El proyecto deberá **llamarse como el alumno** seguido de la cadena 1TRIM, por ejemplo, si yo me examinase, el proyecto tendría que llamarse **ManuelRico1TRIM.**

El proyecto será exportado **en formato .zip o .rar.**

**El examen tiene una duración de 3 HORAS, la entrega fuera de tiempo y forma acarrea el suspenso automático de la prueba.**

**No se permite utilizar ningún plugin de IA ni ninguna herramienta más allá del IDE, Postman, WAMP y un programa ofimático para leer este enunciado.**

1. Crea los Entities principales **1 punto**:

Motor:

* Integer id.
* Enumerado tipo => V6, W6, V12, ELECTRICO, HIDROGENO
* String fabricante.

Coche:

* Integer id.
* String modelo.
* Enumerado tipo => DEPORTIVO, UTILITARIO, SUV
* String matricula => 0000 AAA.

Taller:

* Integer id.
* String nombre.

Es necesario poder saber qué motor lleva cada coche. Cada coche podrá llevar únicamente un motor. Un motor tan sólo podrá pertenecer a un coche, darse cuenta de que tiene id, es decir, puede haber muchos motores v12 fabricados con Mercedes, todos y cada uno de ellos con un id distinto.

Un taller puede albergar muchos coches, mientras que un coche puede pasar **a lo largo del tiempo** por distintos talleres, de este modo, **será importante saber qué coche pasó por qué taller en qué fecha.**

1. Utiliza Spring Data para establecer las relaciones entre los entities que sean necesarios (**2 puntos**).
2. Crea las clases necesarias (y los INSERT neesarios) para que, usando Spring Rest, podamos hacer uso de los siguientes endpoints:
   1. “/coches/all” devuelva una lista con los coches de nuestra base de datos. (0.75 puntos).
   2. “/coches/id” devuelva el coche que tiene ese id, en caso de no existir, deberá devolver un código “NO CONTENT”. (1 punto).
   3. “/coches/guardar” debe guardar el coche que se le pasa en la base de datos. (0.5 puntos).
   4. “/coches/borrar/id” debe eliminar de la bbdd el coche con ese id y, en caso de no existir, tendrá que devolver un código “NO CONTENT”, en caso de eliminarlo bien, con un “OK” será suficiente. (0.5 puntos).
   5. “/coches/actualizar” deberá devolver el coche de la base de datos con los valores actualizados, en caso de haber algún problema con la actualización, escoged el código que creáis conveniente. (1 punto).
3. Haz un endpoint que sea “/historial/anhoMax” de tal manera que, pasando un año de una fecha, devuelva todos los coches que estuvieron en algun taller antes de ese año, ANTES. (1 punto).
4. Crea un endpoint que sea “/coches/tipo” y devuelva el tipo de coches que se pasa por la URL. Es decir, “/coches/DEPORTIVO” devolvería todos los deportivos. (1 puntos).
5. Haz un endpoint que sea “/talleres/nombre” de tal manera que pasando por parámetro el nombre del taller, veamos las propiedades del taller y todo el historial de coches que han pasado por ese taller. (1.25 puntos).