

“Iteración 5 – Transacciones Distribuidas”

Juan J. Díaz Baquero, Jose M. Suarez Lopera

Claudia D. Bedoya Motta, Johnathan S. Salamanca Lancheros

Transacciones Distribuidas

Universidad de los Andes, Bogotá, Colombia

{cd.bedoya212, js.salamanca1967, jm.suarez201, jj.diaz1067 }@uniandes.edu.co

Fecha de presentación: Mayo 20 de 2014

Tabla de contenido

| | | |
|---|--|---|
| 1 | Introducción | 1 |
| 2 | Alistamiento de bases de datos y servicios | 1 |
| 3 | Restricciones existentes para los sitios | 1 |
| 4 | Ajustes en las arquitecturas | 3 |
| 5 | Requerimientos | 4 |

1 Introducción

El presente documento tiene como objetivo revisar los conceptos de vistos en clase con el fin de profundizar y poner en práctica lo aprendido. Por otro lado, también se busca integrar requerimientos con el fin de implementar transacciones distribuidas, integrando dos bases de datos diferentes.

2 Alistamiento de bases de datos y servicios

En primer lugar, se realizó una copia de las bases de datos de tal forma que los datos se mantengan seguros mientras se realiza la conexión entre las aplicaciones por medio de las colas de mensajes. Asimismo, se ha asegurado la disponibilidad de cada aplicación por medio de JMS de tal manera que se pueda garantizar una conexión entre las aplicaciones por medio de las colas de mensajes.

3 Restricciones existentes para los sitios

En primer lugar se debe aclarar que ambas aplicaciones poseerán la interfaz XA para garantizar un protocolo de comunicación entre las aplicaciones tal que se pueda garantizar la escritura y lectura de la cola de mensajes por parte de las mismas de forma que se garantice un protocolo pregunta-respuesta que ayude a ambas aplicaciones a comunicarse entre sí y a ayudar a responder los

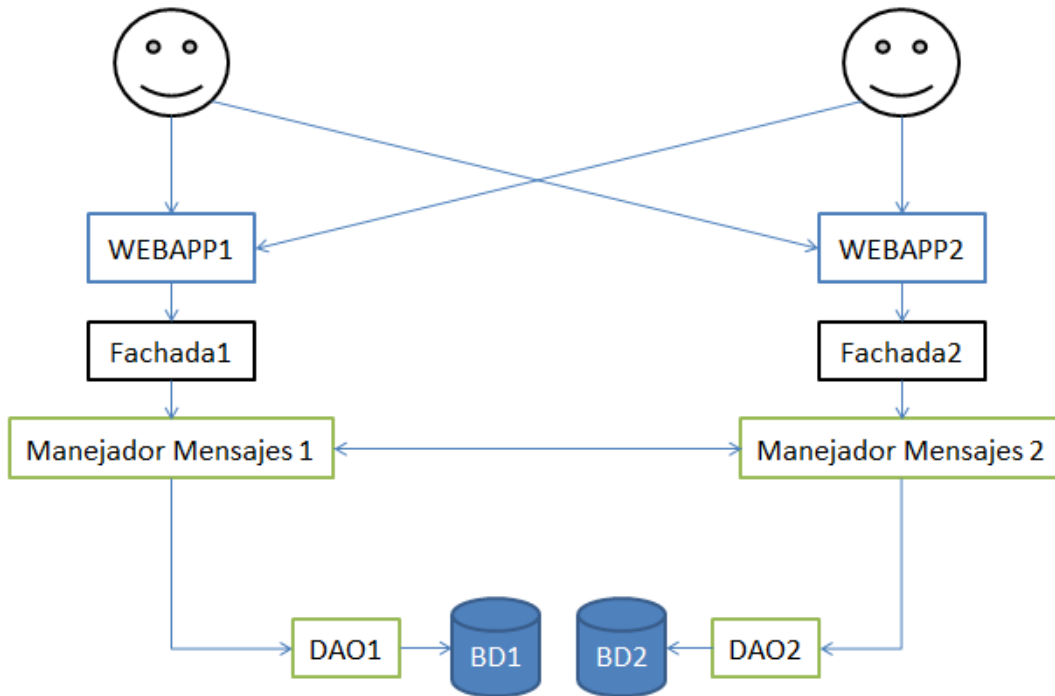
requerimientos funcionales que solicitan una respuesta mutua ante los requerimientos de un usuario de una aplicación.

