



---

# PROYECTO DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA

---



Autor/a/es:

**Jorge Sánchez Berrocoso**

Tutor/a:

Departamento de Informática y Comunicaciones  
**I.E.S. Luis Vives**

# 1. Introducción

---

El sector de alquiler de automóviles es una industria en constante evolución que desempeña un papel fundamental en la movilidad de las personas en la actualidad. La creciente demanda de soluciones flexibles de transporte y la creación de experiencias de viaje personalizadas han impulsado el interés en este mercado. En este contexto, el presente proyecto de fin de grado tiene como objetivo desarrollar una tienda de alquiler de automóviles innovadora y orientada al cliente, adaptada a las necesidades cambiantes de movilidad y a las tendencias del mercado.

Esta documentación presenta el resultado de un arduo trabajo de investigación, diseño y desarrollo de una plataforma de alquiler de automóviles que busca destacarse en un mercado altamente competitivo. Nuestro enfoque se basa en la incorporación de tecnología de vanguardia, un servicio al cliente excepcional y la creación de una experiencia de usuario sin igual. El proyecto se enmarca en un contexto en el que la sostenibilidad, la eficiencia y la conveniencia son aspectos cruciales para los consumidores, por lo que nuestra tienda de alquiler de automóviles abordará estos factores en su oferta de servicios.

En las próximas secciones, se detallarán los aspectos clave del proyecto, desde la conceptualización inicial hasta la implementación tecnológica. Se abordarán los desafíos específicos que enfrentamos, las soluciones propuestas y los resultados obtenidos. Este proyecto no solo busca cumplir con los requisitos académicos, sino también sentar las bases para una posible implementación real en el mercado, brindando a los interesados una visión completa y práctica de cómo una tienda de alquiler de automóviles puede prosperar en la era de la movilidad moderna.

