



Universidad Politécnica  
de Madrid



**Escuela Técnica Superior de  
Ingenieros Informáticos**

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Plataforma para el Alquiler de Coches a  
través de Tecnologías Web, On-Premise  
y Móvil: parte móvil**

Autor: <<nombre y apellidos>> Ricardo López Crespo

Tutor(a): <<nombre y apellidos>> Alejandro Rodríguez  
González

Madrid, Abril 2021<<mes-año>>

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en ~~<<título del grado>>~~ Ingeniería Informática

Título: Plataforma para el Alquiler de Coches a través de Tecnologías Web, On-Premise y Móvil: parte móvil

~~Título del Trabajo, con Mayúscula en Todas las Palabras que no Sean Conectivas (Artículos, Preposiciones, Conjunciones)~~

~~MAbril es Año2021~~

Autor: ~~<<nombre y apellidos>>~~ Ricardo López Crespo

Tutor:

Alejandro Rodríguez González

~~<<nombre y apellidos>>~~

~~<<departamento>>~~

ETSI Informáticos

Universidad Politécnica de Madrid

# Resumen

El objetivo del trabajo de fin de grado es desarrollar una plataforma para el alquiler de vehículos. Esta constará de diferentes subsistemas que permitan la gestión de diferentes funcionalidades, entre las cuales están: acceso aplicación administrador y acceso aplicación móvil. Este proyecto será realizado por dos alumnos simultáneamente en el que habrá una parte común compuesta por: creación de una base de datos SQL, una API REST para interactuar con la misma y la aplicación ya mencionada para los administradores, esta será una aplicación de escritorio programada en java y usando como apoyo la librería SWT para el front-end.

Por otra parte, este proyecto en particular, además de las tareas comunes se encargará de desarrollar una aplicación en Android con Android Studio, esta aplicación servirá como cliente para el sistema de alquiler de coches planteado, mientras que la aplicación de escritorio realizada en común ejercerá la función de administrador.

Entre las operaciones que realizarán están:

- Para la aplicación del administrador:
  - Log in.
  - Consultar facturación.
  - Registrar coches.
  - Registrar trabajadores.
  
- Para la aplicación del cliente:
  - Log in.
  - Buscar ofertas.
  - Realizar/Modificar/Cancelar una reserva.

Con este proyecto se busca implementar un sistema que se podría comparar con otros actuales, ya que se emplean tecnologías actuales, es multiplataforma siguiendo con las necesidades actuales de los usuarios y sigue con los estándares de desarrollo e implementación.

# Abstract

The aim of the final degree project is to develop a platform for car rental. This will consist of different subsystems that allow the management of different functionalities, among which are: administrator application access and mobile application access. This project will be carried out by two students simultaneously in which there will be a common part composed of: creation of a SQL database, a REST API to interact with it and the application already mentioned for admins, this will be a programmed desktop application in java and using as support the SWT library for the front-end.

On the other hand, this particular project, in addition to the common tasks, will be responsible for developing an application on Android with Android Studio, this application will serve as a client for the proposed car rental system, while the desktop application carried out in common will exercise the administrator role.

Among the operations they will carry out are:

- For the administrator application:
  - o Log in.
  - o Check billing.
  - o Register cars.
  - o Register workers.
  
- For customer application:
  - o Log in.
  - o Search for offers.
  - o Make / Modify / Cancel a reservation.

This project seeks to implement a system that could be compared with other modern ones, since current technologies are used, it is multiplatform and is based on development and implementation standards.

