

Práctica 1. Microservicios

Enunciado

El objetivo de esta práctica consiste en ampliar una saga implementada en una aplicación existente. Habría que añadir un nuevo servicio a la aplicación y añadir una nueva transacción a la saga.

La aplicación base puede elegirse entre dos opciones:

- 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>

Fase 1: Nuevo microservicio (8 puntos):

Añadir un nuevo microservicio a la aplicación:

- Eventuate Tram: Añadir un servicio de productos que gestione la descripción del productos y su stock.
- Kafka y Máquina de estados: Añadir un servicio de Delivery en el que se verifique si hay capacidad de delivery del producto para una ciudad concreta y se genere un “delivery-id”. Añadir al pedido una ciudad de entrega. El servicio de delivery deberá generar internamente una ruta (que será simulada con un String random) que se guardará en la BBDD.

Será necesario implementar la API REST para este nuevo servicio:

- Productos: Creación y consulta de productos con stock.
- Delivery: Crear ciudades con número de envíos. Consulta de la ruta de los envíos.

Se deberá entregar un proyecto con tests de API REST funcionales en los que se verifique que funcionan todas las situaciones:

- Eventuate Tram:
 - Pedido correcto.
 - Pedido rechazado porque falla el saldo del cliente
 - Pedido rechazado porque falla el stock de producto
- Kafka + State Machine:
 - Pedido correcto.
 - Pedido rechazado porque falla el saldo del cliente
 - Pedido rechazado porque falla el stock del producto
 - Pedido rechazado porque no hay disponibilidad de delivery a esa ciudad.

Estos tests podrán acceder a los servicios directamente.

Fase 2: Ampliar el API Gateway (2 puntos):

- Se deberá incluir un API Gateway basado en Spring Cloud Gateway que permita obtener, con una única petición, información del pedido e información del producto en un pedido concreto. Para el caso del delivery, también deberá llevar información de la ruta.
- Además se incluirá en el API Gateway en enrutado de los servicios individuales.

Se deberá incluir un test adicional que verifique que cuando un pedido se crea correctamente, se puede obtener toda su información en una única petición.

Se deberá entregar una documentación en el README que muestre los pasos a seguir para levantar los microservicios y ejecutar los tests.

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