

Facultad de Ingeniería – Instituto de Computación
Introducción al Middleware
Evaluación Escrita – 02 de Diciembre de 2015

Duración: 3 horas

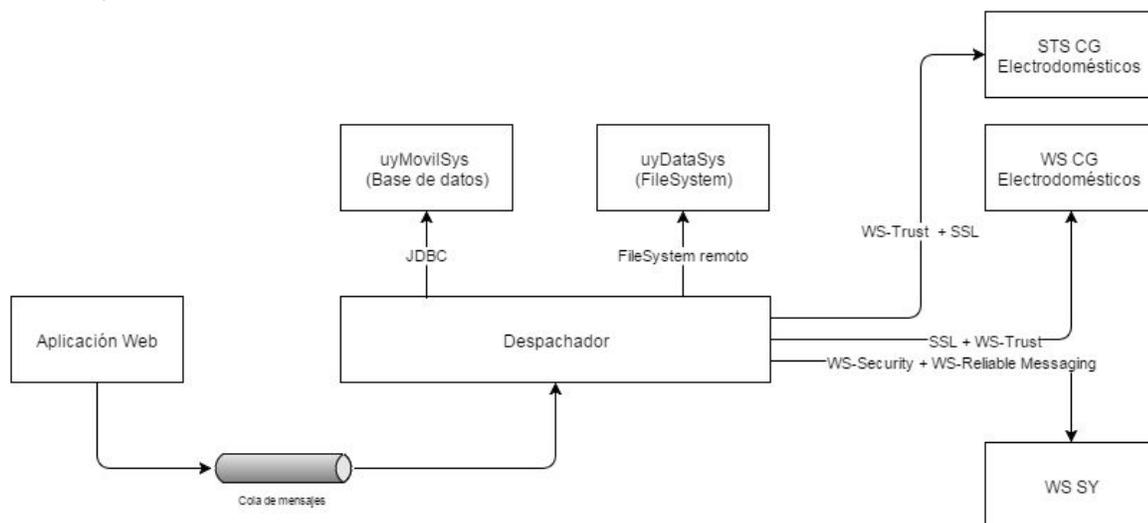
Preguntas (60 puntos)

Ver teórico.

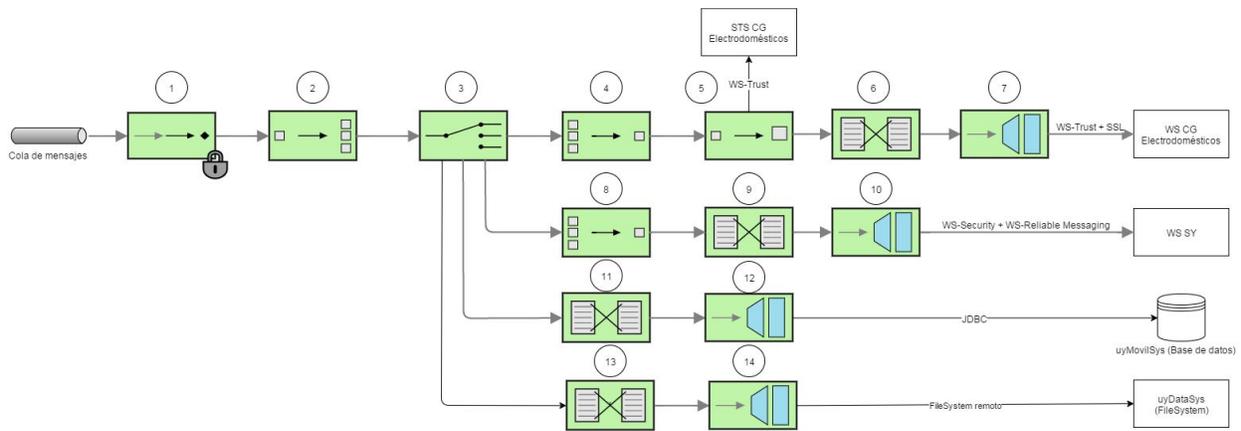
Problema (40 puntos)

En alto nivel, la solución se compone de:

- Una aplicación Web encargada de proveer una UI para la creación de órdenes de compra. La aplicación Web se comunica con el despachador mediante una cola de mensajes JMS.
- El sistema despachador es el encargado de recibir las órdenes de compra y distribuir las a los demás sistemas dependiendo del tipo de producto. Está suscrito a la cola de mensajes como un event-driven consumer con suscripción durable para evitar la pérdida de órdenes de compra.
- La comunicación con uyMovilSys se realiza vía JDBC contra una de sus bases de datos.
- La comunicación con uyDataSys se realiza vía una conexión con un FileSystem remoto depositando archivos CSV.
- La comunicación con CG Electrodomésticos se realiza vía el protocolo WS-Trust y SSL. Los tokens SAML se obtienen del STS de CG Electrodomésticos.
- La comunicación con SY se realiza vía Web Services SOAP, asegurando la integridad la información con firmas usando WS-Security. La confiabilidad en la entrega de mensajes se garantiza usando WS-Reliable Messaging, con una política de exactamente una vez y en orden.



A continuación se describe en detalle la arquitectura del sistema despachador, ilustrado en la figura.



Componentes:

1. Event-driven consumer con suscripción durable: Encargado de recibir la orden de compra desde la cola de mensajes.
2. Splitter: Encargado de dividir la orden de compra en una lista de productos.
3. Content Based Router: Encargado de dirigir los mensajes a un flujo determinado dependiendo del tipo de producto. Se asume que cada mensaje tiene el tipo de producto para poder hacer el ruteo.
4. Aggregator: Encargado de agrupar los productos en paquetes de a 1000 según el requerimiento de CG Electrodomésticos.
5. Content Enricher: Encargado de consultar al STS de CG Electrodomésticos y obtener el token SAML para consumir su Web Service.
6. Message Transformer: Encargado de transformar el formato del paquete de 1000 productos a un mensaje en formato SOAP.
7. Conector SOAP/SSL+WS-Trust: Encargado de enviar el mensaje al Web Service de GC Electrodomésticos vía SOAP/SSL + WS-Trust. Se adjunta token SAML (obtenido en 5) al mensaje.
8. Aggregator: Encargado de agrupar los productos en paquetes de a 500 según el requerimiento de SY.
9. Message Transformer: Encargado de transformar el formato del paquete de 500 productos a un mensaje en formato SOAP.
10. Conector SOAP+WS-Security+WS-Reliable Messaging: Encargado de enviar el mensaje al Web Service de SY vía SOAP, con WS-Security para la firma del mensaje y WS-Reliable Messaging para garantizar la entrega de mensajes. La política de entrega es exactamente una vez y en orden.
11. Transformer: Se encarga de transformar el mensaje a los tipos de dato de la BD de uyMovilSys.
12. Conector JDBC: Encargado de insertar una tupla en la BD de uyMovilSys destinada a este propósito.

13. Transformer: Se encarga de transformar el mensaje al formato CSV requerido por uyDataSys.
14. Conector Filesystem: se encarga de crear un archivo en FileSystem remoto del sistema uyDataSys con el formato CSV generado en 13..

Variantes:

- Uso del patrón Record Batcher para la comunicación con el sistema uyDataSys. Para usar este patrón, se debe definir un servicio en el ESB donde recibir la orden de compra.
- Usar un tópico de mensajes en conjunto con un Message Filter en lugar de un Splitter y un Content Based Router.

En caso de no proporcionar un dibujo de la solución, se debe explicar cómo es la interacción entre los componentes.