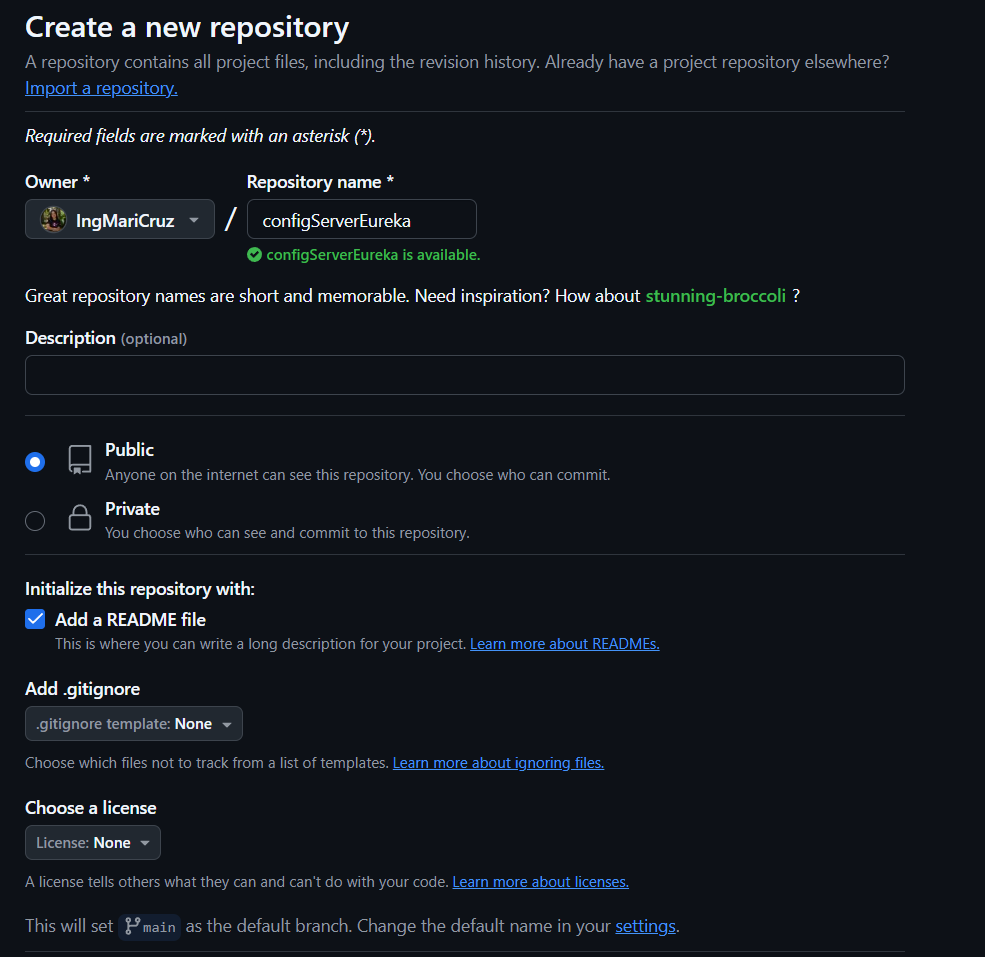
**GUIA PARA CREAR CONFIG SERVER**

**Mariana Cruz**

# Crear el repositorio en github

* 1. Crea un repositorio para el configServer



* 1. Abre add file – create new file

Captura de pantalla de un celular

Descripción generada automáticamente

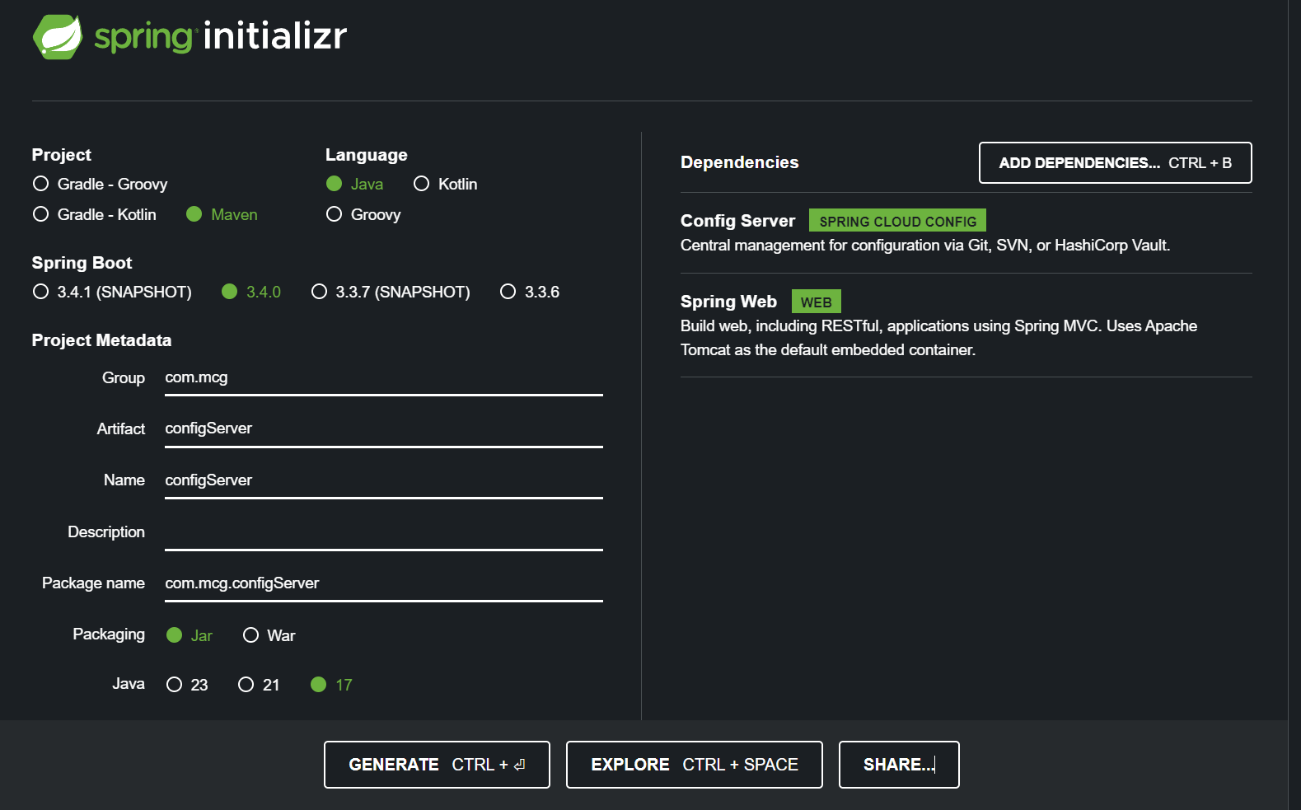
* 1. Vamos a crear 3 archivos:
* Eureka.properties
* Usuario.properties
* curso.properties

Captura de pantalla de un celular

Descripción generada automáticamente

