



UNIÓN EUROPEA  
Fondo Social Europeo  
El FSE invierte en tu futuro

# **MEMORIA FINAL DEL PROYECTO**

## ***AutoSelect: Soluciones de Alquiler Premium***

### **CICLO FORMATIVO DE GRADO SUPERIOR**

#### **DESARROLLO DE APICACIONES MULTIPLATAFORMA**

#### **CURSO 2022-2023**

**AUTOR:**

**Jorge Sánchez Berrocoso**

**TUTOR:**

**DEPARTAMENTO DE INFORMÁTICA Y COMUNICACIONES**

**I.E.S. LUIS VIVES**



# 1. Resumen

---

El presente proyecto de fin de grado se enfoca en crear una tienda innovadora y centrada en el cliente para el alquiler de automóviles. Esto responde a la necesidad creciente de soluciones de transporte flexibles y experiencias de viaje personalizadas.

La documentación abarca todo el proceso de investigación, diseño y desarrollo de una plataforma de alquiler de vehículos. El proyecto se destaca por su integración de tecnología avanzada, un servicio al cliente excepcional y una experiencia de usuario inigualable.

En un contexto donde la sostenibilidad, la eficiencia y la conveniencia son cruciales para los consumidores, el proyecto enfoca estos aspectos en su oferta de servicios. Se exploran en detalle los desafíos a los que se enfrentó el equipo, así como las soluciones propuestas y los resultados obtenidos.

El objetivo es no solo cumplir con los requisitos académicos, sino también establecer las bases para una posible implementación real en el mercado. El proyecto ofrece una visión completa y práctica de cómo una tienda de alquiler de automóviles puede prosperar en la era moderna de la movilidad.



## 2. 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.



### 3. Justificación

---

Durante mi formación académica, he detectado carencias específicas en el sector del alquiler de vehículos. Existe una ausencia notable de herramientas que faciliten el alquiler eficaz y rápido de coches, especialmente para grupos.

Este proyecto se orienta hacia el diseño, desarrollo y puesta en marcha de una plataforma web para una empresa de alquiler de vehículos. El objetivo es enriquecer la experiencia del usuario, simplificando y agilizando el proceso de localización, elección y reserva de vehículos. Además, se busca mejorar la administración del negocio optimizando la gestión de las reservas y la flota vehicular.



## 4. Objetivos

---

Los objetivos centrales del proyecto se articulan de la siguiente manera:

**Autenticación y Registro de Usuarios:** Implementar un mecanismo de registro que permita a los usuarios establecer una cuenta personalizada y acceder a la plataforma a través de un sistema de autenticación seguro y eficiente.

**Reservas en Línea en Vivo:** Crear un sistema interactivo de reservas que habilite a los usuarios para escoger y reservar fechas específicas, verificar la disponibilidad de los vehículos en tiempo real y concretar la reserva con facilidad.

**Evaluaciones y Retroalimentación:** Incorporar una funcionalidad de comentarios y puntuaciones, posibilitando a los clientes compartir sus opiniones y experiencias sobre los vehículos y el servicio, lo que a su vez, servirá de guía para futuros usuarios.

**Catálogo Digital de Vehículos:** Desarrollar una interfaz de usuario intuitiva y visualmente atractiva que exhiba un inventario exhaustivo de vehículos disponibles, con detalles como imágenes, características técnicas y tarifas, mientras se ofrece a los administradores herramientas eficaces para la gestión del catálogo, incluyendo la adición, eliminación y actualización de los datos de los coches.

**Geolocalización y Direcciones:** Añadir un componente de mapa interactivo que muestre la ubicación de las tiendas de alquiler.

**Crecimiento Profesional:** Este proyecto representa una oportunidad significativa para abordar retos en los ámbitos técnico y de diseño, proporcionando un terreno fértil para el desarrollo y refinamiento de mis habilidades como profesional emergente en tecnología.



## 5. Alcance

---

El alcance del proyecto "Optimización de la Experiencia del Cliente en el Alquiler de Vehículos" incluye hitos estratégicos como:

**Desarrollo de una Plataforma Integral:** Elaborar una página web intuitiva que agilice el proceso de búsqueda, selección y reserva de vehículos a través de una interfaz en línea.

**Sistema de Reservas Avanzado:** Establecer un sistema de reservas ágil y en tiempo real, proporcionando a los usuarios un método eficaz y sencillo para asegurar sus vehículos preferidos.

**Interfaz de Usuario Innovadora:** Construir una interfaz visualmente atractiva y fácil de navegar que ofrezca un catálogo detallado de los coches disponibles, incluyendo imágenes vívidas, especificaciones técnicas y precios.

**Comunicación Mejorada:** Mejorar la interacción directa entre los clientes y la empresa de alquiler mediante un sistema de retroalimentación en cada ficha de vehículo.

**Desarrollo Profesional:** Aprovechar la oportunidad para ganar experiencia valiosa en el desarrollo de soluciones web y móviles, lo que contribuirá al crecimiento de habilidades técnicas en el ámbito tecnológico.

