

ARQUITECTURA DEL **SERVICIO REST**

| URL | Método HTTP | Descripción | Código de respuesta correcta |
|--------------------|-------------|---|------------------------------|
| /api/empleados | GET | Obtener la lista de todos los empleados | 200 OK |
| /api/empleado/{id} | GET | Obtener los datos del empleado {id} | 200 OK |
| /api/empleado | POST | Crear un nuevo empleado | 201 Created |
| /api/empleado/{id} | PUT | Modificar un empleado existente | 200 OK |
| /api/empleado/{id} | DELETE | Eliminar un empleado existente | 200 OK |



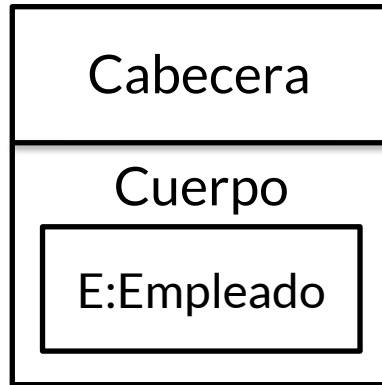
@RequestBody

- Inyectamos en un argumento del controlador el cuerpo de una petición

```
@PostMapping("/empleado")
public Empleado createEmpleado(@RequestBody Empleado empleado, HttpServletResponse response) {
    Empleado nuevo = new Empleado(empleado.getNombre(), empleado.getApellidos(), empleado.getFechaNacimiento());
    response.setStatus(201);
    return empleadoRepository.save(nuevo);
}
```

RequestEntity<?>

- ▶ Nos permite acceder a la petición completa (cabeceras y cuerpo)
- ▶ Más versátil



FORMAS DE DEVOLVER **RESPONSE CODE**

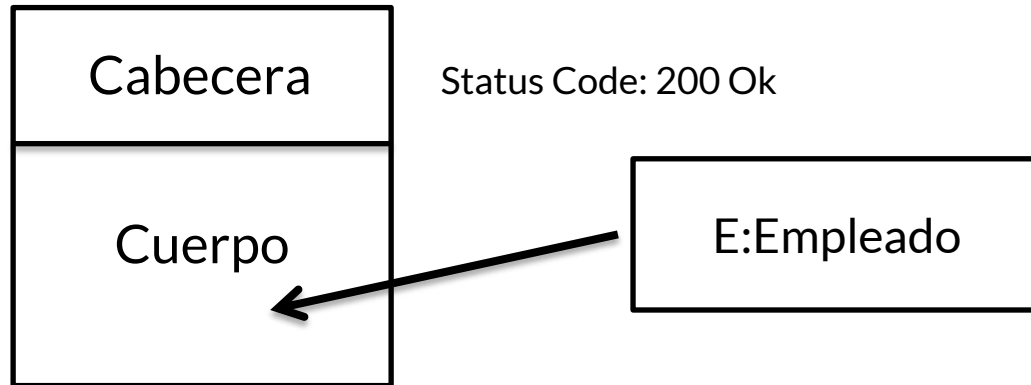
- ▶ HttpServletResponse
- ▶ ResponseEntity<?>

Posibles *respuestas*:

- ▶ 201 (Creado)
- ▶ 400 (Error en la petición)
- ▶ 409 (Conflicto)

ResponseEntity<?>

- ▶ Clase que encapsula una respuesta HTTP (con cabeceras y cuerpo).
- ▶ En el cuerpo podemos enviar instancias de nuestro modelo.



ResponseEntity<?>

- ▶ Nos permite enviar otros códigos de respuesta diferentes a Ok