# Crear el config server

* 1. Crea un nuevo proyecto de Spring Boot usando Maven. Puedes hacerlo desde [Spring Initializr](https://start.spring.io/):
* **Project**: Maven
* **Language**: Java
* **Spring Boot Version**: Escoge la más reciente.
* **Group**: com.{nombre}
* **Artifact**: {nombre}
* **Dependencies**: Config Server, Spring Web.



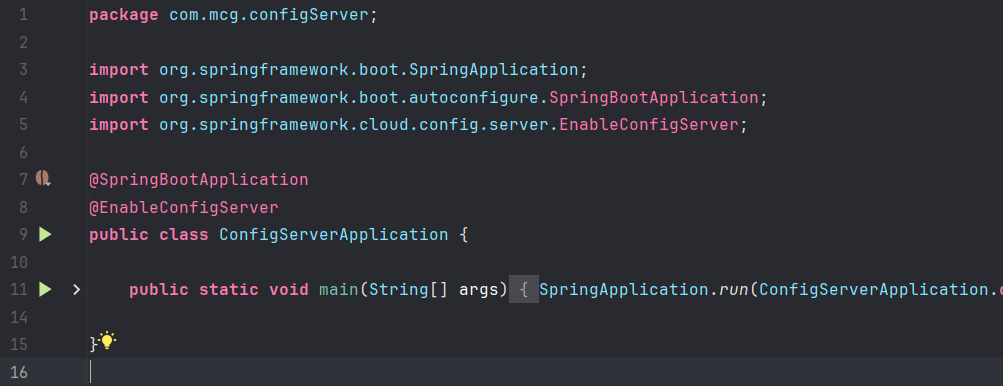
* 1. Abre el proyecto en tu entorno de trabajo
  2. Debes tener un pom así:

