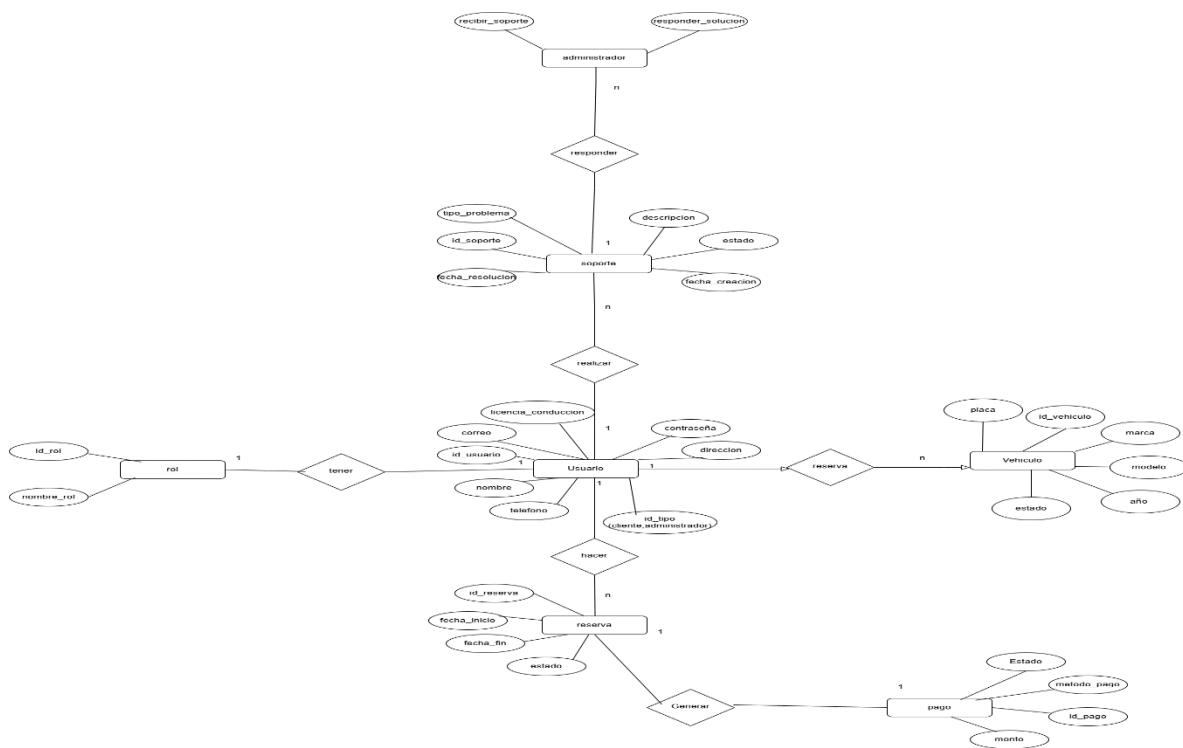
La clase *soporte* representa las solicitudes de ayuda, quejas, reclamos o sugerencias que los clientes envían al sistema. Contiene datos como el tipo de problema, el estado de la solicitud, la fecha en que se realizó y su resolución. Esta clase se relaciona con *cliente*, ya que un cliente puede generar múltiples solicitudes de soporte. Es decir, la relación es de uno a muchos, ya que un cliente puede registrar tantas solicitudes como necesite.



## Manual Técnico

### Modelo Entidad Relación

El **Modelo Entidad–Relación (MER)** sirve para representar de forma gráfica y organizada cómo se estructura la información dentro del sistema Road Luxury. Permite identificar claramente las entidades principales, sus atributos y las relaciones que existen entre ellas, facilitando el diseño de la base de datos. Gracias al MER, es posible comprender cómo se almacenan los datos, cómo se conectan entre sí (usuarios, roles, vehículos, reservas, pagos y soporte) y cómo fluye la información dentro del sistema. En resumen, el MER sirve como guía fundamental para construir una base de datos coherente, funcional y bien organizada.

