



# Problema Lab 04

Java

# INSTALA UNA BASE DE DATOS CON DOCKER, CREA ENTIDADES Y GENERA REPOSITORIOS

En este lab aprenderemos todo sobre el proceso de instalación de bases de datos con Docker, cómo se trabaja la incorporación de dependencias y de qué manera se debe llevar a cabo la configuración de conexiones. Para ello, te animamos a que sigas desarrollando tu app de registro de vehículos con nuevas funcionalidades.

## Objetivos de este ejercicio

- Seguir desarrollando de forma autónoma la aplicación en Java propuesta en el ejercicio y asentar todos los conocimientos expuestos por el profesor.
- Entender cómo funcionan las operaciones con bases de datos y qué estructuras o pasos se deben seguir para su configuración.

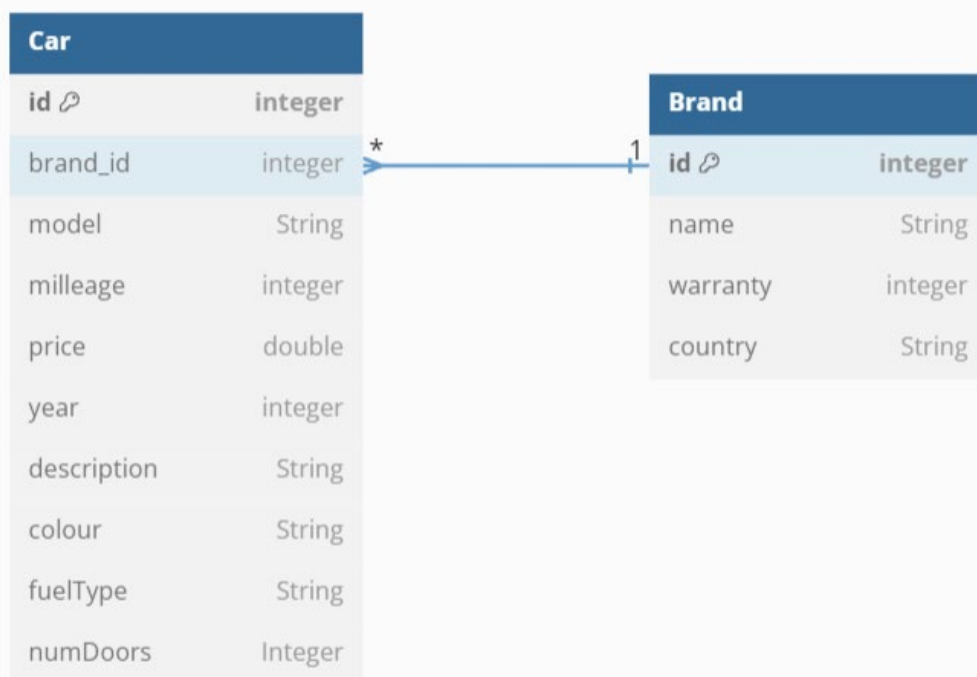
## Descripción de la actividad

Como ya hemos mencionado anteriormente, te pedimos que sigas trabajando en tu app de registro de vehículos. Llegados a este punto, toca añadir nuevas funcionalidades. ¡Toma nota de la siguiente petición! Debes incluir una conexión con una base de datos, así como las distintas operaciones para que se pueda trabajar sin dificultades con los datos.

Por otra parte, también te vamos a pedir que generes la base de datos, ya sea de forma manual por medio de un *script* o, si te apasionan los retos como a nosotros, con *liquibase* (lo dejamos a tu elección). No olvides que, para trabajar dicha base de datos, debes tener en consideración sus entidades, las relaciones entre ellas y los repositorios correspondientes.

A continuación, te compartimos las condiciones generales que debes tener en cuenta durante tu proceso de trabajo. ¡Toma nota!

1. Crea la base de datos a partir de dos tablas a partir del esquema que encontrarás justo debajo de estas líneas.



2. Configura la conexión a la BD con tu proyecto. Incluye las dependencias y propiedades que sean necesarias.

3. Crea las entidades que hagan referencia a las tablas de la base de datos, así como las relaciones entre ellas.
4. Añade los métodos y *endpoints* que se requieren para poder realizar un CRUD (*Create, Read, Update, Delete*) sobre los elementos de la base de datos.
5. Gestiona correctamente los códigos de respuesta para cada caso.

### **Formato de entrega**

Envía tu ejercicio en un archivo con extensión .java o, si has utilizado más de una clase para resolver la actividad, un archivo comprimido en formato .zip o .rar con el conjunto de clases empleadas durante la resolución del problema.

### **Criterios de corrección**

Te compartimos algunos puntos que debes comprobar sí o sí cuando vayas a autoevaluar tu ejercicio antes de darlo por resuelto:

1. en primer lugar, ten en cuenta que la aplicación debe funcionar a la perfección;
2. que la conexión con la base de datos se ha realizado de manera correcta. Para tener una trazabilidad de las llamadas recuerda que puedes valerte de los *logs*.



[Qualentum.com](https://www.qualentum.com)