

Facultad de Ingeniería – Instituto de Computación
Introducción al Middleware
Evaluación Escrita – 03 de Diciembre de 2014

Duración: 3 horas

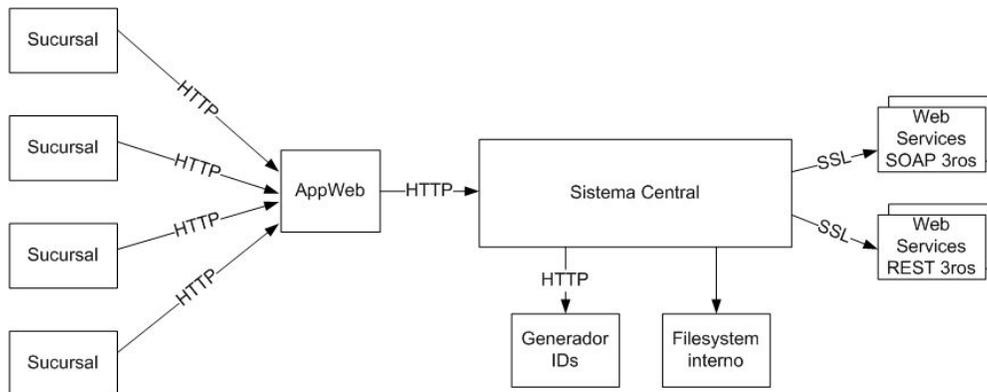
Preguntas (60 puntos)

Ver teórico.

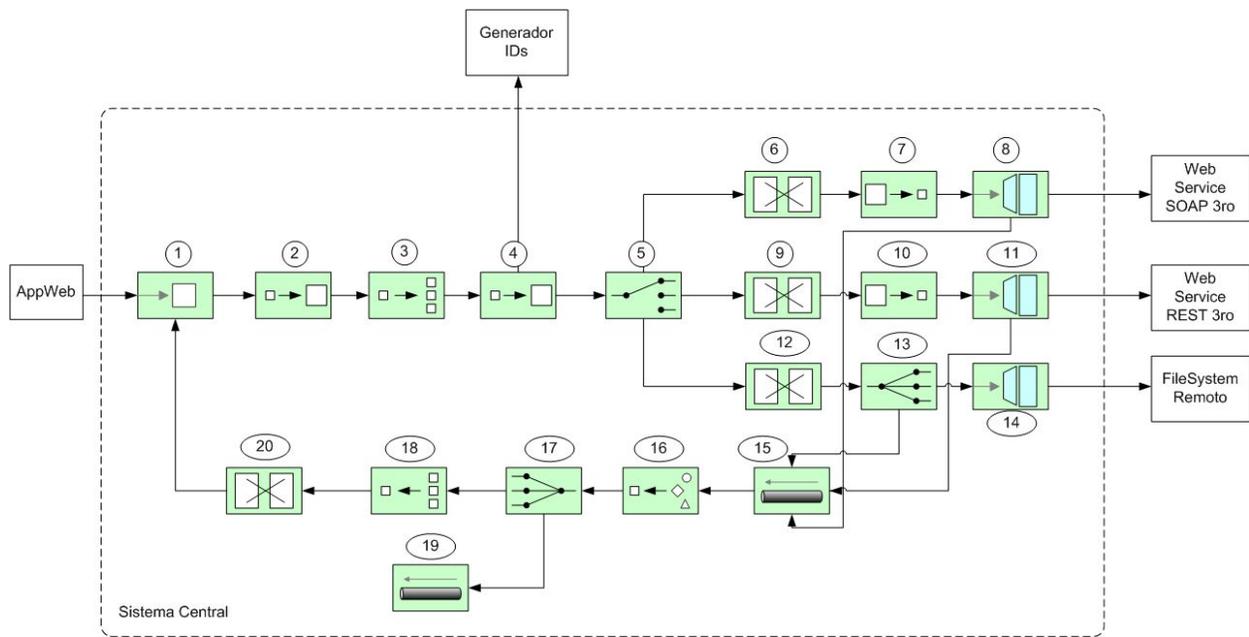
Problema (40 puntos)

En alto nivel, la solución se compone de:

- Una aplicación Web encargada de proveer una UI para el pago de servicios. La aplicación Web comunica los pagos al Sistema Central vía Web Services REST, en formato XML y protocolo HTTP.
- El sistema central encargado de recibir una lista de pagos y distribuirlos entre los Web Services (REST o SOAP) de las empresas terceras o en caso de no poseer un Web Service, enviar el pago a un Filesystem interno para su posterior procesamiento por un sistema interno. La comunicación con los Web Services (SOAP y REST) es vía SSL y con el Filesystem, vía una carpeta compartida. La comunicación con el generador de IDs es vía Web Services REST por HTTP y mensajes JSON.
- Los Web Services REST están asegurados mediante HTTP Basic Authentication y SSL
- Los Web Services SOAP están asegurados mediante WS-Security/Username Token y SSL
- Sucursales que se comunican con la aplicación Web vía HTTP.