## 2. Requisitos

---

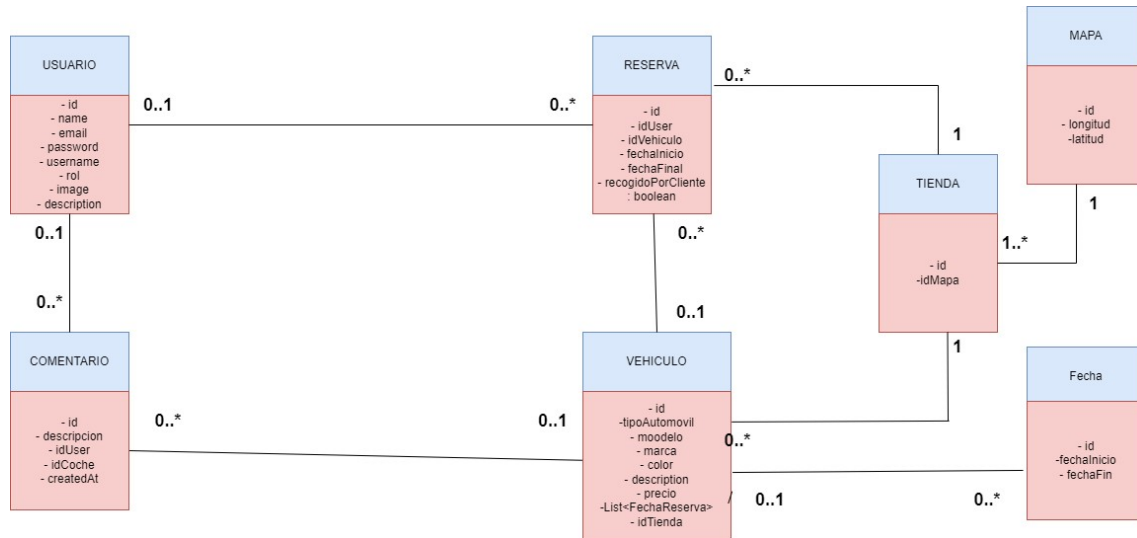
### 1. Requisitos Funcionales

- Registro e Inicio de Sesión:
  - Los usuarios deben poder crear una cuenta personal en la plataforma.
  - Debe existir un proceso de inicio de sesión seguro y eficiente.
- Sistema de Reservas en Tiempo Real:
  - Los usuarios deben poder seleccionar fechas de inicio y finalización del alquiler.
  - El sistema debe mostrar la disponibilidad de coches en tiempo real.
  - Debe ser posible confirmar reservas de manera conveniente.
- Comentarios y Calificaciones:
  - Los usuarios deben poder dejar reseñas y calificaciones sobre los coches y sus experiencias de alquiler.
  - Estas reseñas y calificaciones deben ser visibles para otros usuarios.
- Catálogo de Coches en Línea:
  - La plataforma debe presentar un catálogo completo de coches disponibles para alquiler.
  - Los usuarios deben poder ver detalles de cada vehículo, incluyendo imágenes, especificaciones y precios.
  - El administrador debe tener la capacidad de gestionar el catálogo, incluyendo agregar, eliminar y actualizar automóviles así como usuarios.
- Mapa de Ubicación:
  - La plataforma debe incluir un mapa interactivo que muestre la ubicación de las tiendas de alquiler de coches.

### 2. Requisitos No Funcionales

- Front-end desarrollado en Angular.
- Back-end desarrollado en Kotlin.
- La seguridad del back-end con JWT Bearer.
- Notificaciones de comentarios en tiempo real.

### 3. Diagrama de Clases



## 4. Competencia

---

### Hertz:

Hertz es una de las empresas de alquiler de automóviles más conocidas a nivel mundial. Su aplicación móvil permite a los usuarios reservar coches en una amplia variedad de ubicaciones y ofrece opciones de alquiler a corto y largo plazo. Proporciona un amplio catálogo de vehículos, desde económicos hasta de lujo.



### Enterprise Rent-A-Car:

Enterprise es otra empresa líder en el alquiler de coches. Su aplicación permite a los usuarios buscar vehículos por tipo, marca y ubicación. También ofrecen opciones de alquiler a largo plazo y programas de fidelidad para clientes frecuentes.



### Turo:

Turo se diferencia al permitir que los propietarios de automóviles alquilen sus vehículos a otros usuarios. Su plataforma conecta a arrendadores y arrendatarios, lo que brinda una amplia variedad de opciones de automóviles, desde automóviles económicos hasta vehículos exóticos.



## 4. Elección Tecnológica

---

### Creación API Rest

Para la creación de la API Rest tenemos dos opciones:

#### Spring

Es el framework más utilizado en Java.

- Ventajas:
  - Utiliza el patrón de inyección de dependencias.
  - Está dividido en módulos que nos ayudan a gestionar diferentes necesidades.
  - Permite el desarrollo de aplicaciones flexibles y escalables, por lo que el mantenimiento del programa siempre será más fácil.
  - Está basado en la programación orientada a interfaces.
- Desventajas:
  - Cuando hay actualizaciones del framework cuesta mucho trabajo migrarlo a la siguiente versión.
  - Es complejo para poder utilizarlo.
  - Si no tienes experiencia, cuesta un poco aprenderlo.
  - Utiliza muchos XML.



#### Ktor

Framework para crear servidores y clientes asíncronos creado por JetBrains.

- Ventajas:
  - Código abierto.
  - Fácil de empezar con Ktor.
  - Utiliza DSL.
  - Ligero.
  - Flexible.
  - Multiplataforma.
  - Uso de corrutinas.
  - Buena documentación.
- Desventajas:
  - Solamente se puede utilizar el lenguaje Kotlin.
  - Muy nuevo.



Para la creación de la API Rest he escogido Spring.

Porque permite el desarrollo de aplicaciones flexibles y escalables, por lo que el mantenimiento del programa siempre será más fácil, usa el lenguaje kotlin y es fácil de empezar a utilizarlo.

Además es un framework que he utilizado anteriormente y me gustaría poder mejorar y descubrir más sobre él.

## Base de datos

### MySQL

Es una base de datos relacional, actualmente uno de los más utilizados y de código abierto.

- Ventajas:
  - Es multiplataforma
  - Requiere de pocos recursos (Hardware)
  - Tiene un buen rendimiento en volúmenes de datos medianos o pequeños.
  - Realizar las consultas rápidamente.
  - Baja probabilidad de corromper los datos.
  - Esquema fijo.
  - Seguro.
  - Soporte de transacciones.
- Desventajas:
  - En las modificaciones de las estructuras de la BDD pueden aparecer pequeños errores.
  - Mal rendimiento con cantidades de datos muy grandes.
  - No es intuitivo, se necesita un estudio previo para poder entenderlo.