# Tabla de contenidos

<b>1</b>	<b>Introducción.....</b>	<b>1</b>
<b>2</b>	<b>Desarrollo .....</b>	<b>3</b>
2.1	Planificación.....	3
2.1.1	Diagrama de Gantt.....	3
2.1.2	Organización previa .....	4
2.2	Diseño y lanzamiento Base de Datos .....	5
2.2.1	Diseño Base de Datos .....	5
2.2.2	Creación y lanzamiento Base de Datos .....	6
2.3	Desarrollo API REST.....	7
2.3.1	Métodos y funciones .....	7
2.3.2	Despliegue del servicio .....	7
2.3.2.1	Configuración .....	7
2.3.2.2	Problemas encontrados.....	7
2.4	Desarrollo Aplicación Escritorio.....	8
2.4.1	Back-end .....	8
2.4.2	Front-end.....	8
2.5	Desarrollo Aplicación Móvil .....	9
<b>3</b>	<b>Resultados y conclusiones .....</b>	<b>10</b>
<b>4</b>	<b>Análisis de Impacto .....</b>	<b>11</b>
<b>5</b>	<b>Bibliografía .....</b>	<b>12</b>
<b>6</b>	<b>Anexos.....</b>	<b>13</b>

# 1 Introducción

Este proyecto surgió debido a que a ambos nos gustaba la idea de hacer una aplicación, en concreto hacer una que interactuase con una base de datos para gestionar sus datos. En mi caso prefería hacer una aplicación web mientras que mi compañero se inclinaba más hacia las aplicaciones móviles. Debido a que nuestra idea era muy parecida decidimos idear un proyecto algo más complejo suficiente para darnos trabajo a ambos y pudiéramos presentarlo como TFG. Finalmente, acabamos con la idea de realizar un sistema de alquiler de coches, los clientes accederían a este servicio por medio de una aplicación web o de móvil, que estas a su vez usarían una api para hacer las operaciones en la base de datos. A parte de eso decidimos implementar una aplicación en java que asumiría el rol de administrador dentro de este sistema y actuaría directamente sobre la base de datos.

Las elecciones de las herramientas usadas han sido, MySQL Workbench y xampp para la base de datos on-premise, java para la aplicación de escritorio, y yo en mi caso usare html, css y javascript para la aplicación web. La elección ha sido esta debido a que son herramientas y lenguajes que hemos estado usando durante la carrera y son tecnologías que actualmente son muy usadas en el mundo laboral como se puede ver en los gráficos a continuación.

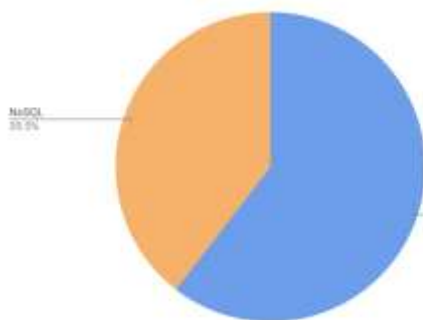
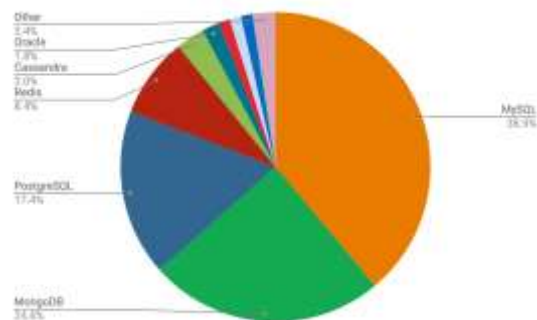


Figura 1 – SQL vs NoSQL.



*Figura 2 - Bases de datos más populares.*

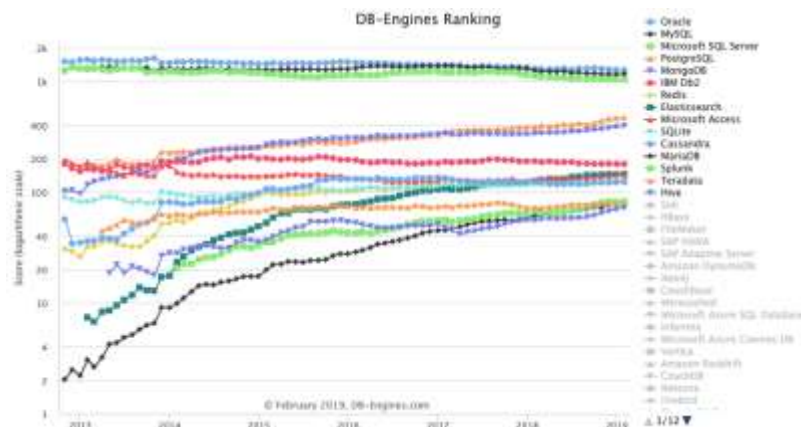


Figura 3 - Diagrama bases de datos.