De esta manera se quiere también que ambas aplicaciones cuenten con las siguientes colas de mensajes:

| Nombre | Grupo 5 | Grupo 9 | Descripción |
|----------------------------------|-----------|-----------|--|
| ColaPeticonesGrupo5CabAndes | Escritura | Lectura | Cola usada por el grupo 5 para poner mensajes con peticiones de requerimientos al otro grupo |
| ColaPeticonesGrupo9CabAndes | Lectura | Escritura | Cola usada por el grupo 9 para poner mensajes con peticiones de requerimientos al otro grupo |
| ColaRespuestasRF25Grupo5CabAndes | Escritura | Lectura | Cola usada por el grupo 5 para poner mensajes con las respuestas de las peticiones de RF25 |
| ColaRespuestasRF25Grupo9CabAndes | Lectura | Escritura | Cola usada por el grupo 9 para poner mensajes con las respuestas de las peticiones de RF25 |
| ColaRespuestasRF26Grupo5CabAndes | Escritura | Lectura | Cola usada por el grupo 5 para poner mensajes con las respuestas de las peticiones de RF26 |
| ColaRespuestasRF26Grupo9CabAndes | Lectura | Escritura | Cola usada por el grupo 9 para poner mensajes con las respuestas de las peticiones de RF26 |

De tal forma que se asegure una óptima comprensión y comunicación entre ambas aplicaciones asegurando que entre ellas exista un protocolo claro de comunicación por medio de JMS y la interfaz XA para que se pueda dar cumplimiento al protocolo de pregunta-respuesta para los requerimiento que lo necesitan (25 y 26) y de simplemente peticiones para los que también lo requieren (23 y 24)

4 Ajustes en las arquitecturas



Se puede observar que los usuarios de ambas aplicaciones web se comunicarán solamente con la interfaz de la aplicación, no hay distinción de cual utilizan por lo cual cualquier usuario puede acceder a cualquier aplicación y obtendrá los mismos resultados en la una o en la otra. La interacción entre la base de datos y la aplicación sigue siendo de la misma forma (Se comunican a través de la DAO) el único cambio presente es que ahora se poseerán manejadores de mensajes en ambas aplicaciones los cuales se comunicarán gracias a JMS por medio del protocolo de Colas de Mensajes

5 Requerimientos

REQUERIMIENTO FUNCIONAL 23

Especificación de la lógica – Diagrama de flujo

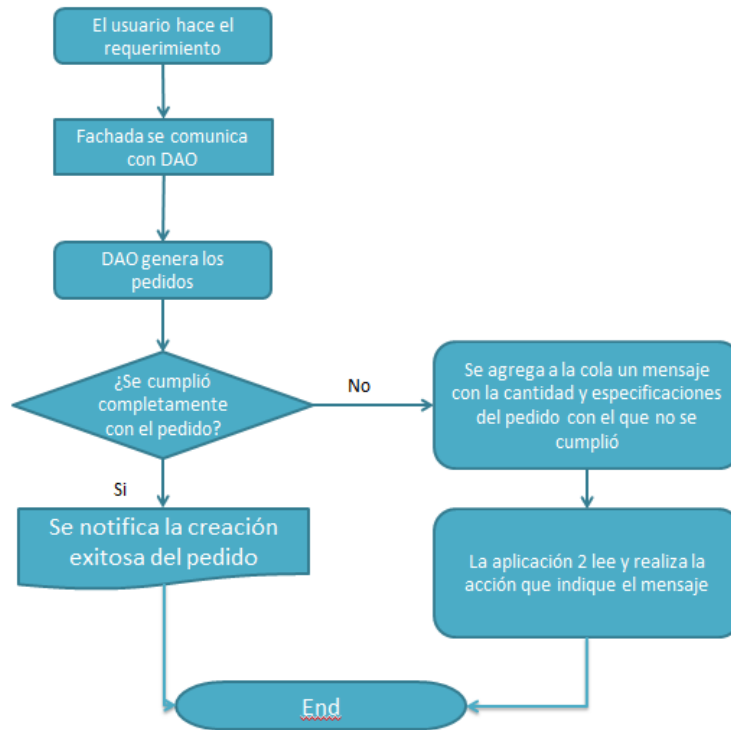
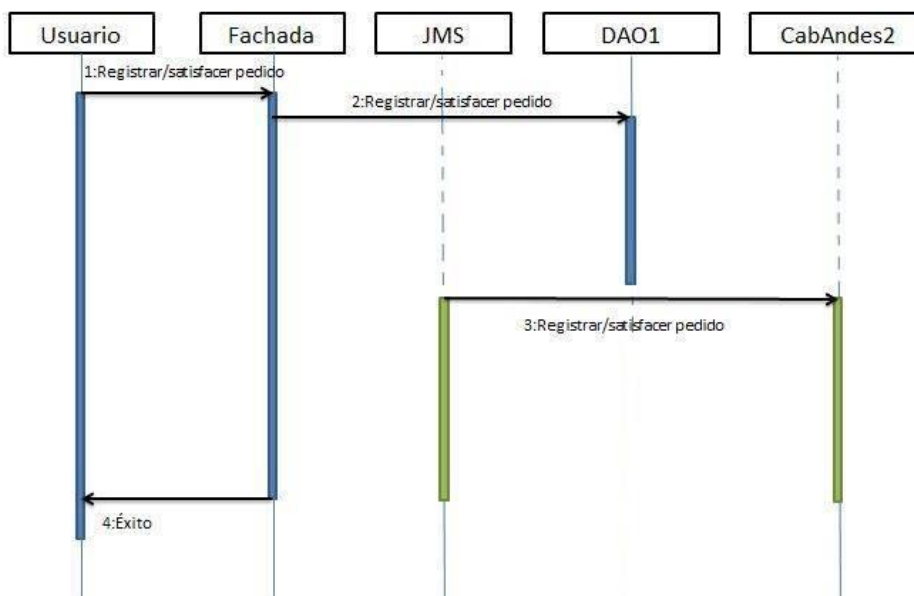


