Oblicagorio 1

Middleware

Grupo: 13

Integrantes:

Javier Hagopian: FALTA TU CI JAVI

Rafael García: 3.704.870-4

* Herramientas de desarrollo utilizadas:

1. Eclipse
2. Git
3. Maven

* Módulos entregados:

1. client
2. dispatcher
3. mobile-sys
4. red-strberry

* Formato de mensajes:

1. XML enviado desde al cliente y el dispatcher:

<order>

<head>

<number></number><!--NUMERO DE LA ORDEN-->

<date></date><!--FECHA DE LA ORDEN-->

<client-id></client-id><!--NUMERO DEL CLIENTE-->

<payment-mode></payment-mode><!--MEDIO DE PAGO-->

</head>

<detail>

<currency></currency><!--MONEDA-->

<amount></amount><!--TOTAL DE LA ORDEN-->

<payments></payments><!-- CUOTAS -->

</detail>

<item>

<number></number><!-- NUMERO DEL ITEM -->

<category></category><!-- CATEGORIA -->

<sku></sku><!-- ID PRODUCTO -->

<description></description><!--DESCRIPCION PRODUCTO-->

<quantity></quantity><!-- CANTADIAD -->

<price></price><!-- PRECIO -->

</item>

</order>

1. XML enviado por el dispatcher a mobile-sys

<mobileItem>

<clientId></clientId>><!--NUMERO DEL CLIENTE-->

<productId></productId><!-- ID PRODUCTO -->

<dateTime></dateTime><!--FECHA DE LA ORDEN-->

<IdOrdenItem></IdOrdenItem><!-—ID ORDEN + ID ITEM-->

</mobileItem>

1. Formato de la tabla de data-sys

**create** **table** IF **NOT** **EXISTS** ORDERS(CLIENT bigint,PRODUCT bigint,QUANTITY **int**,ORDER\_ID bigint,ITEM\_ID bigint,**PRIMARY** **KEY** (ORDER\_ID,ITEM\_ID));

1. Formato Servicio red-strawberry

**private** Long idProducto;

**private** **int** cantidad;

**private** String idTransaccion;

**private** Date fecha;

* Archivos de configuración:

1. dispatcher\config\application.properties:

server.port=8080

#puesto http, usado para spring-boot

logging.level=INFO

#nivel de loggin

ws.user=user

# usuario ws del servicio de red-strawberry

ws.password=password

# contraseña ws del servicio de red-strawberry

ws.wsdlLocation=https://localhost:8443/services/ItemsListsService?wsdl

# url ws del servicio de red-strawberry

aggregator.batchSize=1000

# tamaño de los bloques para enviar al ws

message.retry.quantity=2

# cantidad de reintentos de lectura de la cola jms antes de mandarlo a “DEAD LETTER”

broker.url=tcp://localhost:61616

#URL del broker JMS

jdbc.url=jdbc:postgresql://localhost/middleware

#url de la base data-sys

db.user=postgres

#usuario de la base data-sys

db.pass=postgres

#contraseña de la base data-sys

1. Client/middleware/JMSProducer.java

props.setProperty(Context.***PROVIDER\_URL***,"tcp://localhost:61616");

#url broker jms

1. mobile-sys\src\main\resources\META-INF\mobilesys\jms.xml

<property name=*"brokerURL"* value=*"tcp://localhost:61616"*/>

#url broker jms