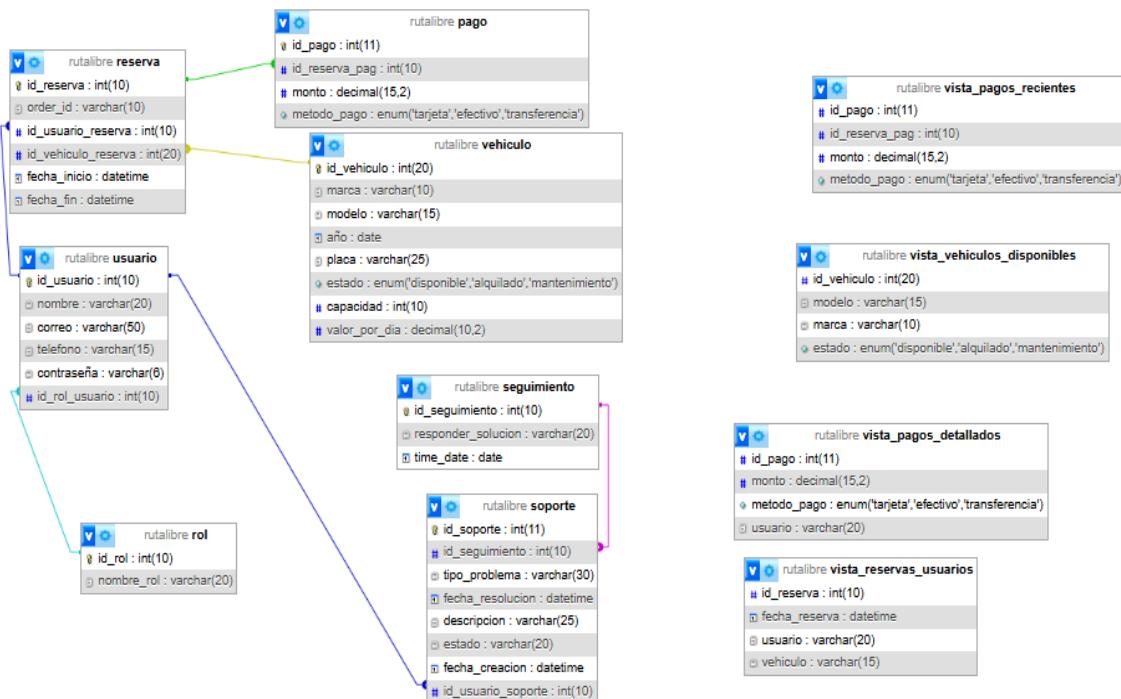




## Manual Técnico

### Modelo relacional

El **Modelo Relacional** sirve para transformar el diseño lógico del DER en una estructura formal de tablas que se implementará directamente en la base de datos del sistema Road Luxury. En este modelo se definen las tablas, sus columnas, las llaves primarias, llaves foráneas y la forma en que cada tabla se relaciona con las demás. Su función principal es garantizar que la base de datos pueda ser creada en un motor como SQLite de manera correcta, ordenada y sin inconsistencias. En pocas palabras, el Modelo Relacional convierte el diseño del DER en tablas reales listas para funcionar dentro del sistema.





## Manual Técnico

### Diccionario de datos

#### Rol

Campo	Tipo	Descripción
id	INTEGER PK	Identificador único del rol.
nom_rl	VARCHAR(20)	Nombre del rol (Administrador / Cliente).

#### Usuario

Campo	Tipo	Descripción
id	INTEGER PK	Identificador único del usuario.
nom_usr	VARCHAR(50)	Nombre completo del usuario.
ema_usr	VARCHAR(254) UNIQUE	Correo electrónico del usuario.
tel_usr	VARCHAR(15)	Número de teléfono.
psw_usr	VARCHAR(128)	Contraseña encriptada del usuario.
rol_id	BIGINT FK → users_rol.id	Rol asignado al usuario.