Diagrama de secuencia



El diagrama de flujo y el de secuencia expresan claramente la consecución de los pasos en el desarrollo del requerimiento funcional al leerse de izquierda a derecha y de arriba abajo asumiendo que arriba es el inicio del flujo. Se observa que el usuario realiza su petición, la fachada le comunica dicha decisión a la DAO y se tratará de satisfacer el pedido. En caso que las existencias del producto no alcancen para suplir la demanda se procederá a pedir a la aplicación 2 que supla los faltantes.

Implantación de la estrategia solicitada

Para el correcto funcionamiento del método se utilizó el protocolo de Colas de Mensajes utilizando JMS, para lo cual se creó una cola exclusivamente para que la aplicación se comunicara con la otra y se distinguiera este requerimiento de los demás para ambas aplicaciones de tal forma que no hubiera confusiones y se siguieron los siguientes pasos:

- i. Al recibir la orden de la fachada la DAO ejecuta el requerimiento normalmente asegurándose de cumplir al máximo con el pedido solicitado por el usuario. Ne le momento en que no se tengan unidades para suplir la demanda se comenzará a escribir un mensaje con la siguiente forma:

```
RF23
idProducto1;idPresentacion1;cantidad1
idProducto2;idPresentacion2;cantidad2
...
```

- ii. Si faltan unidades por suplir se mandará un mensaje con las unidades faltantes a la cola de mensajes tal que la otra aplicación lo sepa y supla lo faltante. En caso contrario, aún se mandará un mensaje a la cola de mensajes diciendo que la cantidad faltante es 0.

Análisis de las estrategias del requerimiento

La estrategia planteada para es bastante eficiente puesto que asegura que se supla completamente el requerimiento del usuario lo que genera, además, al ser secuencial asegura que la segunda parte de la transacción (la que realiza la segunda aplicación web) sea realizada a cabalidad. Se garantiza el correcto funcionamiento del requerimiento teniendo en cuenta que cada uno de los requerimientos posee una cola especial para lectura y escritura y teniendo en cuenta que las colas utilizadas son FIFO (First In, First Out) y que no se terminará la transacción hasta que la otra aplicación responda el mensaje se garantiza que aún en un escenario concurrente se mantenga la transaccionalidad.

Asimismo, al utilizar Two Phase Commit se asegura el perfecto cumplimiento de la labor sin contratiempos y de la mejor manera posible, teniendo en cuenta que son solamente 2 bifurcaciones el protocolo ayudaría demasiado al correcto y eficiente desarrollo del método teniendo en cuenta que no se acabará la transacción ni se hará commit a menos que ambas aplicaciones funcionen correctamente.

