1.- Crear los microservicios.

2.- Configurar Ribbon para el balanceo de carga de los microservicios.

@RibbonClient(name = "service-name")

3.- Crear el proyecto de *Eureka Server* y configurar,

4.- A los microservicios inyectar la dependencia de *Eureka Dicovery Client* para que se registre en el Eureka server.

@EnableEurekaClient Clase principal

# Especificar el servidor al que que secmunicaran los clientes de eureka

eureka.client.service-url.defaultZone=localhost:8761/eureka

Nota: Si las aplicaciones tienen la dependencia ribbon, se debe remover porque está implícita en eureka Discovery client.

5.- configurar el microservicio para puerto dinámico, para evitar configurar los puertos cuando se levante.

6.- agregar la dependencia de Histryx para tolerancia a fallos, en los microservicios.

Clase principal @EnableCircuitBreaker

En el método donde se va a tener el control de excepciones @HystrixCommand(fallbackMethod = "detalleException")

Configuración de properties

# Configuracion de timeout

hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds=13000

ribbon.ConnectTimeout=3000

ribbon.ReadTimeout=10000

7.- crear y configurar zuul

Clase principal @EnableEurekaClient @EnableZuulProxy

Configuración timeout

# Configuracion de timeout

hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds=13000

ribbon.ConnectTimeout=3000

ribbon.ReadTimeout=10000

8.- cre ar proyecto config-server proyecto que funcionara cono servidor de configuración,

Clase principal @EnableConfigServer

Crear el repositori git y configurar el proyectos

9.- agregar las dependencia de Config cliente al los microservicios.

10.- refhesh actuamtico para configuraciones de git

En el controlador @RefreshScope , agregar la dependencia de actuator

11 crear proyecto Commons para pojos

Remover método main.

Pom.xml remover etiqueta plugin.