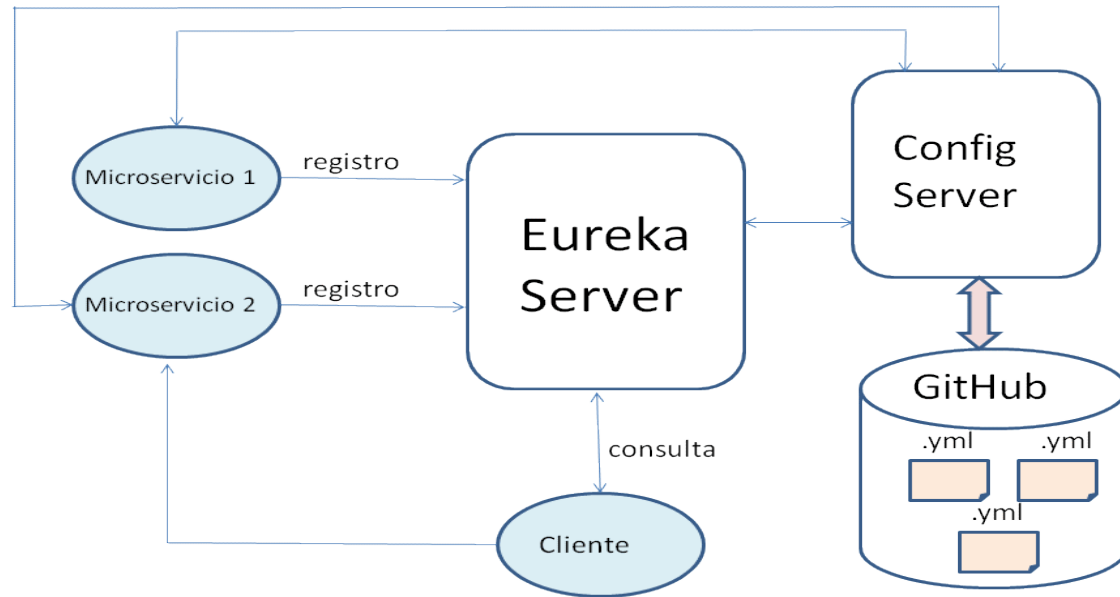


Servidor de configuración

Spring Cloud Config

- Centraliza la información de configuración de un conjunto de servicios dentro de un repositorio GitHub



Creación del servidor

➤ Se crea como microservicio Spring Boot

➤ Requiere la dependencia Config Server:

```
<dependency>  
    <groupId>org.springframework.cloud</groupId>  
    <artifactId>spring-cloud-config-server</artifactId>  
</dependency>
```

➤ Clase main anotada con @EnableConfigServer

```
@EnableConfigServer //lo habilita como servidor de configuración  
@SpringBootApplication  
public class Application {  
    public static void main(String[] args) {  
        SpringApplication.run(Application.class, args);  
    }  
}
```

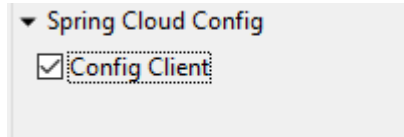
Configuración del servidor

- Mediante la propiedad `spring.cloud.config` se establece la localización del repositorio

```
server:  
  port: 8888  
spring:  
  cloud:  
    config:  
      server:  
        git:  
          uri: https://github.com/amartinsierra/spring-config
```

Configuración de los microservicios

- Los archivos de configuración de los servicios se llevan al repositorio de GitHub, nombrándolos como : nombre-servicio.yml
- Se incluirá el starter Config Client:



- En cada servicio, se incluye un archivo de configuración bootstrap.yml :

```
spring:
  application:
    name: servicio-conf # Nombre del archivo de configuración en github
  cloud:
    config:
      uri: http://localhost:8888 #Localización Spring Cloud Config
```

Versión Spring Boot 2.4

- A partir de la versión 2.4, el bootstrapping no está activado por defecto en los microservicios. O lo que es lo mismo, no se reconoce el archivo bootstrap.yml
- Para poder reconocerlo, habrá que incluir un nuevo starter en los microservicios:

```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-starter-bootstrap</artifactId>  
</dependency>
```

Versión Spring Boot 3.X

- Desde la versión Spring Boot 3.0 el acceso a un servidor de configuración se ha simplificado bastante, ya no es necesario utilizar bootstrapping
- En el propio archivo `application.yml` del microservicio se indica la dirección del servidor mediante:

```
spring:  
  application:  
    name: servicio-conf    # Nombre microservicio  
  config:  
    import: optional:configserver: http://localhost:8888 #Localización Spring Cloud Config
```