REQUERIMIENTO FUNCIONAL 24

Especificación de la lógica – Diagrama de flujo

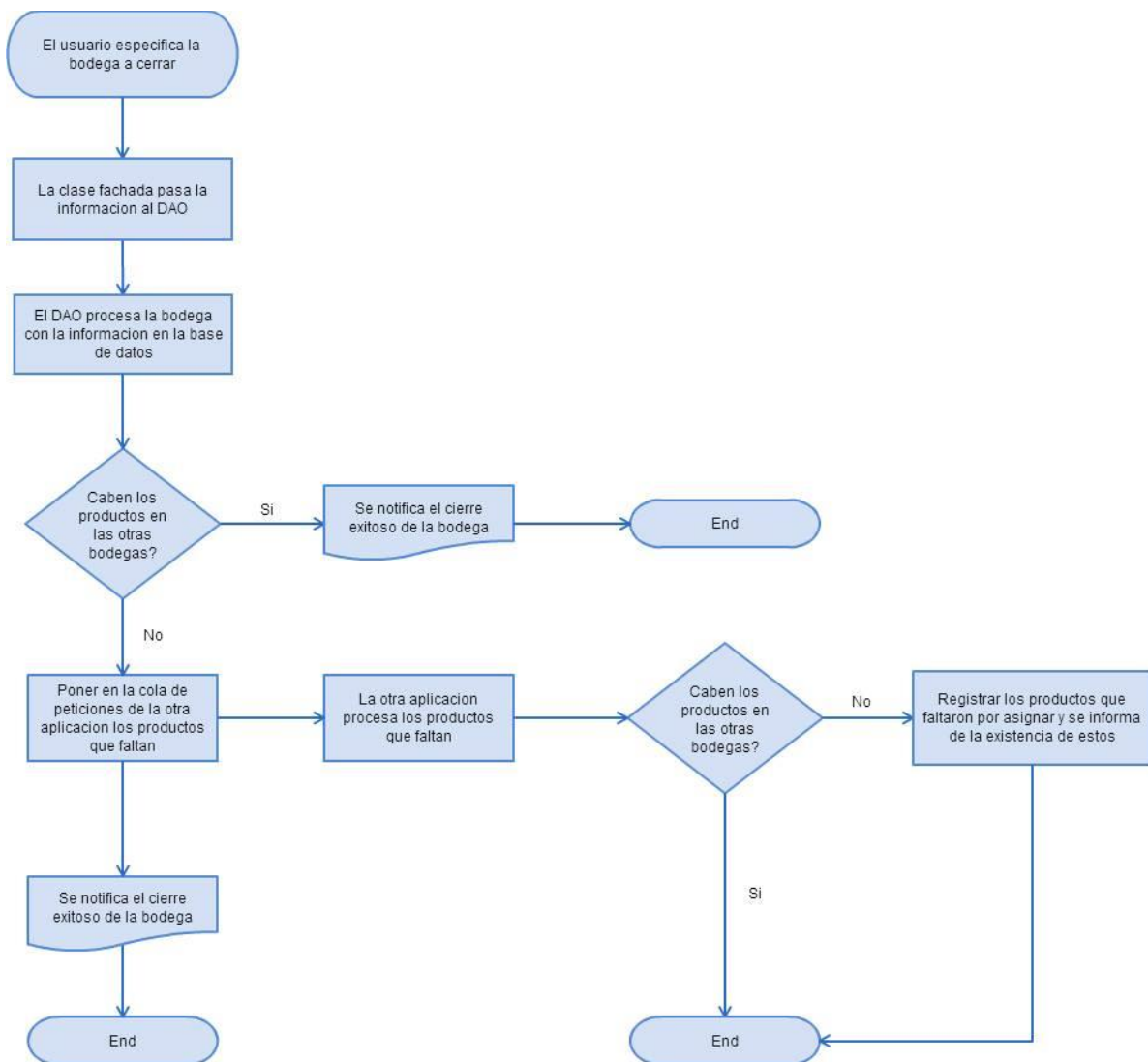
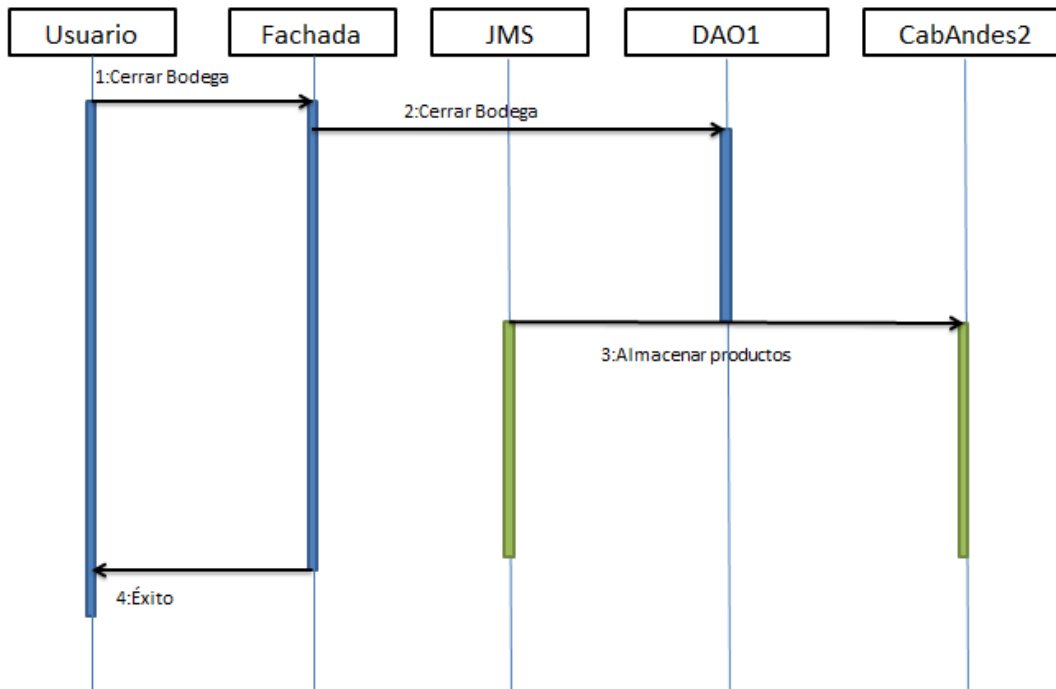


