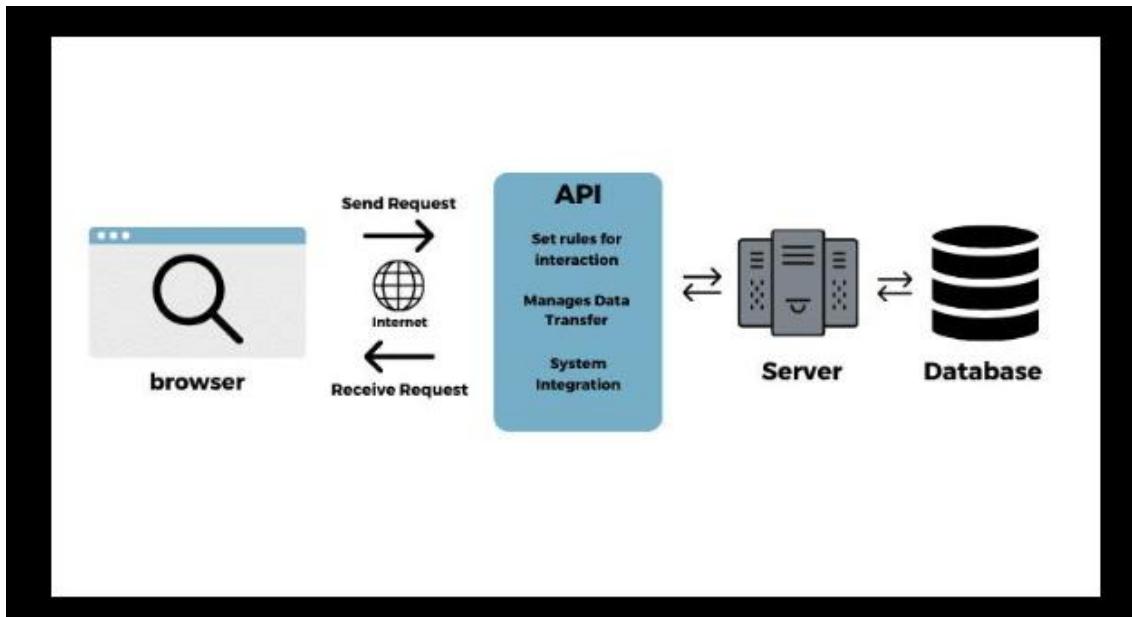


El objetivo de esta práctica consiste en elaborar fullstack un proyecto completo que incluya Angular como frontend (cliente), jpa con mariadb como backend (servidor) así como una API que consuma los métodos del servicio. Por lo tanto, englobaremos en la misma práctica las asignaturas de Cliente, Servidor Java y Módulo Optativo.



¿Por dónde empezamos? “*Nada visual sin datos, nada de datos sin modelo, nada de modelo sin entender el problema.*”

FASE 0. Planteamiento del proyecto: ¿Qué voy a hacer? Pensar en el dominio del problema y casos de uso principales.

FASE 1. Modelo de datos

- Servidor: Definir entidades y realizar el conjunto de operaciones básicas CRUD, relaciones 1-1, 1-N, N-M.

FASE 2. API REST

- OPTATIVA:
 - Endpoints básicos (CRUD). Probarlos con Postman. Debe de tener Swagger.
 - Uso de códigos HTTP correctos (200, 400, etc...)
 - Uso de ResponseEntity

FASE 3. Frontend. ANGULAR

- CLIENTE:
 - Crear componentes, rutas... y consumir la API REST con HttpCliente.
 - Usar modelo para los datos

FASE 4. COMPLEMENTAR CON FUNCIONALIDADES EN BACKEND y CLIENTE

No dejar el proyecto en solo “CRUD”. Añadir métodos en el servicio y endpoints en la API para poder ser accedidos desde cliente.

La diferencia estará en la variedad y complejidad no en hacer más de lo mismo.

EVALUACIÓN Y DEFENSA

La práctica se evaluará en clase, y la calificación dependerá del nivel de complejidad utilizado siguiendo las siguientes indicaciones:

Todos los niveles deben tener el CRUD completo.

Las entidades deben tener como mínimo 5 atributos.

Nivel	JPA / Base de datos	Backend / API REST	Frontend / Angular
Aprobado (Al menos...)	- 3 entidades mínimo - Debe aparecer 1 relación de cada tipo como mínimo (1-1, 1-N, N-M)	- Todo el CRUD - 5 métodos adicionales	- Rutas con 4 opciones - 4 componentes - 4 métodos servicios - 2 componentes que pasen información del hijo al padre y/o viceversa
Notable (Al menos...)	- 5 entidades mínimo - Debe aparecer 1 relación de cada tipo como mínimo (1-1, 1-N, N-M)	- Todo el CRUD - 7 métodos adicionales	- Rutas con 6 opciones - 6 componentes - 6 métodos servicios - 3 componentes que pasen información del hijo al padre y/o viceversa
Sobresaliente (Al menos...)	- 7 entidades mínimo - Debe aparecer 1 relación de cada tipo como mínimo (1-1, 1-N, N-M)	- Todo el CRUD - 9 métodos adicionales	- Rutas con 7 opciones - 7 componentes - 7 métodos servicios - 4 componentes que pasen información del hijo al padre y/o viceversa

Nota: La tabla intermedia generada automáticamente por JPA en una relación **N-M no se contabiliza como entidad** para el mínimo exigido.

Solo se contará como entidad si se modela explícitamente con atributos propios y CRUD independiente