2018/05/17

# Consigna

La prueba consistirá en el desarrollo de 2 aplicaciones, que exponen simples servicios REST, y un proceso Batch. (Cuya especificación detallaremos a continuación).

**Entregable**

1. Se espera como entregable un ZIP con ambos proyectos.
2. Un script que levante las 2 aplicaciones.
3. Un documento corto explicando la solución.

**Tecnologías para todas las aplicaciones**

* Java 8
* Spring 5
* Mysql 5.X
* Maven
* Jetty server Embebido
* Si corresponde usar otras tecnologías

## Aplicaciones

### Aplicación Merchant

Servicios

1. CHECK

Dado el merchant\_id retorna un 200 si existe el merchant y un 404 en caso contrario.

### Aplicación PM (Payment Manager)

Servicios

**1.** SALE

Debe registrar una venta en la base de datos.

En base de datos debe guardarse siempre el amount en USD, (usando el servicio *http://fixer.io/*​ para obtener la cotización de la moneda).

**1.** a. ​**Request**

Request body:

{

"currency": String,

"amount": Decimal,

"transaction\_id": String,

"merchant\_id": String

}

Response Body:

{

"id": String

}

1. b. ​**Restricciones** 
   1. Este servicio debe ser idempotente sobre el transaction\_id para el mismo merchant\_id. Es decir, el servicio debe retornar el mismo "id" para la tupla (merchant\_id, transaction\_id).
   2. El id es autogenerado y único en la plataforma. iii. El servicio de fixer.io se debe consumir no más de una vez cada 30 segundos.
   3. Las transacciones comienzan todas en status "PENDING".
   4. El merchant\_id debe existir validandolo contra el servicio de "CHECK" de la aplicación Merchants.

1. STATUS

Servicio para tener el status de un cobro a partir del id de la respuesta del servicio "SALE".

**2** a. ​**Request**

Response Body

{

"id": String,

"date: String, //fecha de creación

"transaction\_id": String,

"merchant\_id": String "amount\_usd": Decimal,

"status": String // PAID, PENDING, REJECTED

}

3. BALANCE

Debe retornar el el monto acumulado (en USD) de PENDING, PAID y REJECTED, dado el merchant y un rango de fechas.

3. a. ​**Request**

Response Body:

{

"total\_paid": Decimal,

"total\_pending": Decimal,

"total\_rejected": Decimal

}

### Proceso Batch

Job que debe tomar todas las transacciones en status PENDING y con probabilidad 0.7 pase la transacción a PAID y el resto como REJECTED. Debe ejecutarse cada 30 segundos.

### Test

Se debe implementar tests en ambas aplicaciones.