Int. a Middleware 2016
Laboratorio 2

Grupo 12

Integrantes:
Barreiro Deminco, Alejandro
Cabrera Gonzalez, Anthony Martin
Davila Cuevas, Mauricio Fabian
Oyharzabal Masares, Leonardo

1.Introducción

Ticket Inco

Componente que implementa Web Services SOAP utilizando JAX-WS brindando:

- Consulta de entradas disponibles
- Reserva de entradas
- Consulta de estado de reserva
- Confirmación de reserva de venta de entradas
- Anulación de venta de entradas

Partner de negocios (Vendedor)

Cliente SOAP UI que consume servicios de Ticket Inco

Partner de negocios (Medio de pago)

Pagos YA

Partner de negocios externo que brinda una API Rest sincrónica utilizando JAX-RS para confirmación y anulación de pagos.

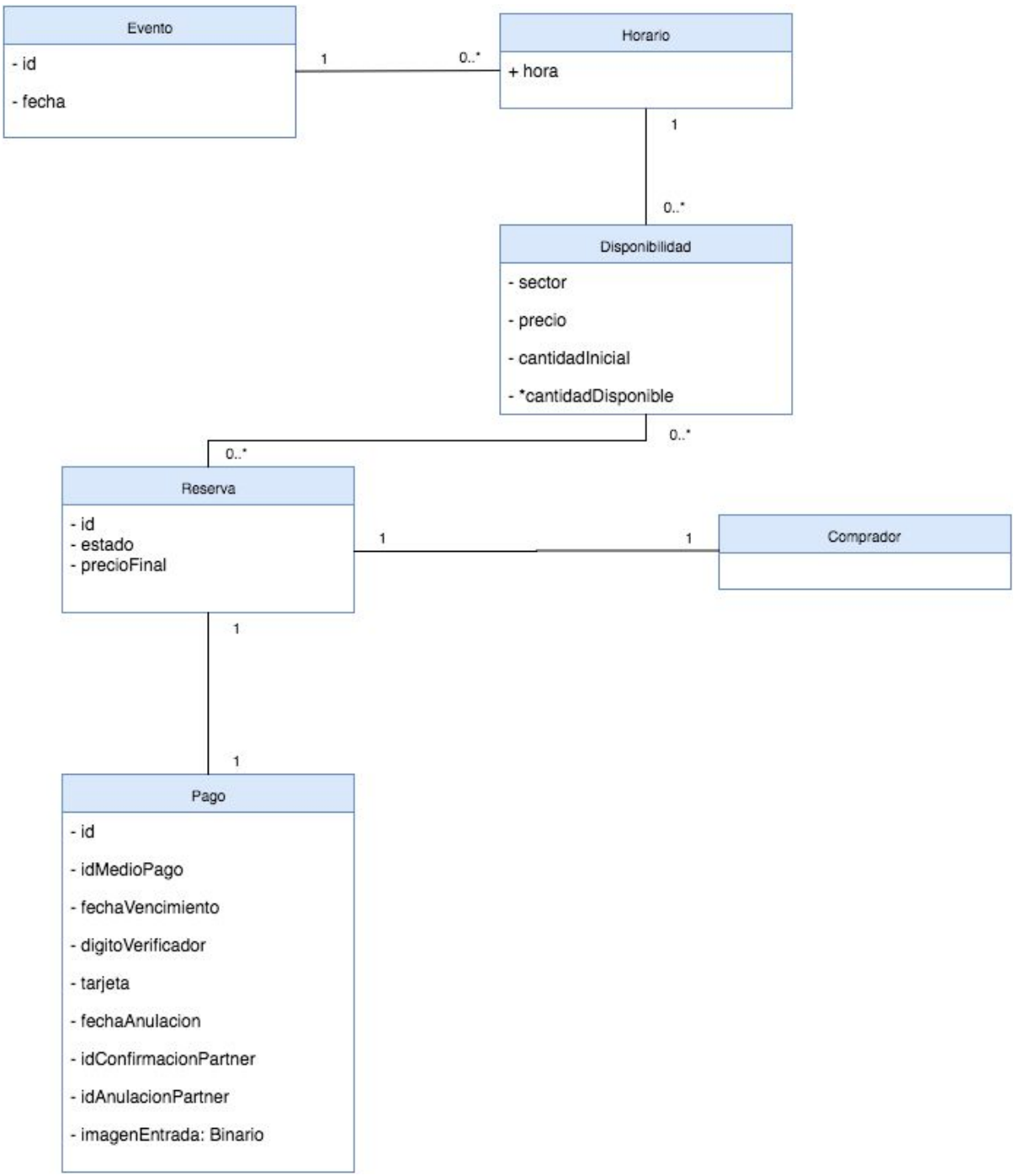
Pago local

Medio de pago propio de Ticket Inco basado en mensajería utilizando ActiveMQ para confirmación y anulación de pagos.