1. <?xml version="1.0" encoding="UTF-8"?>  
   <**project** xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
    <**modelVersion**>4.0.0</**modelVersion**>  
    <**parent**>  
    <**groupId**>org.springframework.boot</**groupId**>  
    <**artifactId**>spring-boot-starter-parent</**artifactId**>  
    <**version**>3.3.5</**version**>  
    <**relativePath**/> <!-- lookup parent from repository -->  
    </**parent**>  
    <**groupId**>com.mcg</**groupId**>  
    <**artifactId**>configServer</**artifactId**>  
    <**version**>0.0.1-SNAPSHOT</**version**>  
    <**name**>configServer</**name**>  
    <**description**>Demo project for Spring Boot</**description**>  
    <**url**/>  
    <**licenses**>  
    <**license**/>  
    </**licenses**>  
    <**developers**>  
    <**developer**/>  
    </**developers**>  
    <**scm**>  
    <**connection**/>  
    <**developerConnection**/>  
    <**tag**/>  
    <**url**/>  
    </**scm**>  
    <**properties**>  
    <**java.version**>17</**java.version**>  
    <**spring-cloud.version**>2022.0.4</**spring-cloud.version**>  
    </**properties**>  
     
    <**dependencies**>  
    <**dependency**>  
    <**groupId**>org.springframework.cloud</**groupId**>  
    <**artifactId**>spring-cloud-config-server</**artifactId**>  
    </**dependency**>  
     
    <**dependency**>  
    <**groupId**>org.springframework.boot</**groupId**>  
    <**artifactId**>spring-boot-starter-test</**artifactId**>  
    <**scope**>test</**scope**>  
    </**dependency**>  
     
    <**dependency**>  
    <**groupId**>org.springframework.boot</**groupId**>  
    <**artifactId**>spring-boot-starter-actuator</**artifactId**>  
    </**dependency**>  
     
    </**dependencies**>  
    <**dependencyManagement**>  
    <**dependencies**>  
    <**dependency**>  
    <**groupId**>org.springframework.cloud</**groupId**>  
    <**artifactId**>spring-cloud-dependencies</**artifactId**>  
    <**version**>${spring-cloud.version}</**version**>  
    <**type**>pom</**type**>  
    <**scope**>import</**scope**>  
    </**dependency**>  
    </**dependencies**>  
    </**dependencyManagement**>  
     
    <**build**>  
    <**plugins**>  
    <**plugin**>  
    <**groupId**>org.springframework.boot</**groupId**>  
    <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
    </**plugin**>  
    </**plugins**>  
    </**build**>  
     
   </**project**>
   1. Abre el archivo src/main/resources/application.properties
   2. configura las siguientes propiedades para conectar el Config Server con GitHub:



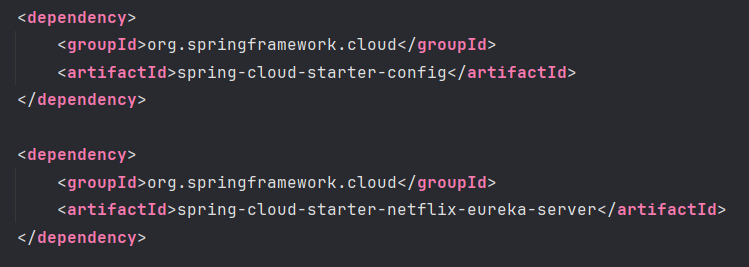
Recuerda agregar tu url del repositorio que creamos anteriormente

* 1. Habilita la clase principal

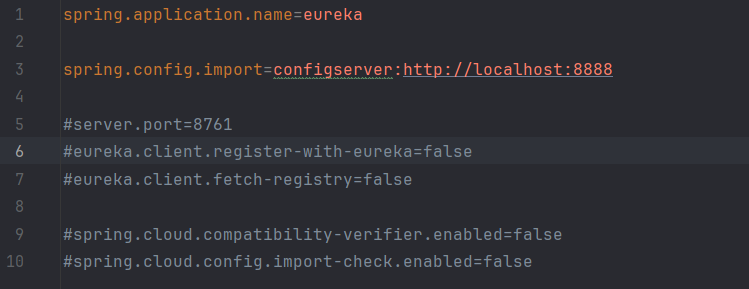


# Configurar Eureka

* 1. Verifica las siguientes dependencias



* 1. Abre el archivo src/main/resources/application.properties
  2. Configura el application.properties



