

Práctica 1. Microservicios

Enunciado

El objetivo de esta práctica consiste en realizar una arquitectura basada en microservicios usando el framework Eventuate.

Se pretende implementar un sistema de préstamos de libros en una biblioteca con las siguientes funcionalidades:

- Un préstamo se asocia a un usuario y a un libro.
- Cada usuario tiene asociado un número de préstamos simultáneos (que se reduce cuando se hace un préstamo)
- Cada libro tiene un número de ejemplares disponibles (que se reduce cuando se hace un préstamo)
- Cuando se hace un préstamo, sólo se crea de forma satisfactoria si el usuario puede tener préstamos disponibles y hay ejemplares del libro
- La aplicación debe ofrecer una API REST de los 3 recursos: préstamos, usuario y libro

La aplicación debe implementarse con algunas de las siguientes tecnologías:

- Aplicación implementada con Eventuate Tram:
<https://github.com/eventuate-tram/eventuate-tram-sagas-examples-customers-and-orders>
- Aplicación implementada con colas con Kafka y Máquina de estados:
<https://github.com/MasterCloudApps/3.4.Aplicaciones-nativas-de-la-nube/tree/master/microservicios/sagas-state-machine>

La implementación se realizará con las siguientes consideraciones (para obtener un 7.5):

- La implementación seguirá una arquitectura de microservicios
- Cada recurso será gestionado por un microservicio diferente (libros, préstamos y usuarios)
- Se escribirá tests e2e por cada recurso de la API REST que verifique que las operaciones de la API REST funcionan correctamente.
- La saga de creación del préstamo tiene las siguientes características:
 - Se reducirá el número de préstamos disponibles del usuario primero.
 - Si no hay ejemplares del libro disponibles se vuelve a aumentar el número de préstamos (como una transacción de compensación)
- Se debe implementar un proxy para que todas las operaciones REST se puedan hacer al mismo endpoint http (por ejemplo NGINX)
- Se usará docker-compose para el despliegue de los servicios, las bases de datos y el proxy.

Implementación opcional (para obtener un 10):

- Se deberá implementar un API Gateway usando Spring API Gateway que permita obtener, con una única petición, información del usuario y del libro cuando se hace un GET de un préstamo con la query <http://server/borrows?fullinfo=true>
- Además se incluirá en el API Gateway el enrutado de los servicios para préstamos, usuarios y libros

Formato de entrega

La práctica se entregará teniendo en cuenta los siguientes aspectos:

- La práctica se entregará como un fichero .zip con el código de los diferentes servicios y los ficheros auxiliares. El nombre del fichero .zip será el correo URJC del alumno (sin @alumnos.urjc.es).

Las prácticas se podrán realizar de forma individual o por parejas. En caso de que la práctica se haga por parejas:

- Sólo será entregada por uno de los alumnos
- El nombre del fichero .zip contendrá el correo de ambos alumnos separado por guión. Por ejemplo p.perezf2019-z.gonzalez2019.zip