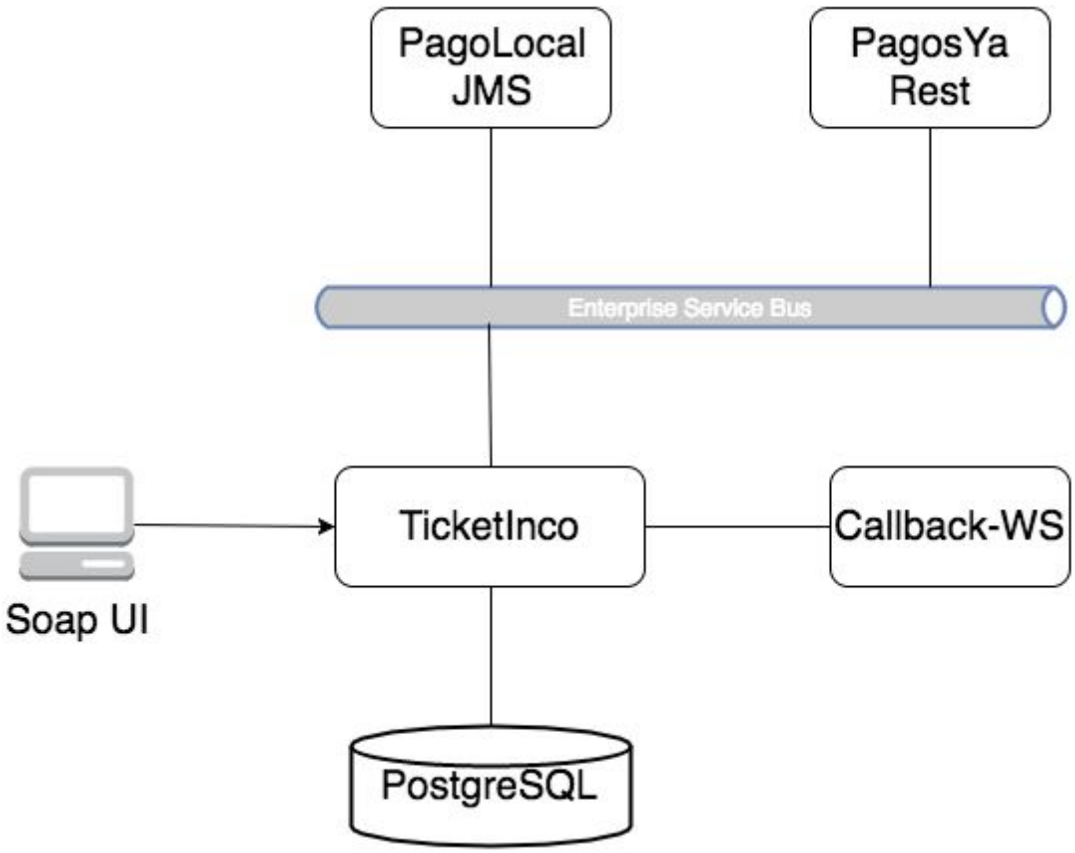
Servicio Callback

Para la recepción de confirmación de pagos y emisión de entradas por parte de los medios de pago se utilizo un WebService SOAP de Callback.

2. Modelo de dominio



3. Arquitectura



4. Detalles de implementación

Web Services SOAP

El sistema expone a través de web services SOAP funcionalidades para el manejo de entradas de Ticket Inco, y la implementación del Callback-WS.
Se utilizó como servidor de aplicaciones Apache Tomcat y como runtime de los servicios Apache CFX.

Security

Con WS-Security se está firmando o bien la cabecera del mensaje SOAP o bien el cuerpo o bien el Envelope SOAP entero, mediante algún tipo de procedimiento de esta manera se está asegurando la integridad del mensaje, del emisor y del receptor y si hay algún cambio en el Mensaje al ir firmado con una clave privada dará un error en el receptor, por ejemplo:

- Mediante Usuario y contraseña
- Mediante Certificado (soporta certificados X509 y rutas de certificados)
- Mediante el TimeStamp
- Combinación de estos anteriores.