Diagrama de secuencia



Se puede observar que el diagrama de flujo es similar al diagrama de flujo del requerimiento pasado con la única particularidad que ahora la conexión con la otra base de datos se utilizará para almacenar los productos que generen sobrecupo en las bodegas de la aplicación 1.

Implantación de la estrategia solicitada

Para el correcto desarrollo del requerimiento se utilizaron colas de mensajes, para este caso se utiliza la misma cola de mensajes que para el requerimiento funcional 23 (Realizar / satisfacer pedido) teniendo en cuenta que es una cola de peticiones únicamente. Para el correcto desarrollo del requerimiento se seguirán los siguientes pasos:

- i. Se desarrolla el requerimiento normalmente, es decir, se cierra la bodega y se trasladan los productos restantes a las diferentes bodegas pertenecientes a CabAndes1. En el momento en que todas las bodegas estén llenas se comenzará a escribir un mensaje con el siguiente contenido:

RF24
 idTipoProducto
 idProducto1;idPresentacion1;cantidad1;numeroLote1; fechaExpiracionLote1
 idProducto2;idPresentacion2;cantidad2;numeroLote2; fechaExpiracionLote2
 ...

- ii. Al acabar de desarrollar el requerimiento solicitado en la aplicación 1 se procede a enviar el mensaje a la cola de mensajes y se termina la transacción al momento de haber recibido una respuesta positiva.

Análisis de las estrategias del requerimiento

La estrategia utilizada para el requerimiento es adecuada para el correcto desarrollo y funcionamiento del requerimiento funcional teniendo en cuenta que la cola de mensajes asegurará en un solo mensaje que se cumpla a cabalidad la transaccionalidad del requerimiento puesto que la transacción no terminará hasta que la aplicación 2 realice lo que le ha solicitado la aplicación 1. Esto se asegura teniendo en cuenta que la cola es FIFO.

Por otro lado, al utilizar Two Phase Commit se asegura el perfecto cumplimiento del requerimiento sin contratiempos en caso de que todas las aplicaciones estén disponibles y que ninguna haga Rollback varias veces. Hay que tener en cuenta que en two phase commit si se garantiza la consistencia de ambas bases de datos, cosa que no se garantiza con el protocolo de colas de mensajes. Por ultimo cabe resaltar que al tener que sincronizar solo dos aplicaciones, la probabilidad de que ocurran varios rollback seguidos antes de que complete la transacción es poca y la deficiencia de este protocolo no afectaría tanto el comportamiento de las aplicaciones. Es por lo anterior que se decide que el protocolo two phase commit es el protocolo más adecuado para manejar este requerimiento funcional.