* 1. Agrega las configuraciones del servicio en el properties del github

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

* 1. Asegúrate de añadir la anotación @EnableEurekaServer en la clase principal de Eureka:

Texto

Descripción generada automáticamente

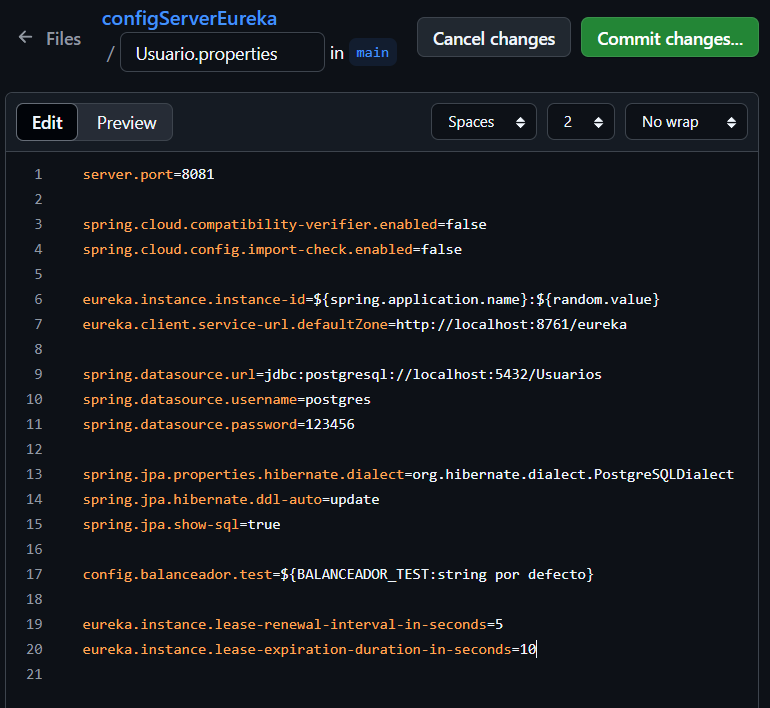
# Configurar Microservicio Usuario

* 1. Verifica las siguientes dependencias

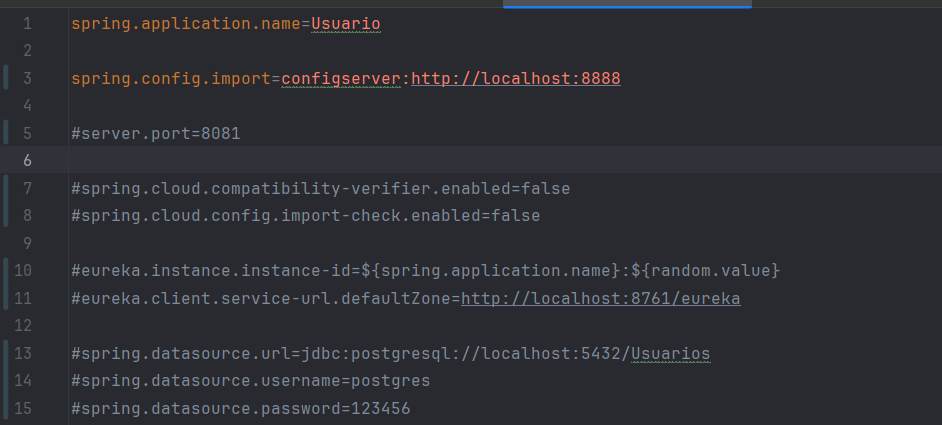
Texto

Descripción generada automáticamente

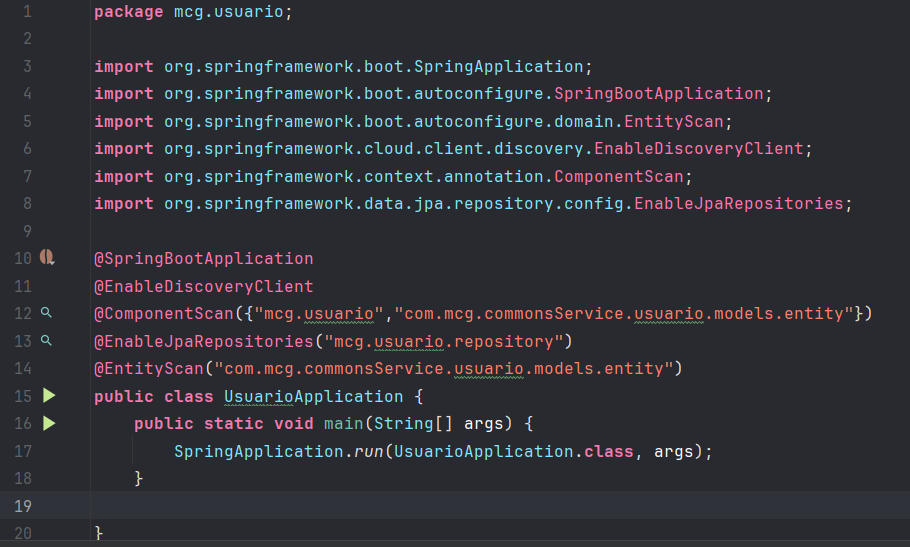