WS-Security no es más que la implementación del estándar de OASIS Web Services Security TC, con lo cual está implementando los siguientes estándares:

- OASIS Web Services Security: SOAP Message Security 1.0 Standard 200401, March 2004
- Username Token profile V1.0
- X.509 Token Profile V1.0

En el caso de ticketinco se utilizó ws-security para el servicio SOAP de reserva de entradas, en concreto para firmar los mensajes enviados desde el cliente, de esta manera nos aseguramos la autenticidad del cliente que hace la petición.

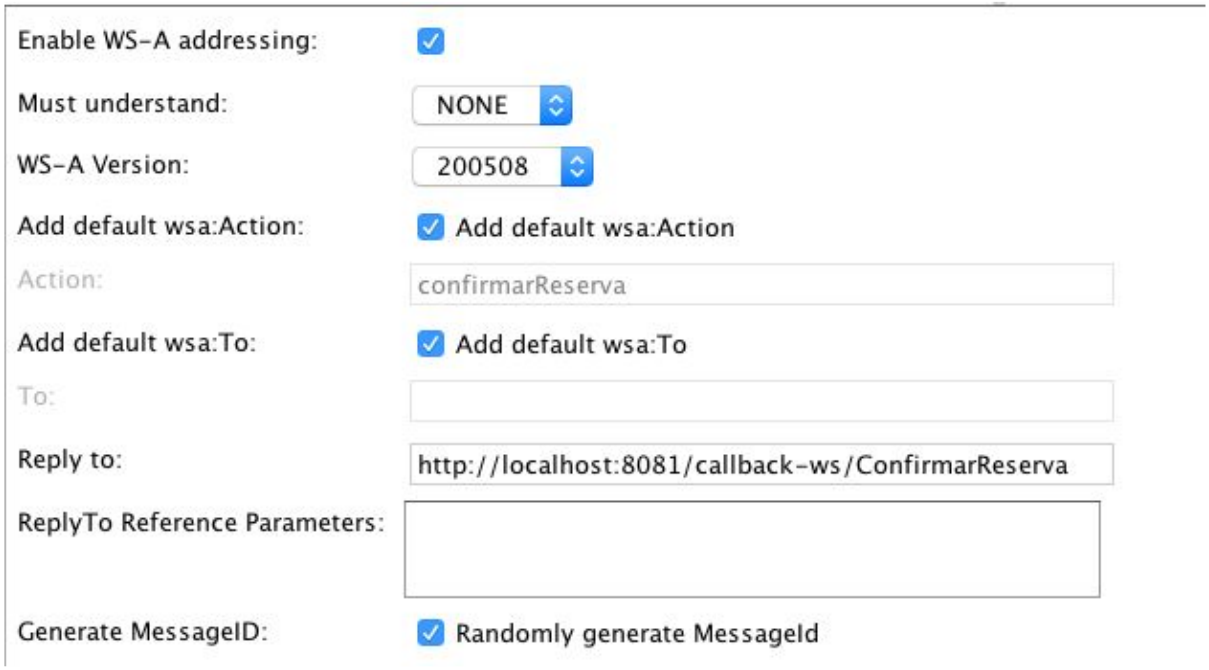
El otro servicio que se protegió con ws-s fue el de confirmación de reserva, en este caso los mensajes deben venir cifrados con ws-s. Se utiliza el cifrado con WS-Security para proveer confidencialidad en la comunicación

WS-Addressing

Para utilizar WS-Addressing fue necesario la implementación de un servicio de callback que es invocado por parte de Ticket Inco. Esto se implementó mediante el uso de la notación @Addressing(enabled = true).

El nombre y el namespace del servicio de Ticket Inco coincide con el servicio de callback, a su vez se especifica en @WebResult el nombre del parámetro que será enviado para su procesamiento.

Finalmente desde Soap UI es necesario invocar el servicio de Ticket Inco con WS-Addressing activado y especificar el servicio de respuesta.