REQUERIMIENTO FUNCIONAL 25

Especificación de la lógica – Diagrama de flujo

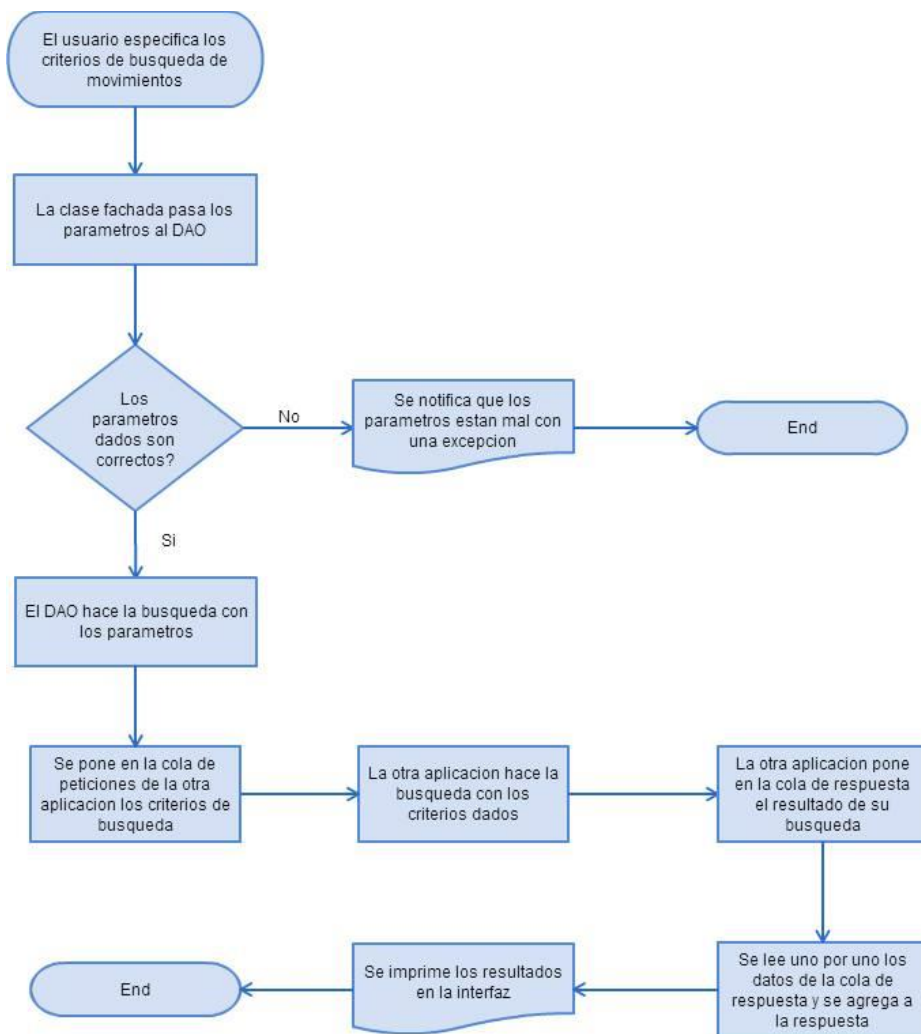
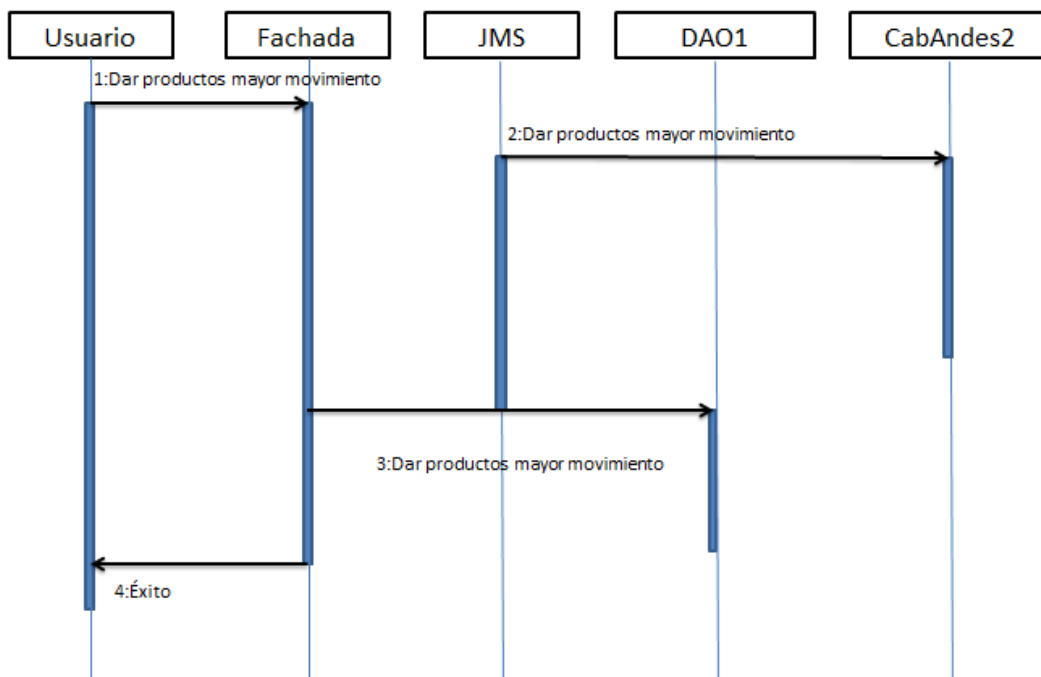


Diagrama de secuencia



Se puede observar que ahora el funcionamiento del requerimiento necesita que se pida primero a la aplicación 2 que dé respuesta al requerimiento tal que se pueda añadir dicha respuesta a la que brinde la aplicación actual.