* 1. Agrega las configuraciones del servicio en el properties del github



* 1. Abre el archivo src/main/resources/application.properties
  2. Configura el application.properties:



* 1. Agrega la anotación @EnableDiscoveryClient en la clase principal



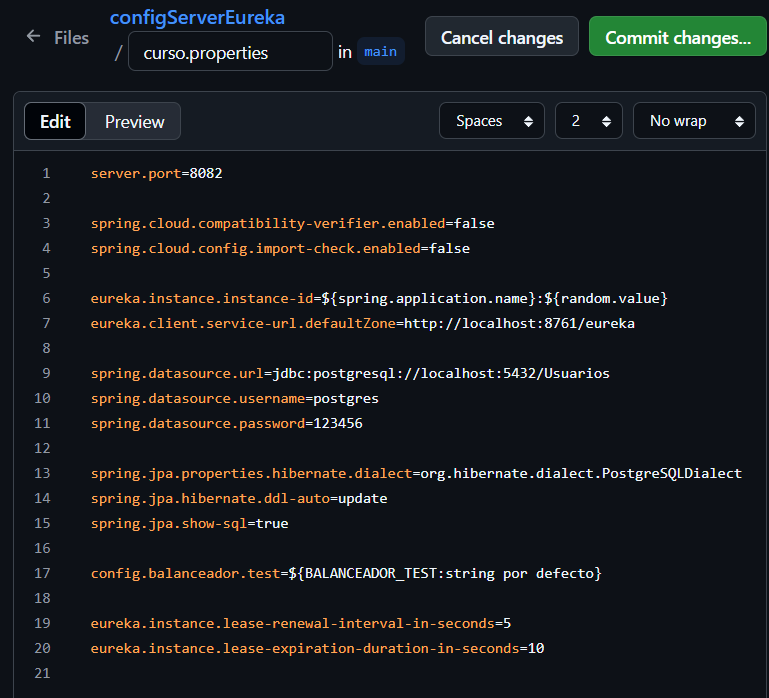
# Configurar Microservicio Curso

* 1. Verifica las siguientes dependencias

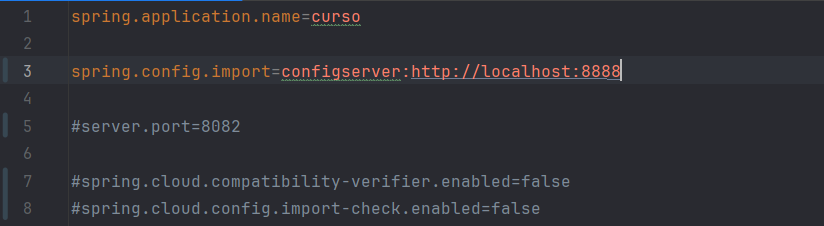
Texto

Descripción generada automáticamente

* 1. Agrega las configuraciones del servicio en el properties del github



* 1. Configura el application.properties:



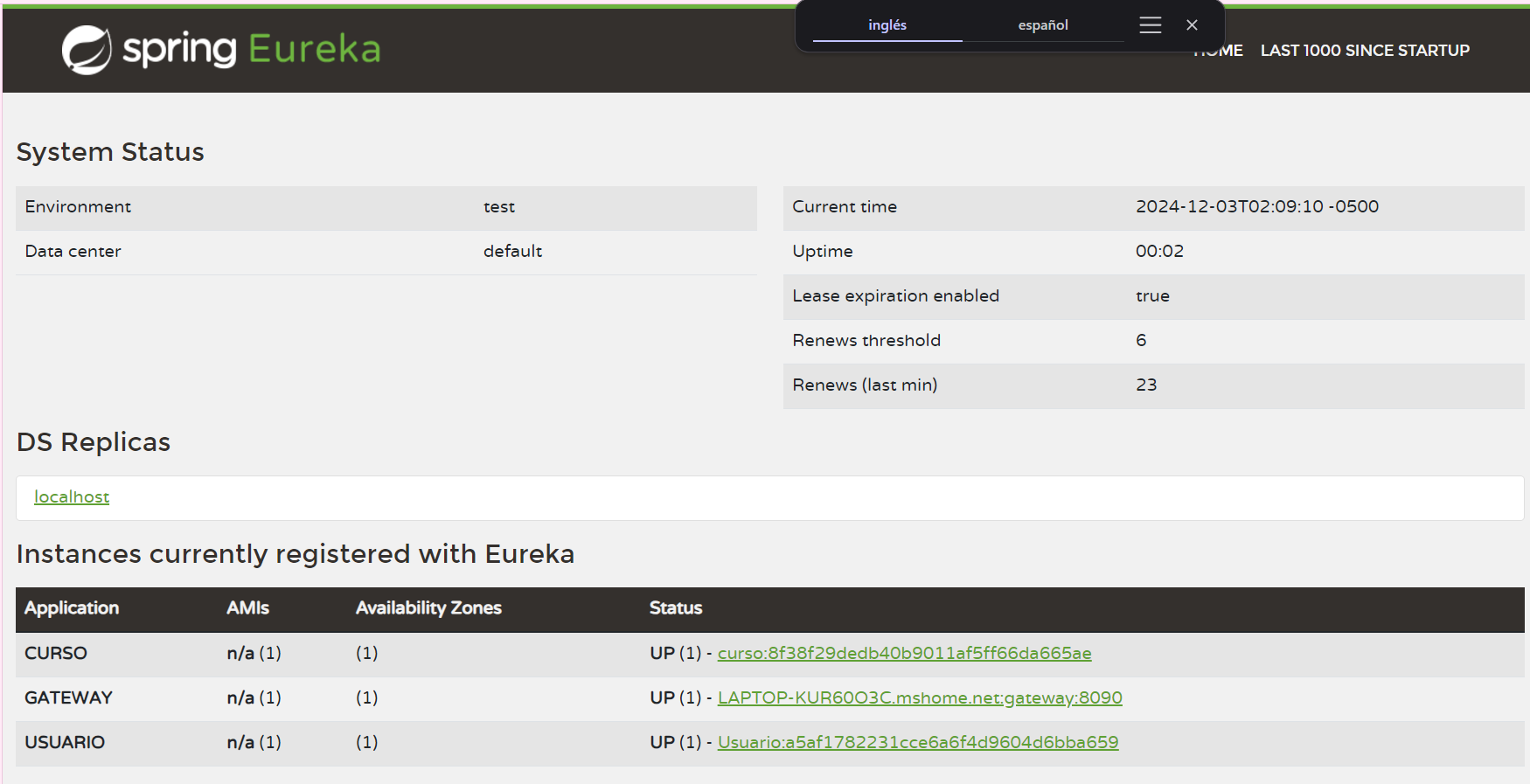
* 1. Agrega la anotación @EnableDiscoveryClient en la clase principal