**Impacto en la Industria:** Aportar al mercado de alquiler de coches en línea mediante la elevación de la experiencia del cliente y la optimización de la operatividad, generando beneficios para proveedores y consumidores del sector.



## 6. Requisitos

---

### 1. Requisitos Funcionales

#### **Autenticación de Usuarios:**

- Implementación de una funcionalidad de registro que permita la creación de cuentas personales en la plataforma.
- Proceso de inicio de sesión robusto y ágil para garantizar la seguridad y eficiencia del acceso a la plataforma.

#### **Gestión de Reservas Dinámica:**

- Capacidad para que los usuarios elijan y programen las fechas de recogida y devolución de los vehículos.
- Visualización en tiempo real de la disponibilidad de vehículos, permitiendo a los usuarios reservar de manera práctica y rápida.

#### **Sistema de Feedback Integrado:**

- Posibilidad para los usuarios de publicar opiniones y valoraciones sobre los vehículos y la experiencia de alquiler.
- Accesibilidad a las reseñas y puntuaciones por parte de otros usuarios para facilitar una toma de decisiones basada en experiencias previas.

#### **Catálogo Digital Interactivo:**

- Exhibición de un catálogo completo y actualizado de vehículos disponibles, con detalles exhaustivos como imágenes, especificaciones técnicas y precios.
- Herramientas administrativas para la gestión eficiente del catálogo, permitiendo añadir, eliminar y actualizar la información de los vehículos y usuarios.



## 2. Requisitos No Funcionales

### Desarrollo Web y Móvil:

- Construcción del front-end utilizando el framework Angular para una experiencia de usuario óptima.
- Programación del back-end con Kotlin, asegurando un rendimiento eficiente y mantenible.

### Seguridad Informática Avanzada:

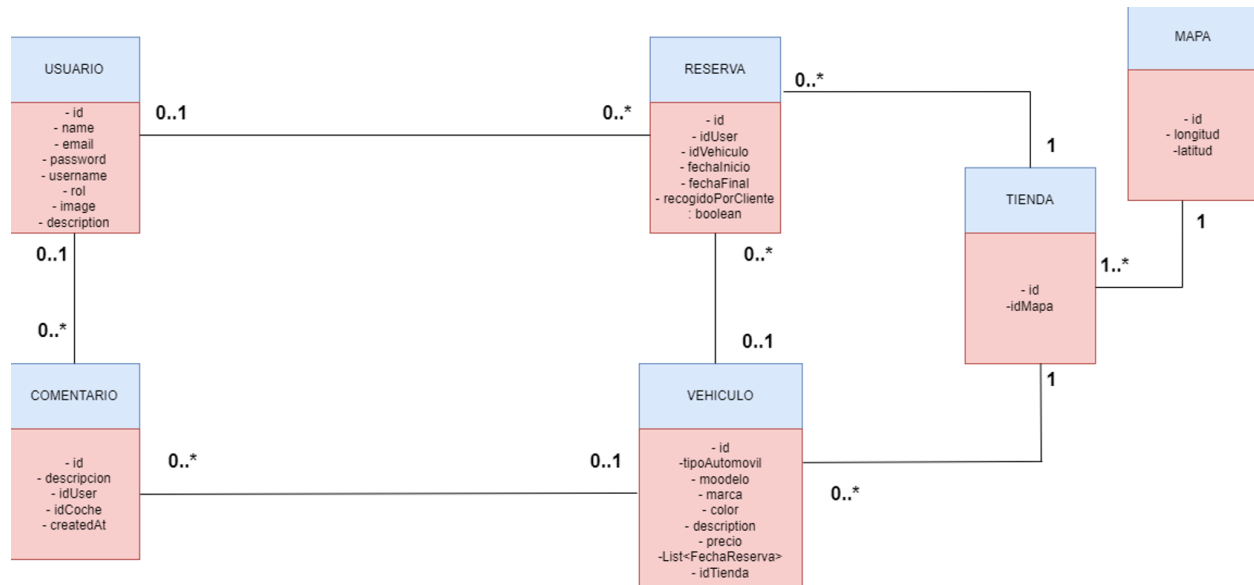
- Seguridad del back-end reforzada con la implementación de tokens JWT Bearer.

### Interactividad y Notificaciones:

- Sistema de notificaciones en tiempo real para mantener informados a los usuarios sobre nuevos comentarios y actualizaciones en la plataforma.



## 7. Diagrama de Clases



## 8. 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.





UNIÓN EUROPEA  
Fondo Social Europeo  
El FSE invierte en tu futuro

### 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.

Rent Better Cars



**Turo.com**



## 9. 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.



mongoDB®

**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.







UNIÓN EUROPEA  
Fondo Social Europeo  
El FSE invierte en tu futuro

- 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.**

# 10. Despliegue

---

## 1. Contratación de Servicios de Hosting y Dominio:

Se seleccionó Hetzner como proveedor de servicios de hosting debido a su rendimiento, estabilidad y costo-efectividad. Para la plataforma de alquiler de vehículos, se ha registrado el dominio **alquilaenmadrid.com**, lo que proporciona una identificación clara y relevante para los usuarios que buscan servicios de alquiler en la región de Madrid.