Por otro lado, la implementación de una API REST que converja todas las funcionalidades principales de nuestras plataformas tanto web como móvil. Este enfoque de desarrollo conocido como “REST” aparece en la mayoría de proyectos actuales ya que es el estándar más habitual en la creación de servicios

Web (Figura 4 y 5), esto se sirve de peticiones HTTP para obtener o generar datos tanto en XML como en JSON. Las ventajas que ofrece son varias: separación entre el cliente y servidor que provoca una mayor capacidad de escalabilidad y flexibilidad a la hora de trabajar y, además, esta API es independiente al tipo de plataforma y lenguaje de programación utilizado.

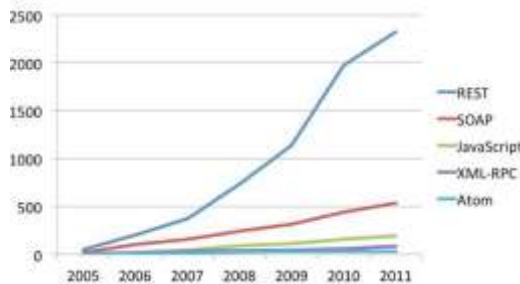


Figura 4 - Tendencia uso REST.

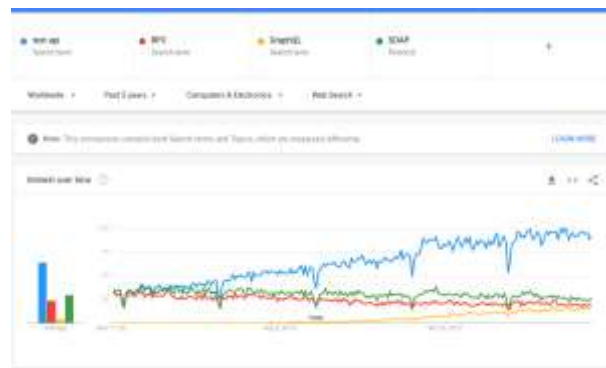


Figura 5 - Búsquedas de REST frente a otros.

Debido a ser un trabajo conjunto añadimos una aplicación de escritorio específicamente para los administradores, esta estará programada en Java y con la ayuda de SWT realizamos el “front-end”. Este software permitirá a sus usuarios realizar una gestión completa de la base de datos, permitiendo la gestión de los usuarios, trabajadores, filiales y reservas.

Por último, en cuanto a la parte del cliente y en concreto mi parte específica es la del cliente en Android, este software permitirá a los usuarios tras iniciar sesión multitud de funciones, entre las cuales están: realización de una reserva, consulta de reservas, búsqueda de ofertas, etc. En cuanto a por qué escoger Android, destacar que este Sistema Operativo es un sistema de código abierto en el que tienes más libertad a la hora de modificar o mejorar las “apps”, además, aunque ha aparecido un nuevo lenguaje conocido como “Kotlin”, tiene sus bases en Java, tiene una cantidad de usuarios inmensa (Figura 6 y 7) y permite una administración de la memoria, permitiendo así una mayor efectividad del uso de la misma.



Figura 6 - Tendencia en ventas por SO.