Implantación de la estrategia solicitada

Para el correcto desarrollo del requerimiento se plantea utilizar dos colas de mensajes para asegurar un correcto protocolo de pregunta-respuesta para el manejo de las comunicaciones entre las dos aplicaciones puesto que se necesita obtener una respuesta por parte de la aplicación 2. Para el correcto funcionamiento del requerimiento se seguirán los siguientes pasos:

- i. A penas comience a ejecutarse la transacción se mandará un mensaje a la cola de mensajes correspondiente con el siguiente mensaje:

```
RF25
cantidad
idProducto
idPresentacion
idBodegaOrigen
idLugarDestino
fechaMinima
fechaMaxima
```

- ii. Se esperará una respuesta por parte de la aplicación 2, la cual tendrá el siguiente formato:

```
RF25
numeroMovimiento1;idLugarOrigen1;nombreLugarOrigen1;idLugarDestino1;nombreLugarDestino1;fechaMovimiento1;idProducto1;idPresentacion1;cantidad1
```

```
numeroMovimiento2;idLugarOrigen2;nombreLugarOrigen2;idLugarDestino2;nombreLugarDestino2;fechaMovimiento2;idProducto2;idPresentacion2;cantidad2  
...
```

- iii. Se procederá al desarrollo del requerimiento por parte de la aplicación 1 y se agregarán los resultados encontrados a los retornados por la aplicación 2.

Análisis de las estrategias del requerimiento

La estrategia a seguir es bastante eficaz y óptima teniendo en cuenta que garantizará la transaccionalidad de los requerimientos teniendo en cuenta que la transacción no acabará a menos que reciba una respuesta por parte de la aplicación 2. Cuando esto suceda la aplicación podrá seguir el desarrollo del requerimiento de forma normal lo cual asegura que se mantendrá ACID a lo largo de la misma en todo momento.

REQUERIMIENTO FUNCIONAL 26

Especificación de la lógica – Diagrama de flujo

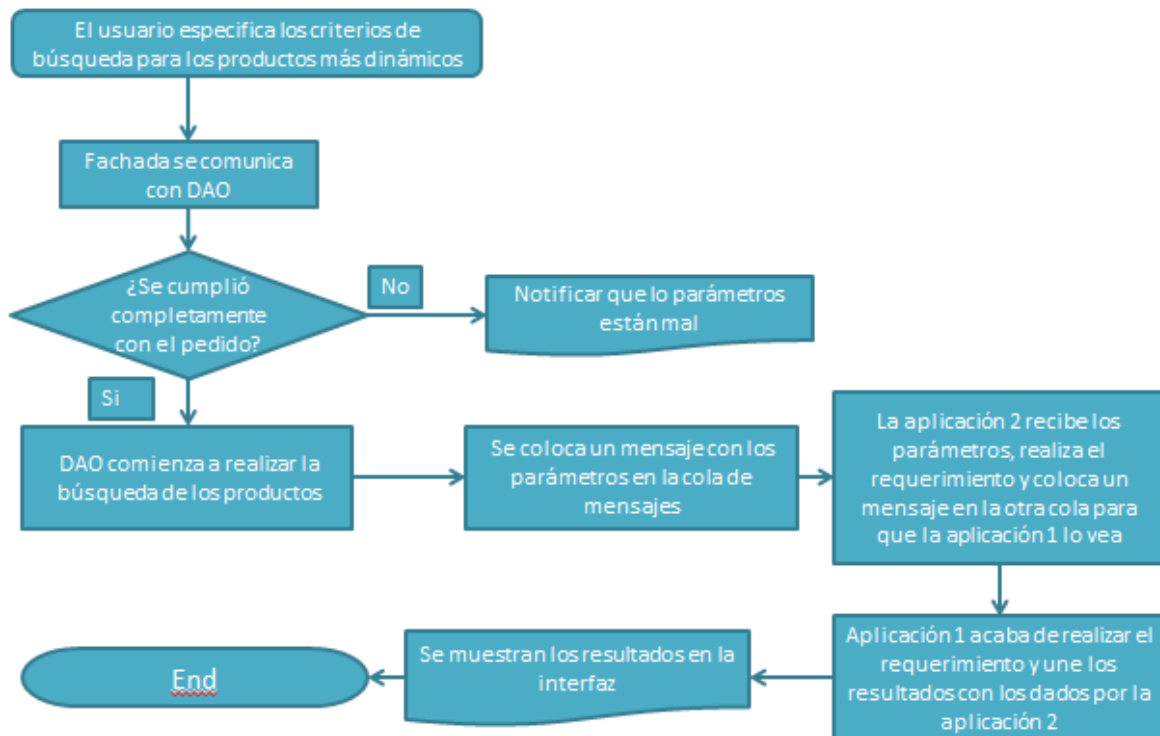
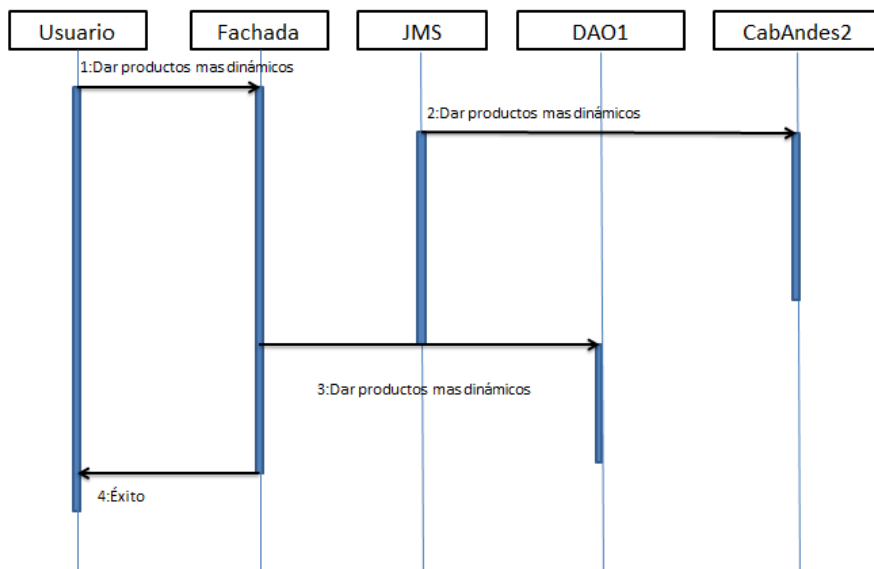