## Manual Técnico

### Vehículo

Campo	Tipo	Descripción
id	INTEGER PK	Identificador del vehículo.
name	VARCHAR(100)	Nombre o modelo del vehículo.
category	VARCHAR(50)	Categoría del vehículo (Deportivo, Premium, SUV, etc.).
year	INTEGER	Año del vehículo.
engine	VARCHAR(100)	Tipo de motor.
power	VARCHAR(50)	Potencia del vehículo.
transmission	VARCHAR(50)	Tipo de transmisión.
fuel	VARCHAR(50)	Tipo de combustible.
price	INTEGER	Precio por día del alquiler.
availability	VARCHAR(20)	Estado de disponibilidad.
specs	TEXT	Especificaciones adicionales.
image	VARCHAR(100)	Ruta de la imagen del vehículo.

### Reserva

Campo	Tipo	Descripción
id	INTEGER PK	Identificador de la reserva.
fci_rsv	DATE	Fecha de inicio de la reserva.
fch_rsv	DATE	Fecha de finalización de la reserva.
estado	VARCHAR(20)	Estado de la reserva (Activa, Cancelada, Finalizada).
usuario_id	BIGINT FK → users_usuario.id	Usuario que realiza la reserva.
vehiculo_id	BIGINT FK → users_vehiculo.id	Vehículo reservado.



## Manual Técnico

### Pago

Campo	Tipo	Descripción
id	INTEGER PK	Identificador del pago.
mnt_pag	DECIMAL	Monto del pago.
mtd_pag	VARCHAR(20)	Método de pago (efectivo, tarjeta, etc.).
fecha_pag	DATETIME	Fecha y hora en que se procesó el pago.
reserva_id	BIGINT FK → users_reserva.id	Reserva a la que corresponde el pago.
procesado_por_id	BIGINT FK → users_usuario.id	Administrador que procesó el pago.

### Contrato

Campo	Tipo	Descripción
id	INTEGER PK	Identificador del contrato.
archivo_pdf	VARCHAR(100)	Ruta del archivo PDF del contrato.
firmado	BOOLEAN	Indica si el contrato fue firmado.
fecha_creacion	DATETIME	Fecha en que se generó el contrato.
fecha_firma	DATETIME	Fecha en que se firmó.
reserva_id	BIGINT FK UNIQUE → users_reserva.id	Reserva asociada al contrato (1 a 1).
usuario_id	BIGINT FK → users_usuario.id	Usuario que firma el contrato.



## Manual Técnico

### Seguimiento

Campo	Tipo	Descripción
id	INTEGER PK	Identificador del seguimiento.
rsp_sgm	VARCHAR(50)	Estado o respuesta del seguimiento. (Ej: Abierto, En proceso, Cerrado).

### Soporte

Campo	Tipo	Descripción
id	INTEGER PK	Identificador de la solicitud de soporte.
tp_spt	VARCHAR(30)	Tipo de soporte (Queja, Reclamo, Pregunta, Sugerencia).
fch_spt	DATE	Fecha en que se generó la solicitud.
dsc_spt	TEXT	Descripción detallada del problema.
cre_spt	DATE	Fecha de creación del registro.
seguimiento_id	BIGINT FK → users_seguimiento.id	Estado del proceso de seguimiento.



## Manual Técnico

### Glosario

#### Administrador

Usuario con permisos avanzados que puede gestionar vehículos, reservas, pagos, usuarios y solicitudes de soporte dentro del sistema.

#### API

Interfaz de Programación de Aplicaciones. Permite la comunicación entre diferentes componentes del sistema, especialmente si se conecta a una app móvil en el futuro.

#### Autenticación

Proceso mediante el cual un usuario ingresa al sistema confirmando su identidad con correo y contraseña.

#### Base de Datos

Conjunto organizado de información estructurada que almacena los datos del sistema, como usuarios, vehículos, reservas y pagos.

#### Cliente

Usuario final del sistema que puede realizar reservas, consultar vehículos y generar solicitudes de soporte.

#### Contrato

Documento digital generado por el sistema para formalizar el alquiler de un vehículo por parte de un usuario.



### CRUD

Acrónimo de *Create, Read, Update, Delete*. Operaciones básicas de creación, consulta, edición y eliminación usadas en el sistema.

### Disponibilidad

Estado que indica si un vehículo está disponible para ser alquilado o se encuentra reservado.

### Django

Framework de desarrollo en Python utilizado para construir el backend del sistema Road Luxury.

