

TECNOLOGÍAS DEL SECTOR FINANCIERO

Práctica 12: Cloud computing

Andoni Alcelay

MÁSTER UNIVERSITARIO EN TECNOLOGÍAS DEL SECTOR FINANCIERO: FINTECH

# Introducción

En este documento se explicará el proceso seguido para la realización de la práctica 12, cloud computing. Este proyecto consiste en realizar varias actividades con microservicios y eureka para unirlos todos se usan otros servicios como Feing y Zuul para intercomunicar mejor las aplicaciones entre ellas. El objetivo es crear una aplicación similar a Netflix. Antes de entrar en detalle, cabe destacar que la versión de spring-boot que se dice durante todo el documento no está disponible a día de hoy así que ha habido grandes problemas para la realización de esta práctica, ya que la mayoría de las dependencias no están disponibles.

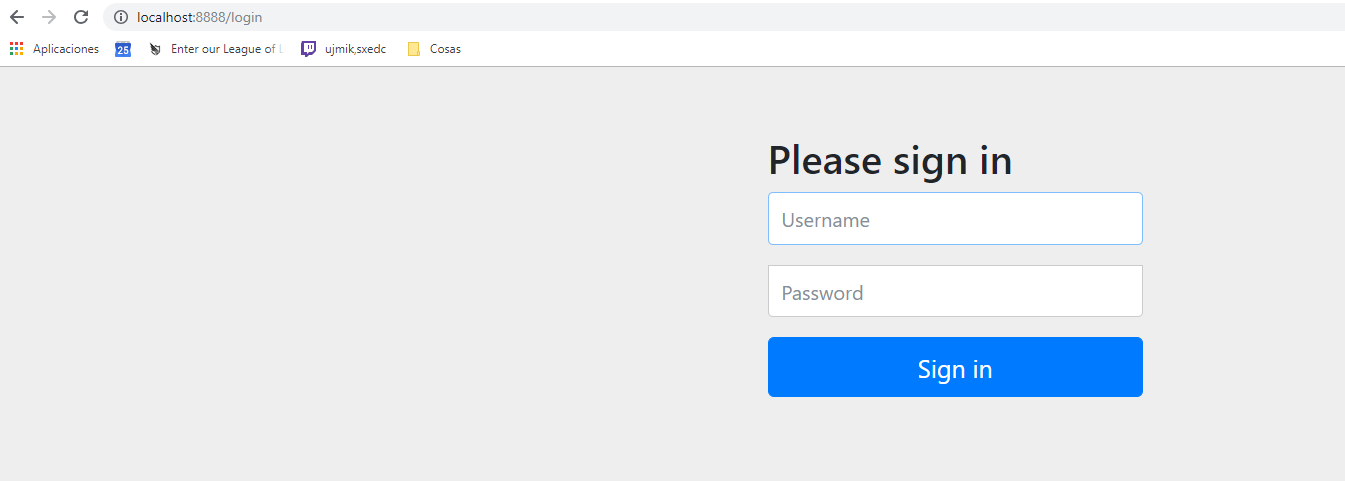
# Primer microservicio con spring boot

Este Proyecto consiste en crear un microservicio con spring boot que devolvía en web un resultado de “Hello world”, una vez ejecutado el servicio el resultado ha sido este:



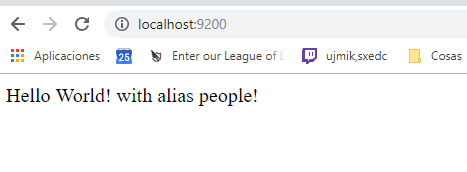
# Servicio de configuraciones

A continuación, se crea un servicio config server que estará securizado con un usuario y contraseña definido previamente, este proyecto se lanzará en el puerto 8888 y el resultado de su ejecución es el siguiente:



# Config Server y Microservicio con configuración

Una vez creadas los dos proyectos se procede a unir el uno con el otro. Para esto se han añadido unas variables al microservicio que serán añadidas desde el config server. Para eso se ha cambiado el valor del text que se muestra en el microservicio siendo este:



# Descubridor de servicios Eureka

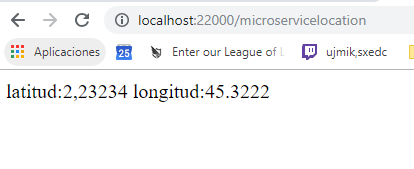
Después de haber unidos los dos proyectos se añadirá Eureka para poder localizarlos todos. Para ello se ha creado un tercer microservicio similar al de ejemplo que mostrará coordenadas. Estos microservicios y el config server, se conectan al servidor eureka para que este pueda localizarlos tal y como se ve en la imagen.

# 

# Enrutado

El enrutado se hace a través de un proyecto basado en Zuul. Esto hace que los dos servicios se enruten con el nombre deseado y se mapean en el puerto 22000, el resultado ha sido el siguiente para cada uno de los dos servicios.

# 



# Balanceo de carga y comunicación entre servicios y Alta disponibilidad entre servicios

Estos dos últimos no han podido desarrollarse porque las dependencias que se muestran en la práctica no concuerdan con los de la versión de spring boot que se ha podido crear con el generador de proyectos spring. Si se intentaba bajar la versión daba problemas con los tests creados por la herramienta, aun así se ha conseguido que el proyecto compile buscando todo con herramientas externas pero el autowired de las variables no ha conseguido que funcione.