### MariaDB

Es una versión mejorada de MySQL

- Ventajas:
  - Mejor rendimiento con cantidades de datos más grandes.
  - Mas seguridad de MySQL.
  - Código abierto.
  - Es más fácil de usar que MySQL.
- Desventajas:
  - No es compatible con todas las aplicaciones.
  - Poca ayuda de soporte.



### Postgresql

- Ventajas:
  - Gratuito.
  - Sencillo de utilizar.
  - Multiplataforma.
  - Gran rendimiento.
  - Es un sistema robusto y fiable.
  - Escalable.
- Desventajas:
  - Rendimiento lento para bases de datos pequeñas.
  - No tiene buen soporte técnico.
  - Necesario tener una buena base del lenguaje SQL.



# MongoDB

Es una base de datos NoSQL ( no sólo SQL )

- Ventajas:
  - Necesita pocos recursos.
  - Código abierto.
  - Buena documentación.
  - Esquema no fijo.
  - Escalable.
  - Flexible por su buena adaptación a evoluciones.
  - Optimización en las consultas con grandes cantidades de datos.
  - Basado en JavaScript
- Desventajas:
  - No tiene joins para las consultas.
  - No admite todas las consultas SQL.
  - No es correcta para las transacciones complejas.



Para el proyecto he elegido MongoDB

Ya que es escalable por lo que cuando aumenten los datos de localización no habrá tantos problemas, son más rápidas, son flexibles ya que las bases de datos NoSQL no siguen ningún esquema y tiene una alta disponibilidad por lo que cuando haya muchos accesos a la base de datos el rendimiento no se verá tan afectado como podría estarlo con otras bases de datos.



## Front-end

### Flutter

- Ventajas:
  - Rápido desarrollo.
  - Recarga de los cambios en caliente.
  - Multiplataforma.
  - Utiliza librerías de widgets.
  - Rendimiento nativo.
  - Open Source.
- Desventajas:
  - Lenguaje de programación Dart.
  - Pesa más que otros frameworks o aplicaciones nativas.
  - Puede haber dificultad para integrar en el proyecto librerías nativas de terceros.
  - Limitación de integración con hardware.



### Jetpack Compose

- Ventajas:
  - Acelera el desarrollo de Interfaz de Usuario.
  - Adaptabilidad del dispositivo.
  - Se usa menos código.
  - Librería independiente.
  - Compatible con código XML.
  - Compatible con todo el código existente.
- Desventajas:
  - Problemas con el tiempo de compilación en la interoperabilidad.
  - Problemas con el tamaño de aplicación en la interoperabilidad.
  - Al ver la vista previa siempre tiene que renderizar.
  - Algunos componentes no son compatibles.
  - Más lento que XML para representar un cambio.



### Angular

- Ventajas:
  - Utiliza TypeScript.
  - Desarrollo rápido.
  - Tiene compatibilidad para el desarrollo de aplicaciones móviles o de escritorio.

- Desarrollo web más rápido.
- Soporte de Google.
- Pruebas sencillas de escribir.
- Productivo.
- Más opciones de lenguajes para escribir el código.



- Desventajas:
  - Puede llegar a ser complejo.
  - Puede tener un aprendizaje un poco difícil.
  - Aplicaciones relativamente grandes.

## VueJS

- Ventajas:
  - Pesa poco.
  - División en componentes individuales.
  - Fácil de aprender.
  - Fácil de utilizar.
  - Reactividad.
- Desventajas:
  - Necesario combinarlo con Electron para hacer aplicaciones de escritorio y móviles.



## React

- Ventajas:
  - Acelera el desarrollo de las aplicaciones.
  - JavaScript.
  - Código abierto.
  - Recarga de cambios en caliente.
  - Buen rendimiento.
  - Apariencia nativa.
- Desventajas:
  - Problemas para depurar.
  - Necesario desarrolladores nativos.
  - Depende de su desarrollador Facebook.
  - No muy buena gestión de la memoria.
  - Actualización lenta de funciones recientes.
  - Mal rendimiento con funciones complejas.



Para la realización del Front-end del proyecto utilizaré Angular ya que es un framework muy utilizado en España, utiliza Typescript, es de desarrollo rápido, tiene mucha comunidad y me puede servir para el futuro.