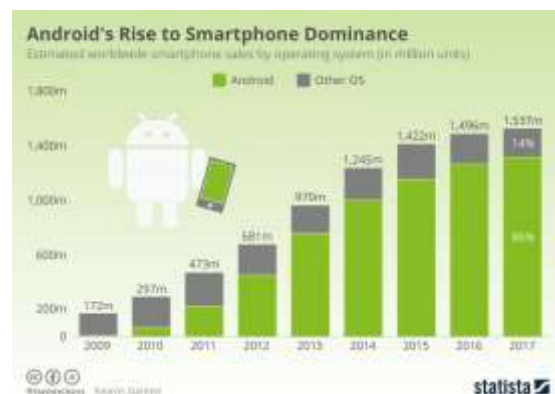


Figura 7 - Dominancia de Android

## 2 Desarrollo

### 2.1 Planificación

Para la planificación del proyecto decidimos organizar todas las tareas y organizar la estructura del sistema, tanto base de datos como funcionalidades de los sistemas. Para la organización de las tareas realizamos un diagrama de Gantt estructurado por semanas y por otro lado unos archivos detallando el funcionamiento.

#### 2.1.1 Diagrama de Gantt



Figura 8 - Diagrama de Gantt inicial

Como se puede ver en la Figura 8, organizamos las tareas en base al tiempo disponible de 13 semanas, y pensamos en una planificación jerárquica en la que empezamos pensando en el diseño de la aplicación y terminamos con las aplicaciones.

En primer lugar, se elaboró una serie de archivos que se verán en el siguiente apartado **2.1.2 Organización previa**, en los que se detallaban a rasgos generales y como una aproximación inicial el formato de la base de datos, las funcionalidades de los administradores y las funcionalidades de los clientes.

A partir de ahí, en las siguientes semanas se realizaría las tareas relacionadas con la base de datos, en esta se aborda el diseño de la misma, creación de esta y el volcado de datos.

A continuación, para poder tener las funcionalidades necesarias para el cliente, comenzaría la fase de desarrollo de la API REST en la cual serían necesarias el diseño de los métodos y la programación de estos.



Por último, sería el momento de la parte de las aplicaciones que corresponderían tanto la del administrador de escritorio como la del cliente en Android.

Sin embargo, pese a esta organización inicial, ha habido ciertos cambios en la misma debido a problemas o retrasos encontrados. Ante esos inconvenientes nos hemos adaptado rápidamente, empezando a realizar tareas de las aplicaciones para poder compensar el tiempo e intentar seguir la planificación lo más posible.



Figura 9 - Nuevo diagrama de planificación

Como se puede observar en el nuevo diagrama a mitad de proyecto, aparece un problema con la configuración del servidor, esto provocó una reacción que consistió en comenzar con el front-end de la aplicación del administrador, buscando adelantar trabajo debido a ese bloqueo que permita. Igualmente se siguieron desarrollando los métodos al mismo tiempo.

