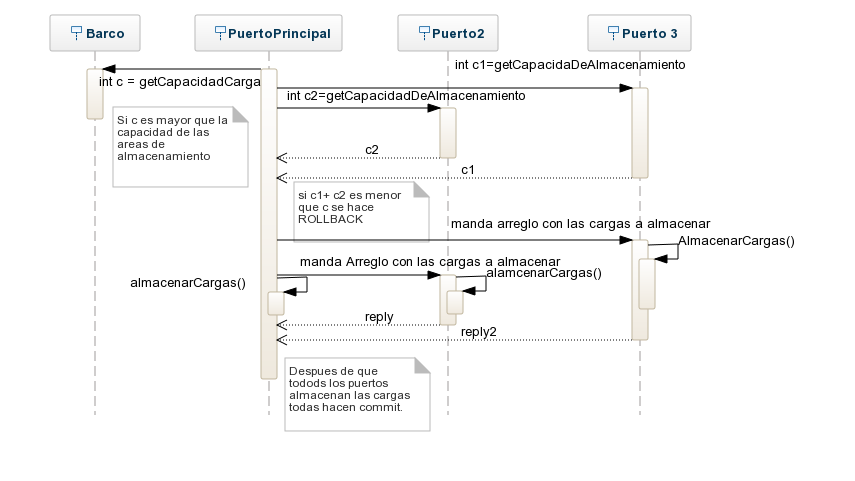
1.3

La lógica que se maneja en el RF14 es:

1. Se verifica en qué áreas de almacenamiento pueden ser guardas las cargas.
2. Se suma las capacidades de estas áreas de almacenamiento
3. Se da el volumen total de las cargas
4. Si el volumen es menor que la capacidad de las áreas de almacenamiento se guardan todas las cargas en el puerto.

Si no, se le pide la información de sus capacidades a los otros puertos, para mandar cargas hacia allá.

1. Se recibe la información de las capacidades de los dos puertos.
2. Se analiza si la carga cabe en los puertos, si no cabe se hace rollback. Si cabe se continua la operacion
3. Se analiza si hay que mandar cargas a un puerto o a los dos.
4. Se le manda un mensaje correspondiente a cada puerto de que cargas van para allá con su capacidad.
5. Cada puerto almacena sus cargas correspondientes
6. Fin de la transacción.



1.4

Si se quiere implementar el requerimiento RF14 con un protocolo de colas de mensaje, La solicitud de descargue es mandada a un puerto (Puerto Principal), acto seguido el puerto tiene que enviar un mensaje por medio de la cola a los otros puertos pidiendo la cantidad de almacenamiento que poseen, estos le envían la información a través de la cola de mensajes. Después el puerto principal se verifica que la suma de las capacidades de los 3 puertos sean puedan almacenar las cargas. En caso negativo el todos los puertos hacen rollback. En caso positivo el PuertoPrincipal les envía un arreglo por medio de la cola a cada puerto de las cargas que tienen que almacenar. Una vez almacenadas todas las cargas se hace commit.

En caso contrario con two phase commit, cuando se envía la solicitud de descargue a la aplicación y se evalúa cuantos puertos son necesarios para almacenar la carga, el manejador le notifica al primer puerto y ese responde si acepta o rechaza la solicitud. Si se necesita más de un puerto y ya uno había dicho que si, pero el siguiente respondió de forma negativa, sería necesario hacer un RollBack para deshacer el commit hecho por la primera transacción. Para esto sería importante crear un checkpoint al inicio de cada transacción para poder devolverse a un estado correcto de la base de datos.

La ventaja de usar colas de mensaje, es que no sería necesario usar checkpoints, ya que la transacción completa no se le hace commit hasta que todas las solicitudes sean aceptadas, por otro lado, al usar two phase commit, se le hace commit a cada solicitud aceptada por cada puerto, por lo que también podría llevar a inconsistencias en la base de datos. Por este motivo, se considera mejor usar el protocolo de colas de mensaje para implementar este requerimiento.