### Entidad

Elemento del modelo de datos que representa un objeto del mundo real, como un usuario, vehículo o reserva.

### Front-End

Parte visual del sistema con la que interactúa el usuario final, construida con HTML, CSS y JavaScript.

### Glosario

Listado de términos técnicos y su significado utilizado para aclarar conceptos dentro de la documentación del sistema.



### **Llave Foránea**

Campo dentro de una tabla que referencia la llave primaria de otra tabla para relacionar datos.

### **Llave Primaria**

Campo único en una tabla que identifica de manera exclusiva cada registro (por ejemplo, el ID de un usuario o vehículo).

### **Migración**

Proceso de Django que permite crear o modificar la estructura de la base de datos a partir de los modelos.

### **Modelo**

Estructura de datos definida en Django que representa una tabla dentro de la base de datos.

### **Panel Administrativo**

Sección del sistema exclusiva para administradores donde se gestionan usuarios, vehículos, reservas, pagos y soporte.

### **Pago**

Registro financiero que confirma la cancelación de una reserva por parte del cliente.



## Manual Técnico

### Reserva

Proceso mediante el cual un usuario arrienda un vehículo por un periodo de tiempo determinado.

### Rol

Categoría que define el nivel de permisos del usuario dentro del sistema (cliente o administrador).

### Seguimiento

Estado asociado a una solicitud de soporte que indica si está abierta, en proceso o solucionada.

### Sesión

Identificador temporal generado cuando un usuario inicia sesión para mantener su acceso al sistema.

### Soporte

Módulo del sistema donde el usuario puede reportar problemas, dudas o solicitudes que deben ser atendidas por un administrador.

### SQLite

Motor de base de datos ligero utilizado por Django para almacenar toda la información del sistema Road Luxury.



## Manual Técnico

### **Usuario**

Persona registrada en el sistema que puede acceder a sus funcionalidades según su rol.

### **Vehículo**

Elemento principal del sistema de alquiler que contiene información como categoría, modelo, motor, precio y disponibilidad.

### **Vista**

Componente de Django encargado de procesar solicitudes del usuario y devolver respuestas al navegador.

### **Transmisión**

Tipo de sistema interno del vehículo (manual, automática, etc.) utilizado como parte de sus especificaciones.

### **Token**

Identificador utilizado en autenticación mediante JWT para validar la identidad de un usuario.

### **UI / Interfaz de Usuario**

Elemento visual que permite al usuario interactuar con el sistema.



## Manual Técnico

### Referencias

Uso de HTML

[https://developer.mozilla.org/es/docs/Learn\\_web\\_development/Core/Structuring\\_content/Basic\\_HTML\\_syntax](https://developer.mozilla.org/es/docs/Learn_web_development/Core/Structuring_content/Basic_HTML_syntax)

Django introducción

[https://developer.mozilla.org/es/docs/Learn\\_web\\_development/Extensions/Server-side/Django/Introduction](https://developer.mozilla.org/es/docs/Learn_web_development/Extensions/Server-side/Django/Introduction)

CSS

[https://developer.mozilla.org/es/docs/Learn\\_web\\_development/Getting\\_started/Your\\_first\\_website/Styling\\_the\\_content](https://developer.mozilla.org/es/docs/Learn_web_development/Getting_started/Your_first_website/Styling_the_content)

Sqlite

[https://www-sqlite-org.translate.goog/about.html?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://www-sqlite-org.translate.goog/about.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

¿Qué es un software libre?

<https://pingback.com/es/resources/software-libre/>

React

[https://developer.mozilla.org/es/docs/Learn\\_web\\_development/Core/Frameworks\\_libraries/React\\_getting\\_started](https://developer.mozilla.org/es/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started)

Bases de datos



## Manual Técnico

<https://support.microsoft.com/es-es/topic/conceptos-b%C3%A1sicos-del-dise%C3%B1o-de-una-base-de-datos-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5>

Guia Python

<https://docs.python.org/es/3/tutorial/>



Centro Industrial y Desarrollo  
**Empresarial Soacha**  
Regional Cundinamarca



## Manual Técnico