Diagrama de secuencia



Se puede observar que, al igual que para el requerimiento anterior, para este requerimiento el diagrama de flujo cambia puesto que ahora se necesita que ambas aplicaciones web respondan a la petición del usuario, no únicamente para un caso especial que no se cumpla alguna condición, por lo cual se realiza la petición a la otra aplicación apenas comienza la transacción.

Implantación de la estrategia solicitada

La estrategia utilizada será una Cola de Mensajes para lo cual se utilizarán dos colas de mensaje para manejar un protocolo de pregunta-respuesta teniendo en cuenta que el requerimiento actual exige que se obtenga una respuesta concreta a la petición realizada a la aplicación 2. Para esto se seguirán los siguientes pasos:

- i. Se manda el mensaje a la aplicación 2 a la cola destinada para ello. El mensaje tendrá el siguiente formato:

```
RF26
fechaMinimaMovimiento
fechaMaximaMovimiento
```

- ii. Se espera una respuesta por parte de aplicación 2 en la siguiente cola de mensajes, la cual debe poseer siguiente formato:

```
RF26
idProducto1;idPresentacion1;promedioPeso1;costoPromedio1;vecesHaSidoPedido1;vece
sHaSidoVendido1
idProducto2;idPresentacion2;promedioPeso2;costoPromedio2;vecesHaSidoPedido2;vece
sHaSidoVendido2
...
```

- iii. Se desarrolla el requerimiento normalmente en la aplicación 1 de tal forma que se una la respuesta a la obtenida por la aplicación 2.

Análisis de las estrategias del requerimiento

La estrategia utilizada para el desarrollo del requerimiento funcional es la más adecuada teniendo en cuenta que asegura perfectamente la transaccionalidad del requerimiento funcional puesto que no sigue adelante si no encuentra una respuesta por parte de la otra aplicación, lo que asegura ACID de la transacción.

ESTRATEGIA GLOBAL MÁS INDICADA

Atendiendo a los diversos análisis realizados se puede observar que el mejor protocolo para desarrollar el requerimiento funcional 23 (Registrar / satisfacer pedidos) es el protocolo de Two Phase Commit teniendo en cuenta que asegurará que la aplicación 2 realice los cambios pertinentes en caso que se necesite que acabe de cubrir el pedido realizado tal que el requerimiento se cumpla a cabalidad. Para el desarrollo de los demás requerimientos se observa que es mejor utilizar las colas de mensajes teniendo en cuenta que el protocolo de pregunta-respuesta utilizado acata con ACID y asegura un mejor funcionamiento de la aplicación. Aun así, cuando se desea hablar de optimizar el tiempo de ejecución de la transacción es mejor utilizar Two Phase Commit en todos los casos puesto que realiza todas las subtransacciones al tiempo y garantiza que el tiempo de ejecución sea el de la subtransacción más demorada.