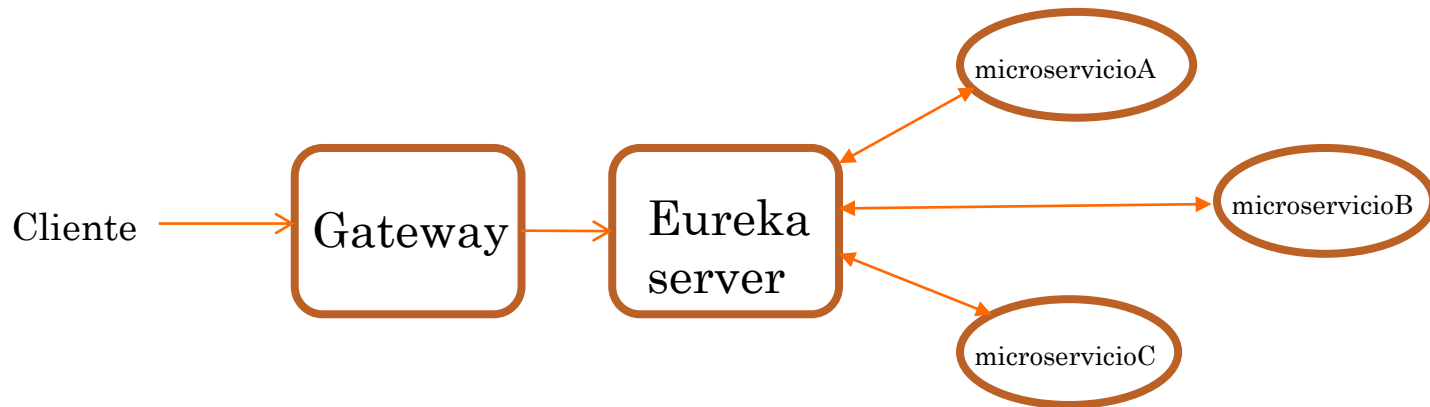




SPRING CLOUD GATEWAY

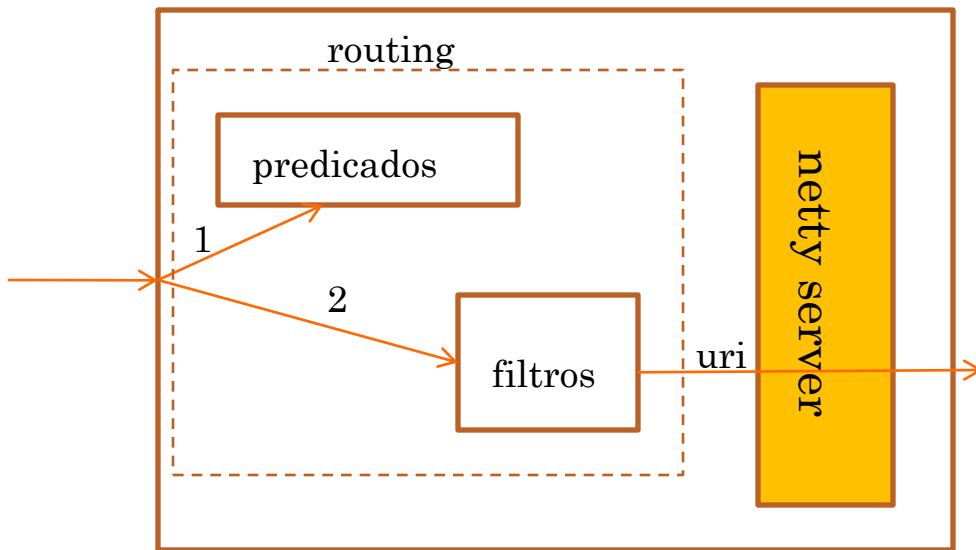
Características

- Servidor que proporciona un punto de acceso único a un conjunto de servicios.
- Utilizado desde Spring 2.4, antes Zuul Server.
- Acceso a los servicios a través de Eureka.



Arquitectura

Spring Cloud Gateway



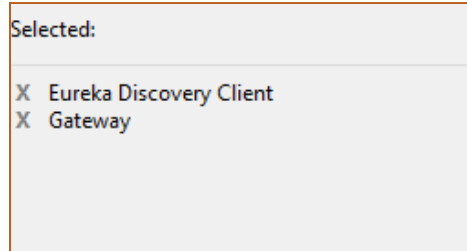
➤ A través de los *enrutados* (routing) se define lo que se debe hacer con una petición específica.

➤ El *predicado* establece la condición (path, cookies, encabezados,...), que debe cumplir una petición para ser enrutada a un determinado destino.

➤ Antes de dirigir la petición al destino, se le pueden aplicar uno o varios *filtros* que modifiquen alguna característica de la petición.

Creación del servidor Gateway

- Se crea como microservicio Spring Boot
- Requiere las dependencias Gateway y Discovery Client, no se debe incluir Spring Web:



- La clase main no tiene que tener ninguna anotación especial



Configuración

- A través de `spring.cloud.gateway.routes`, se indican los mapeos de las peticiones a los servicios.
- Cada ruta incluye un identificador, dirección del servicio (uri) predicados y posibles filtros a aplicar:

```
...
spring:
  cloud:
    gateway:
      routes:
        - id: serv-libros
          #lb:// cuando se va a acceder a través de Eureka
          uri: lb://servicio-libros#http://localhost:8080
          predicates:
            - Path=/slibros/**
          filters:
            - RewritePath=/slibros/*, /
...

```



Configuración CORS

➤ Para evitar bloque de peticiones desde script, se debe incluir una configuración especial:

```
...
spring:
  cloud:
    gateway:
      default-filters:
        - DedupeResponseHeader=Access-Control-Allow-Credentials Access-Control-Allow-Origin
      globalcors:
        corsConfigurations:
          '[/**]':
            allowedOrigins: '*'
            allowedMethods: '*'
            allowedHeaders: '*'
        routes:
          ...
...

```



Utilización Gateway

Cliente

