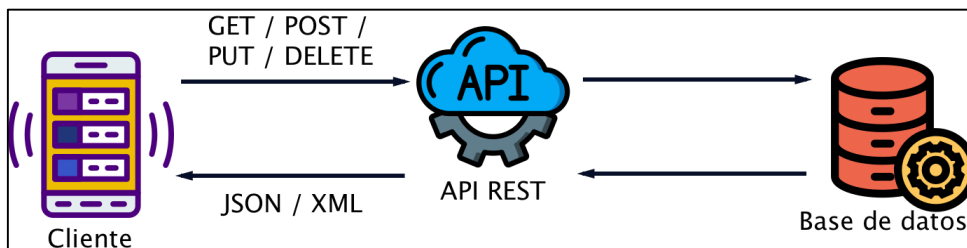


## API REST CRUD



El objetivo de esta actividad es que cada uno de vosotros implemente una API REST con las características vistas en clase usando Spark y GSON, que realice las operaciones CRUD sobre un tipo de objeto que tenéis que inventar vosotros, manipulando la persistencia con Hibernate e implementando las técnicas vistas en clase. Para ello, tenéis que codificar una clase que represente una entidad con al menos las siguientes características:

- **Identificador tipo Long auto generado.**
- **Valor de tipo String único que no puede ser nulo.**
- **Valor numérico (con o sin decimales) que no puede ser nulo.**
- **Una fecha de creación con formato aaaa/mm/dd.**
- **Una imagen asociada que se guarde en un servidor de internet.**
- **Una categoría o clase (varios objetos pueden pertenecer a una misma categoría o clase).**
- **Cualquier otro atributo que veais conveniente.**

Queda terminantemente prohibido que los objetos de la entidad programada sean algún tipo de producto de una tienda para venderse. Es obligatorio darle otro contexto.

Para ello disponéis de un fichero en el campus que contiene ejemplos de posibles bases de datos a modelar del cual podreis elegir alguno.

Tened en cuenta que esto lo utilizaremos de base para la actividad final del modulo

Los endpoints mínimos que tienen que implementar la API serán los siguientes:

- **Listar todos los objetos existentes (los datos se devuelven paginados incluyendo el total de elementos, el número de elementos por página y el número de página actual).**
- **Listar todos los datos que cumplan un filtro tipo LIKE.**
- **Listar todos los datos que cumplan un filtro tipo IN donde un parámetro es una lista.**
- **Consulta que use una función de agregación (avg, count, min, max, etc).**
- **Listar los objetos que cumplan una determinada condición (la que queráis)**

incluyendo en la respuesta solo algunos atributos de los objetos (Clase DTO).

- **Crear nuevo objeto (hacerlo de manera transaccional).**
- **Borrar un objeto a partir del ID (hacerlo de manera transaccional).**
- **Actualizar/modificar un objeto a partir del ID (hacerlo de manera transaccional).**
- **Los endpoints deben estar totalmente parametrizados según el contexto de los mismos.**

### **ADEMÁS:**

- Cada operación anterior debe usarse con el mensaje HTTP adecuado (GET, POST, PUT, DELETE, ...).
- La API tiene que tener suficientes datos iniciales para poder probar la paginación.
- La API debe estar testeada en PostMan en todos sus endpoints.
- La entrega consistirá en el código de la API junto con el fichero de exportación de los tests de Postman.
- Deberá exponerse en clase en una presentación que durará entre 5 y 10 minutos.

**IMPORTANTE:** Estos son unos objetivos básicos; cada uno puede añadir lo que considere oportuno y se tendrá en cuenta para la nota de la actividad que las consultas diseñadas y los datos almacenados tengan algún sentido práctico.