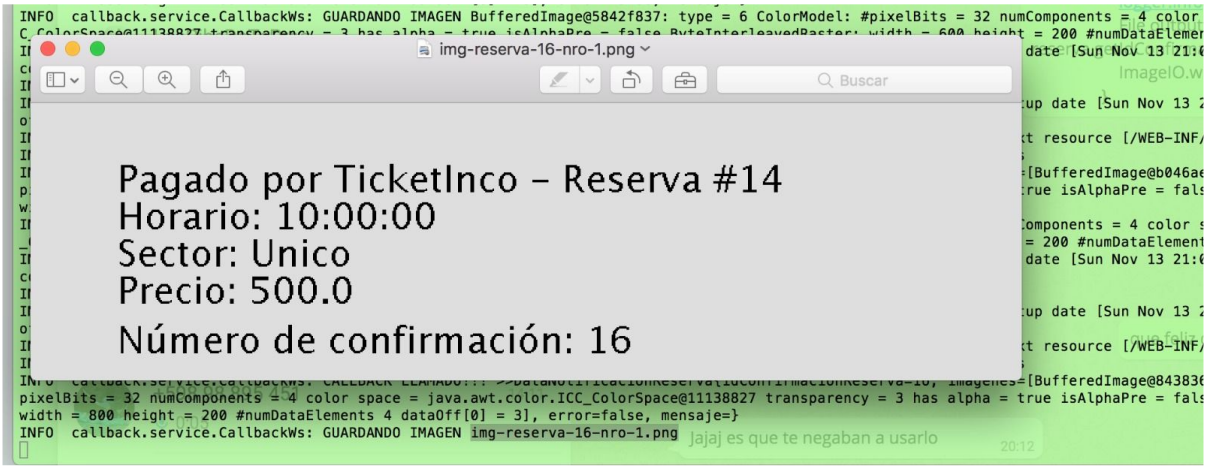


MTOM

SOAP MTOM (Message Transmission Optimization Mechanism) es el uso de MIME para optimizar la transmisión de las corrientes de bits de mensajes SOAP que contienen elementos base64Binary de un tamaño considerable.
El formato de mensajes MTOM permite comprimir la corriente de bits de los datos binarios. Los datos que, de otro modo, se deben codificar en el mensaje SOAP se transmiten como datos binarios en bruto en un parte MIME separada. Un gran fragmento de datos binarios ocupa menos espacio que su representación codificada, por lo tanto, MTOM puede disminuir el tiempo de transmisión, aunque requiere una determinada cantidad de actividad general de proceso. Los elementos que resultan candidatos para ser transmitidos de este modo se definen como base64Binary en el WSDL (XML Schema).

En cuanto a la implementación se agrega la anotación @MTOM(enabled=true) en el endpoint que se quiera configurar.
La anotación @MTOM tiene dos parámetros enabled y threshold. El parámetro enabled tiene un valor booleano e indica si se ha habilitado MTOM para el punto final JAX-WS. El parámetro threshold tiene un valor entero que debe ser superior o igual a cero. Cuando MTOM está habilitado, cualquier dato binario cuyo tamaño, en bytes, supere el valor de umbral se codifica mediante el empaquetado optimizado de binario XML (XOP) o se envía

como archivo adjunto. Cuando el tamaño del mensaje es inferior al valor del umbral, el mensaje se escribe en el documento XML como datos base64 o hexBinary.