### 2.1.2 Organización previa

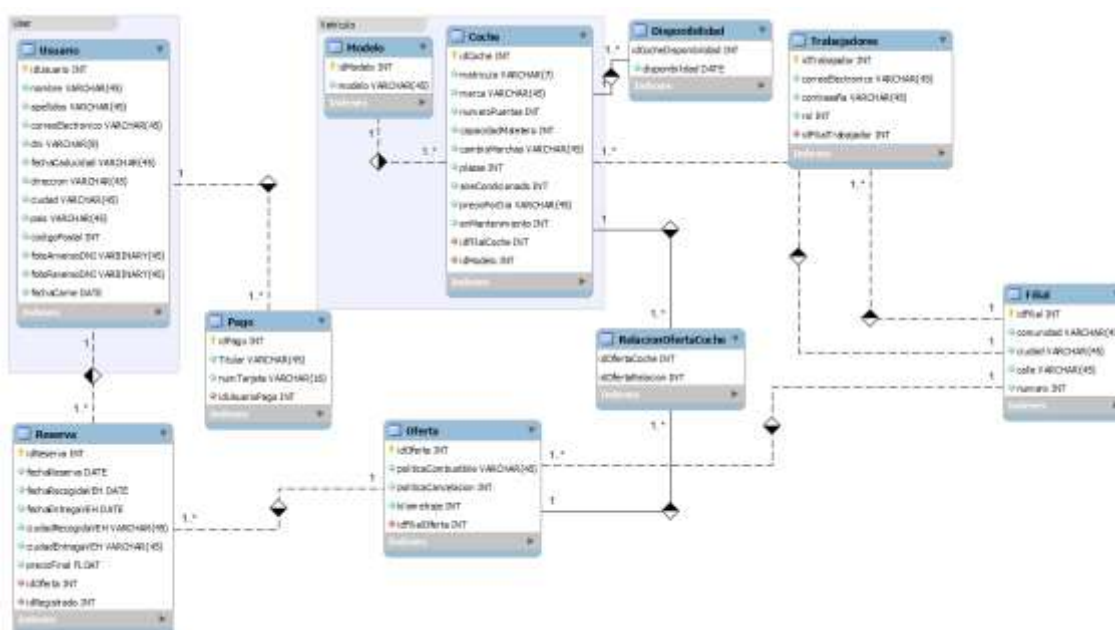
Para tener una estructura organizada del proyecto y desarrollar de forma incremental el software, en primer lugar, tuvimos una reunión en la que se decidió a rasgos generales y viendo ejemplos de otras compañías con sus sistemas, que requisitos necesitaríamos. Gracias a esto se resolvió con una primera aproximación a la estructura de la base de datos, la cual se vería modificada con implementaciones posteriores. Por otro lado, a partir de esa primera estructura, se elaboraron unos documentos con las principales funcionalidades que desarrollaríamos en nuestras aplicaciones, en cada una, se pensó tanto en el flujo de datos, como en la propia jerarquía de clases. Por último, se realizaron unos primeros bocetos para las distintas plataformas, en los que quedaría reflejado el paso de pantallas y los distintos “widgets” disponibles.

## 2.2 Diseño y lanzamiento Base de Datos

En este apartado, se abordarán todos los aspectos relacionados con la base de datos, en primer lugar, se presentará el diseño de la base de datos y los motivos de esa estructura y, en segundo lugar, el procedimiento para crear y lanzar la misma para que esté disponible para nuestras aplicaciones.

### 2.2.1 Diseño Base de Datos

Para realizar el diseño de la base de datos se empleó SQL Workbench, herramienta que nos permite a la vez que se diseña el diagrama de tablas que conforman esta. A su vez, este programa nos permite la posibilidad de generar un script para el lanzamiento.



*Figura 10 - Diagrama EER*

Tabla “Usuario”:

En esta se encuentran todos los datos pedidos en el registro de los clientes, en ella también se encuentran datos relevantes para la empresa, ya sea para tener identificados a sus usuarios o para contactarlos.

Tabla “Reserva”:

Aquí se encuentran guardados datos como la fecha en la has hecho la reserva, la fecha en la que recogerás el coche de la filial, la fecha en la que lo devolverás o el precio final de la reserva. A su vez, esta tabla estará conectada al coche, las condiciones de la reserva y a los datos de pago.

Tabla “Pago”:

Esta tabla es simple y contiene los datos básicos de pago, estará relacionada con la reserva, y actuará de “puente” entre la reserva y el usuario que la hizo.

Tabla “CocheOfertado”:

Contendrá todos los datos del coche a reservar, así como los de las condiciones de la oferta como pueden ser la política de cancelación, gratuita o no, o la cantidad de kilometro que puedes recorrer con el vehículo. Esta estará conectada al modelo del coche que se oferta y a la filial donde se ofrece esta oferta.

Tabla “Modelo”:

Actualmente no hay prácticamente datos en ella, sin embargo, está diseñada para almacenar las imágenes que se usaran en las aplicaciones cliente para mostrar el vehículo en cuestión. Decidimos crear esta tabla ya que los modelos es un dato que se puede repetir numerosas veces en los coches.