The image shows the Hetzner logo, which consists of the word "HETZNER" in a bold, red, sans-serif font.

## 2. Preparación del Servidor y Despliegue de la Aplicación:

El artefacto de la aplicación, un archivo ejecutable .jar, contiene tanto el front-end como el back-end de la aplicación. Al ejecutarse, despliega los recursos del front-end en la carpeta `resources/static`, siguiendo una estructura de proyecto estándar de Spring Boot que facilita la ejecución autónoma de la aplicación.

## 3. Configuración de Nginx para Redirección de Tráfico:

Utilizamos Nginx como servidor web y proxy inverso para gestionar el tráfico entrante. Mediante la configuración de Nginx, se redirige todo el tráfico que llega al puerto 80 (HTTP) al puerto 443 (HTTPS), asegurando que la comunicación con el servidor se realice de manera segura utilizando el protocolo TLS/SSL.





#### 4. Implementación de Certificados SSL con Let's Encrypt:

Para fortalecer la seguridad, se obtuvieron certificados SSL a través de Let's Encrypt, una autoridad de certificación que proporciona certificados gratuitos para la habilitación de HTTPS. El proceso de obtención y renovación de certificados se automatizó utilizando Certbot, una herramienta que simplifica este proceso y garantiza que la comunicación entre el usuario y el servidor sea cifrada.





## 5. Dockerización y Orquestación con Docker Compose:

Para facilitar el despliegue y la escalabilidad, la aplicación y el servidor web Nginx se contenerizaron utilizando Docker. Se creó un archivo docker-compose.yml que define los servicios necesarios para ejecutar la aplicación, incluyendo la imagen de Nginx y el contenedor que ejecuta el archivo .jar de la aplicación. Docker Compose gestiona la creación, configuración y ejecución de estos contenedores, proporcionando un entorno de despliegue coherente y replicable.

```
Version: '3.9'

services:
  nginx:
    image: nginx:latest
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
      - /docker/scripts/B00T-INF/classes/static:/docker/scripts/B00T-INF/classes/static
      - /etc/letsencrypt/live/alquilaenmadrid.com/fullchain.pem:/etc/ssl/certs/alquilaenmadrid.com.pem
      - /etc/letsencrypt/live/alquilaenmadrid.com/privkey.pem:/etc/ssl/private/alquilaenmadrid.com.key
    depends_on:
      - automoviles
    networks:
      - default

  automoviles:
    image: openjdk:17
    restart: unless-stopped
    volumes:
      - /docker/automoviles/data:/app
      - /docker/automoviles/data/logs:/opt/aplicaciones/automoviles/application/web/logs
    ports:
      - "8080:8080"
    command: java -jar /app/Backend-Automoviles-0.0.1-SNAPSHOT.jar
    networks:
      - default

networks:
  default:
    driver: bridge
```



## 6. Configuración del Archivo de Nginx:

El archivo de configuración de Nginx (nginx.conf) se personalizó para definir el comportamiento del servidor web y proxy inverso. Esta configuración especifica detalles como la ubicación de los archivos de certificado SSL, las políticas de redirección y la configuración del servidor que escucha en los puertos 80 y 443, entre otros aspectos críticos para la operatividad y seguridad del sistema.

```
events {
    worker_connections 1024;
}

http {
    include        /etc/nginx/mime.types;
    default_type   application/octet-stream;

    upstream app {
        server automoviles:8080;
    }

    server {
        listen 80;
        server_name aquilaenmadrid.com;
        return 301 https://$server_name$request_uri;
    }

    server {
        listen 443 ssl;
        server_name aquilaenmadrid.com;

        ssl_certificate /etc/ssl/certs/aquilaenmadrid.com.pem;
        ssl_certificate_key /etc/ssl/private/aquilaenmadrid.com.key;

        location / {
            root /docker/scripts/BOOT-INF/classes/static;
            index index.html;
            try_files $uri $uri/ /index.html;
        }

        location @app {
            proxy_pass http://app;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection 'upgrade';
            proxy_set_header Host $host;
            proxy_cache_bypass $http_upgrade;
        }
    }
}
```



## 7. Resumen del Proceso de Despliegue:

En resumen, el proceso de despliegue implica:

- Subir el archivo .jar al servidor contratado.
- Ejecutar Docker Compose para levantar los contenedores de Nginx y la aplicación.
- Configurar Nginx para redirigir el tráfico de HTTP a HTTPS y para servir el front-end desde resources/static.
- Asegurar la comunicación mediante certificados SSL obtenidos y renovados automáticamente.
- Probar y verificar la configuración para asegurar que el despliegue sea exitoso y seguro.

Esta infraestructura y estrategia de despliegue garantiza que la aplicación de alquiler de vehículos sea accesible, eficiente y segura para todos los usuarios finales.