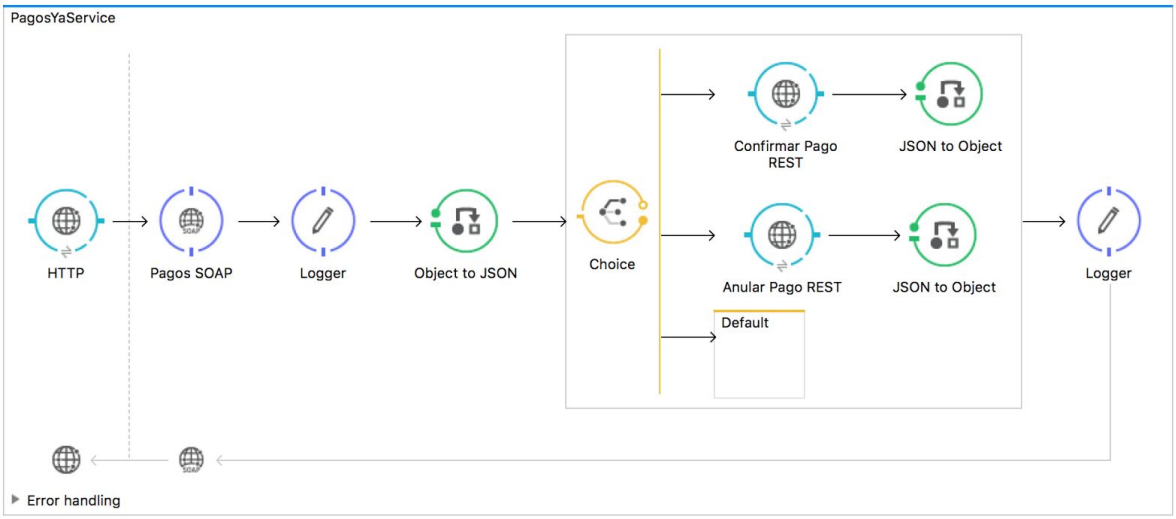
Web Services REST

Es una API que expone WebServices REST sincrónico utilizado para la confirmación y anulación de pagos. La comunicación con esta API se realiza por mensajes JSON a través de la fachada de Web Services SOAP realizada en ESB. Ambos servicios están protegidos con autenticación de tipo http basic, y son ejecutados por usuarios con rol tomcat los cuales son obtenidos del repositorio de usuarios del servidor.

Enterprise Service Bus

Se utiliza MuleESB como producto de tipo Enterprise Service Bus.
Se desarrollaron dos flujos que permiten la comunicación con cada uno de los medios de pago.

Flujo PagosYaService



Fachada que permite la comunicación con el medio de pago externo PagosYa!, el cual consiste en una api REST que permite la confirmación y anulación de pagos.

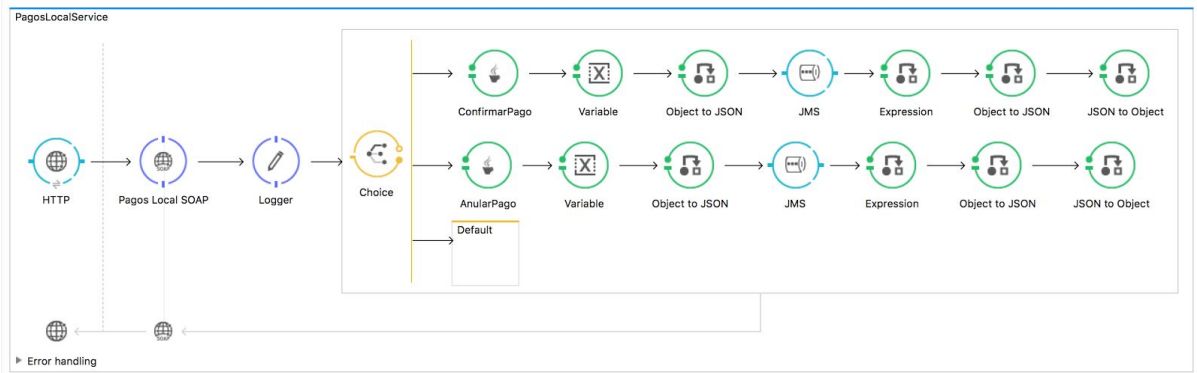
Se define la interfaz de un web service de tipo SOAP el cual tiene dos métodos asociados correspondiente a la confirmación y anulación del pago.

Se utiliza el componente Choice, el cual permite direccionar el pedido dependiendo del método del método ws invocado.

La comunicación con el servicio REST utiliza http basic como método de autenticación y ssl. La api REST utiliza el repositorio de usuario de Tomcat para la autenticación, es por ello que se generó un usuario específico para el componente ESB (ticket_inco_user)

Para la correcta comunicación con la api rest y la transformación de los datos de salida que contemplen el contrato del ws se utilizan elementos como *json to object, object to json*, etc.

Flujo PagosLocalService



Fachada que permite la comunicación con el medio de pago local, el cual permite únicamente una integración basada en colas de mensajes JMS, utilizando activeMQ como servidor de cola de mensajes. Este sistema cuenta con una única cola de mensajes para recibir los pagos como también las anulaciones.

Se define la interfaz de un web service de tipo SOAP el cual tiene dos métodos asociados correspondiente a la confirmación y anulación del pago.

Se utiliza el componente Choice, el cual permite direccionar el pedido dependiendo del método del método ws invocado.

En ambos subflujos se genera con un componente Java el id de confirmación o de anulación correspondientemente y se altera el payload del mensaje para agregar esta información al mismo de manera que quede registrado en la cola de mensajes.

Tanto el id de confirmación como el id de anulación se guardan en una variable del flujo que permite retornarlo como resultado del servicio soap, es por ello que se utilizan elementos del ESB como lo son *set-variable*, *transformers*, *expressions*, etc.