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



Componentes:

1. HTTP Endpoint: Encargado de recibir la lista de pagos desde la aplicación Web. El formato de mensajes es XML.
2. Content Enricher: Encargado de asignar un identificador único al mensaje para ser utilizado como correlationID de la lista de pagos entre todos los componentes y más adelante por el aggregator. También agrega al mensaje la cantidad de pagos que tiene el mensaje.
3. Splitter: Encargado de dividir el mensaje con la lista de pagos en N mensajes, uno con cada pago de la lista. Cada mensaje tiene el correlationID definido en el paso 2.
4. Content Enricher: Encargado de agregar al mensaje el identificador único en cada pago, comunicándose para ello con el sistema generador de Ids interno.
5. Recipient List: Encargado de direccionar el mensaje por diferentes caminos según si la empresa destinataria del pago tiene o no Web Service (REST u SOAP). Se asume que en el mensaje está el nombre de la empresa.
6. Transformer: Se encarga de transformar el mensaje XML en un mensaje SOAP acorde al formato definido por el Web Service SOAP.
7. Content Enricher: Encargado de agregar la url y usuario y contraseña al mensaje. La url se coloca en el cabezal wsa:To de WS-Addressing y el usuario y contraseña en un cabezal WS-Security/Username Token. Los datos se obtienen de una BD interna, no descrita en el dibujo para mantener su claridad.
8. Conector SOAP/SSL: Encargado de enviar el mensaje al Web Service de tercero vía SOAP/SSL según la URL del cabezal de wsa:To de WS-Addressing. La respuesta de este mensaje se envía a una cola de mensajes descrita en el punto 15.
9. Transformer: Se encarga de transformar el mensaje XML en un mensaje JSON acorde al formato definido en los Web Services REST.

10. Content Enricher: Encargado de agregar la url y usuario y contraseña al mensaje ESB. Los datos se obtienen de una BD interna, no descrita en el dibujo para mantener su claridad.
11. Conector REST/SSL: encargado de enviar el mensaje al Web Service REST según la url definida en el mensaje SOAP. La información para la seguridad HTTPBasic se obtiene del mensaje ESB enriquecido en el punto 10. La respuesta de este mensaje se envía a una cola de mensajes descrita en el punto 15.
12. Transformer: Se encarga de transformar el mensaje XML en un mensaje CSV acorde al formato definido por el sistema interno.
13. Recipient List: Se encarga de enviar una copia del mensaje al Conector Filesystem definido en el punto 14 y a la cola de mensajes definida en el punto 15
14. Conector Filesystem: se encarga de copiar el mensaje a una carpeta en el filesystem. Esta carpeta es una carpeta ubicada en un servidor remoto.
15. Cola de mensajes: se encarga de recibir diferentes mensajes: de respuesta o enviados al filesystem interno
16. Normalizador: encargado de transformar los diferentes mensajes que recibe a un formato único en XML. Transforma las respuestas en JSON, en SOAP y el archivo CSV a un mensaje de respuesta normalizado.
17. Recipient List: se encarga de enviar una copia del mensaje a una cola de mensajes para la comunicación con el sistema de comunicaciones, si y solo si, el pago es confirmado correctamente (restricción configurada en RecipientList). Otra copia se envía al aggregator definido en el punto 18.
18. Aggregator: se encarga de agrupar los mensajes según el correlationID definido en el punto 2 y liberar los mensajes una vez que se reciben todas las respuestas. Este dato también está dado por información en cada mensaje. Cada mensaje indica la cantidad total de respuestas. Ver punto 2.
19. Cola de mensajes: encargada de recibir los mensajes confirmados. Punto de extensión para la comunicación con el sistema de comisiones. Este sistema simplemente deberá integrarse con la cola de mensajes para comenzar a recibirlos, sin realizar cambios en el sistema central.
20. Transformer: transforma el mensaje con todas las respuestas de los pagos al formato requerido por la aplicación Web.

Variantes:

- Uso del patrón Record Batcher para la comunicación con el sistema interno de procesamiento de pagos offline (filesystem remoto).
- Utilizar un tópico de mensajes para la extensibilidad con el sistema de comisiones y otros sistemas terceros.
- SSL con autenticación mutua para la seguridad con los Web Services.
- WS-Trust para la seguridad con los Web Services
- SOAP o colas de mensajes para la comunicación entre la appWeb y el sistema central. En caso de usarse colas de mensajes, se explicita que no existe un firewall entre la appWeb y el sistema Central.

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

Parte 3)

Se utiliza Performance como dimensión y tiempo de respuesta como de factor de calidad. La fórmula para calcular el tiempo de respuesta es:

$T_{Resp} = \text{Tiempo Final} - \text{Tiempo inicial}$

Tiempo Final y tiempo inicial se miden en el punto 1 de la solución