Tabla “Disponibilidad”:

Esta tabla representa las fechas en las cuales un coche ya ha sido reservado, la primera columna representa el coche, y las dos siguientes son las fechas entre las cuales el coche ya ha sido reservado.

Tabla “Trabajadores”:

En esta se encontrarán tanto los datos con los que un trabajador iniciara sesión, así como su rol dentro de la empresa y la filial en la que trabaja

Tabla “Filial”:

Esta tabla la he mencionado varias veces con anterioridad, en ella se encuentran los datos de las filiales de la empresa, de esta manera se sabrá donde trabajan los empleados, o poder saber si los coches ofertados están en tu ciudad.

### **2.2.2 Creación y lanzamiento Base de Datos**

Tras investigar formas de configurar y lanzar la base de datos, decidimos hacerla con MySQL Workbench y xampp, ya que en la asignatura de bases de datos era así como lanzábamos el servidor donde se encontraba la BDD. También vimos que se podía hacer en linux con comandos y descargándote un plugin en vscode, sin embargo, hacerlo de la primera forma parecía más fácil en windows, lo único que hay que hacer para lanzarla es abrir el xampp y encender los servidores de apache y mysql.

A la hora de configurar la base de datos tuvimos que abrir una conexión con el workbench con un usuario y una contraseña, que más tarde usaremos para acceder con el jdbc, y una vez teniendo la conexión solo hay que ejecutar un script de sql que creara la estructura de tablas y sus atributos y conexiones.

## **2.3 Desarrollo API REST**

### **2.3.1 Métodos y funciones**

### **2.3.2 Despliegue del servicio**

#### **2.3.2.1 Configuración**

#### **2.3.2.2 Problemas encontrados**

## **2.4 Desarrollo Aplicación Escritorio**

### **2.4.1 Back-end**

### **2.4.2 Front-end**

## **2.5 Desarrollo Aplicación Móvil**

### **3 Resultados y conclusiones**

Resumen de resultados obtenidos en el TFG. Y conclusiones personales del estudiante sobre el trabajo realizado.

## 4 Análisis de Impacto

En este capítulo se realizará un análisis del impacto potencial de los resultados obtenidos durante la realización del TFG en diferentes contextos:

- personal
- empresarial
- social
- económico
- medioambiental
- cultural

En dicho análisis se destacarán los beneficios esperados, así como también los posibles efectos perjudiciales.

Se recomienda analizar también el potencial impacto haciendo referencia a los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 que sean relevantes para la solución propuesta (<https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>).

En este capítulo se destacarán también aquellas decisiones tomadas a lo largo del trabajo que tienen como base la consideración del impacto.



## 5 Bibliografía

Publicaciones utilizadas en el estudio y desarrollo del trabajo.

Hay que utilizar un sistema internacional para referencias bibliográficas, de acuerdo con las indicaciones del tutor. Por ejemplo, el [sistema de IEEE](#).

<https://www.azulschool.net/8-motivos-por-los-cuales-aprender-sql-es-una-excelente-inversion/>

<http://highscalability.com/blog/2019/3/6/2019-database-trends-sql-vs-nosql-top-databases-single-vs-mu.html>

<https://lccopen.tech/5-ventajas-programar-para-android/#:~:text=Programar%20para%20Android%20es%20%E2%80%9Csalta rse,de%20aplicaciones%20que%20sus%20competidores.&text=Adem%C3%A1 s%2C%20otra%20ventaja%20de%20Play,y%20llamativo%20para%20los%20u suarios>

<https://www.campusmvp.es/recursos/post/buenas-razones-para-aprender-sql.aspx>

<https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/#:~:text=En%20la%20actualidad%20no%20existe,REST%20y%20la s%20APIs%20REST>

## **6 Anexos**

Este capítulo es opcional, y se escribirá de acuerdo con las indicaciones del Tutor.