Texto

Descripción generada automáticamente

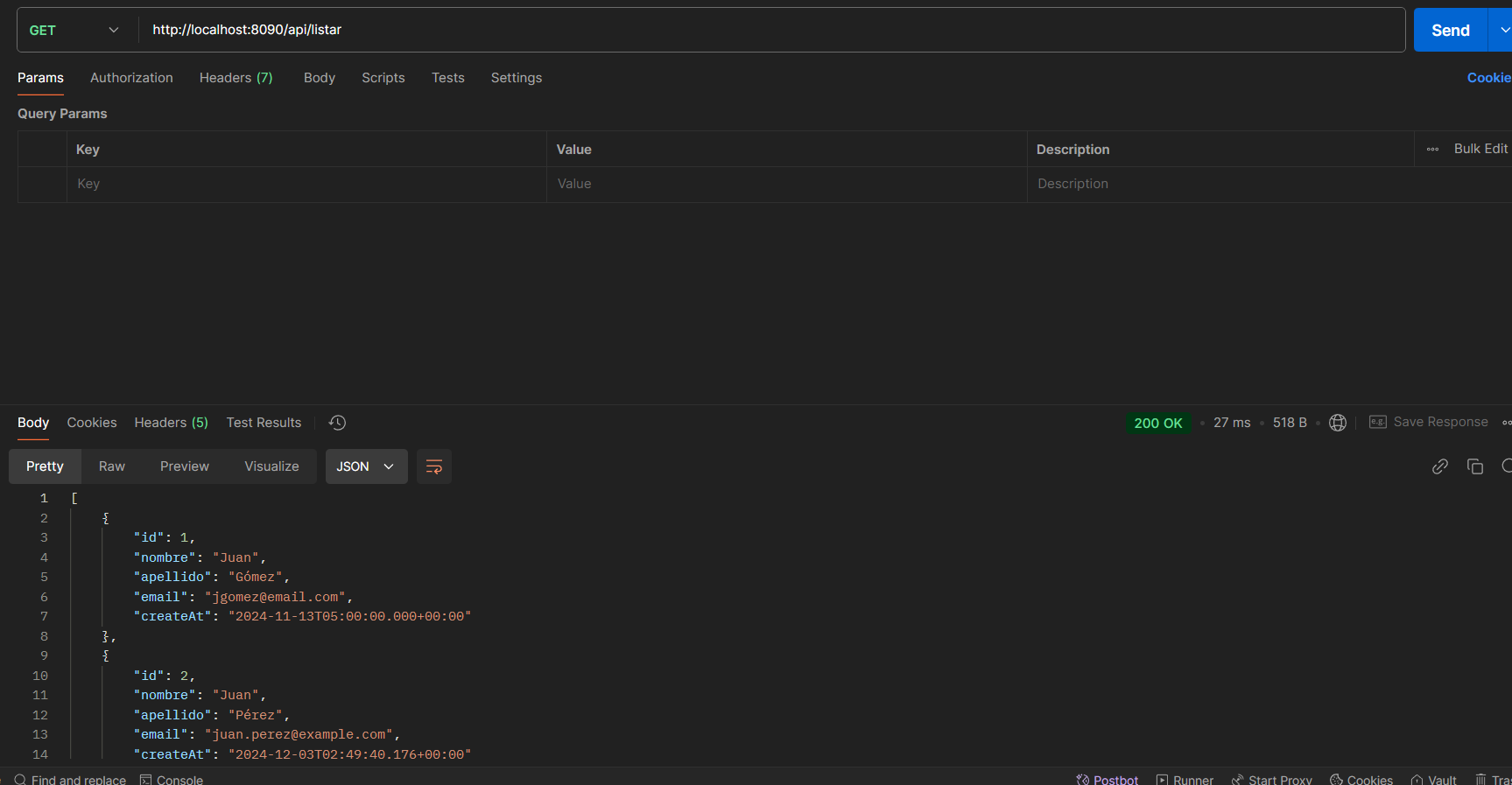
# Iniciar los microservicios

* 1. La secuencia es:
* ConfigServer
* Eureka
* Usuario
* Curso
* Gateway



# Verificar la configuración

* 1. Para Usuario:



* 1. Para Curso:

Captura de pantalla de computadora

Descripción generada automáticamente

# Conclusión

Has configurado con éxito un **Config Server** que centraliza las configuraciones de tus microservicios y los conecta con un repositorio en **GitHub**. Todos los microservicios, incluidos **Eureka**, **Usuario**, y **Curso**, están ahora configurados para obtener su configuración desde el Config Server y están registrados en **Eureka** para facilitar su descubrimiento.