

CODELAB 4

En este caso, **OpenFeign** cumple las siguientes funciones clave:

1. Consulta a Eureka para obtener la dirección del producto-service

OpenFeign se integra con Eureka (el servidor de descubrimiento) para encontrar la instancia correcta del servicio destino (por ejemplo, producto-service).

Esto permite que no tengas que codificar manualmente la URL del servicio, ya que se resuelve automáticamente a través del nombre del servicio registrado.

2. Abstrae las llamadas HTTP REST

- No necesitas escribir manualmente código para RestTemplate o WebClient.
- Simplemente defines una interfaz Java con anotaciones (@FeignClient, @GetMapping, etc.) y OpenFeign genera la implementación automáticamente.

3. Serializa y deserializa objetos

- Convierte automáticamente los objetos Java a JSON (o al formato requerido) para enviarlos por HTTP.
- Y también convierte las respuestas JSON en objetos Java al recibirlos.

4. Maneja la conexión HTTP

- Se encarga de abrir, mantener y cerrar las conexiones HTTP.
- También puede gestionar timeouts, retries, balanceo de carga

Project

- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- producto-service
 - .mvn
 - src
 - main
 - java
 - com.productoservice.producto_service
 - Producto
 - ProductoController
 - ProductoRepository
 - ProductoServiceApplication
 - RequestProducto
- resources

```
1 package com.productoservice.producto_service;
2
3 import ...
4
5
6
7 @RestController
8 @RequestMapping("/api/productos")
9 public class ProductoController {
10
11     private final ProductoRepository repository; 7 usages
12
13     @Autowired
14     public ProductoController(ProductoRepository repository) { this.repository = repository; }
15
16
17
18     @GetMapping
19     public List<Producto> obtenerTodos() {
20         List<Producto> productos = repository.findAll();
21         return productos;
22     }
23 }
```

Services

Project

- CodeLabs_Sincronico C:\Users\JSBARONA\IdeaProjects\CodeLabs_Sincronico
 - idea
 - eureka-server
 - .mvn
 - src
 - main
 - java
 - com.eurekaserver.eureka_server
 - EurekaServerApplication
 - resources
 - test
 - target
 - .gitattributes
 - .gitignore
 - mvnw
 - mvnw.cmd

```
1 package com.eurekaserver.eureka_server;
2
3 import ...
4
5
6
7 @SpringBootApplication
8 @EnableEurekaServer
9 public class EurekaServerApplication {
10
11     public static void main(String[] args) { SpringApplication.run(EurekaServerApplication.class, args); }
12
13
14
15
16 }
```

Project

- CodeLabs_Sincronico C:\Users\JSBARONA\IdeaProjects\CodeLabs_Sincronico
 - idea
 - eureka-server
 - orden-service
 - .mvn
 - src
 - main
 - java
 - com.example.ordenservice
 - Orden
 - OrdenController
 - OrdenRepository
 - OrdenServiceApplication
 - ProductoClient
 - ProductoModel
 - RequestOrden
 - resources

```
1 package com.example.ordenservice;
2
3 import ...
4
5
6
7 @RestController
8 @RequestMapping("/api/orden")
9 public class OrdenController {
10
11     @Autowired
12     private final OrdenRepository repository;
13     private final ProductoClient productoClient; 2 usages
14
15     @Autowired
16     public OrdenController(OrdenRepository repository, ProductoClient productoClient) {
17         this.repository = repository;
18     }
19 }
```

Spring Boot

- EurekaServerApplication :8761/
- OrdenServiceApplication :8083/
- ProductoServiceApplication :8081/

Consulta

Historial de Consultas

Scratch Pad x

1 SELECT * FROM public.orden

2 ORDER BY id_orden ASC

Data Output

Mensajes

Notificaciones

	id_orden [PK] integer	estado character varying (255)	fecha_creacion date	id_usuario integer	total double precision
1	1	entregada	2025-05-12	2	4500000