



UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE INGENIERÍA  
Año 2015 - 2<sup>do</sup> Cuatrimestre

## **TALLER DE PROGRAMACIÓN III (75.61)**

TRABAJO PRÁCTICO N.º 2  
TEMA: Colas en RabbitMQ  
FECHA: 05/10/2015

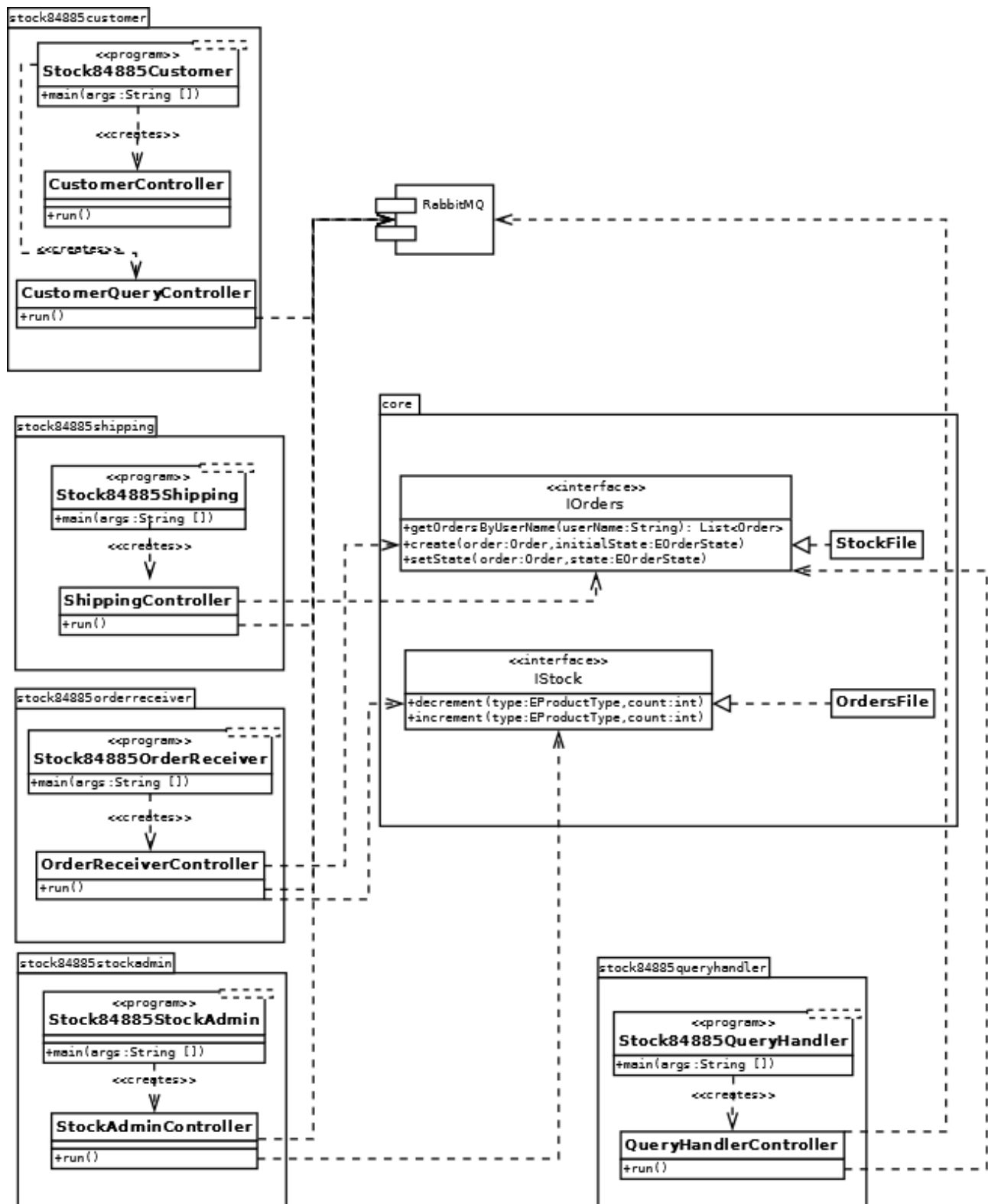
Autor  
Darío Eduardo Ramos

Padrón  
84885

## **Objetivos del sistema**

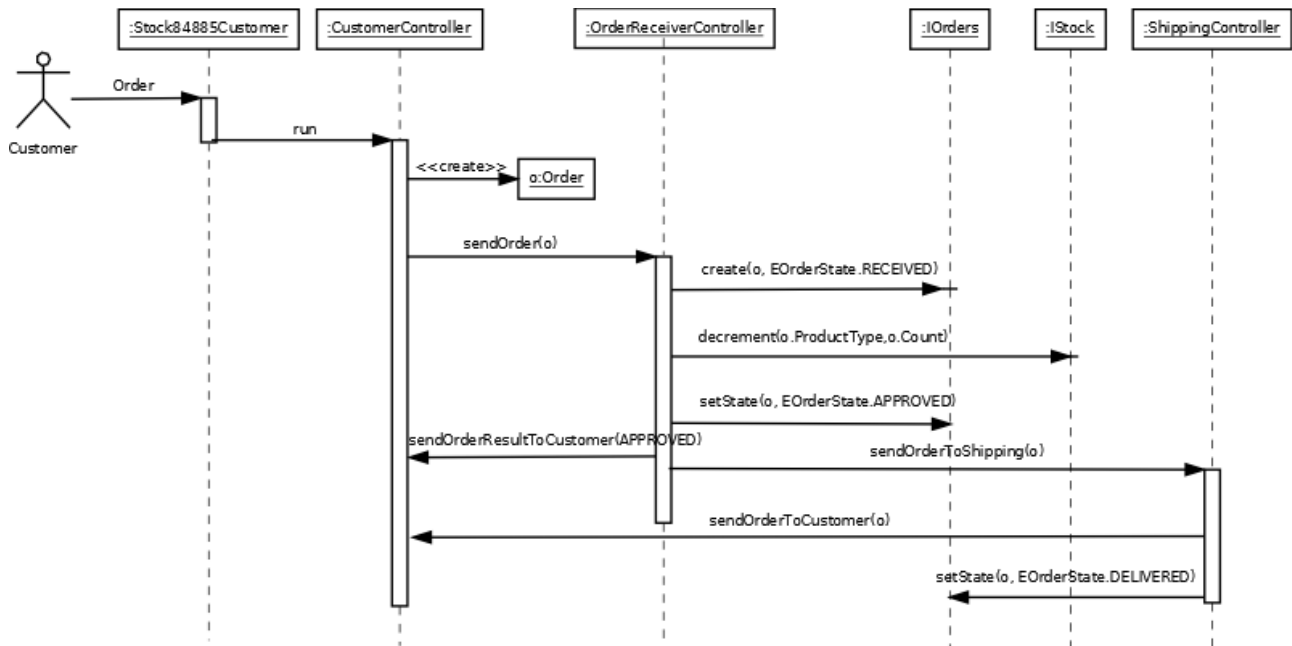
- Tomar pedidos de una cantidad masiva de usuarios simultáneos (escalabilidad).
- Mantener un control de stock consistente (siempre mayor a cero) en todo momento.
- Diseñar los componentes de manera que puedan ser fácilmente distribuidos.
- Mantener un control detallado de todos los pedidos, con fines de auditoría.
- Utilizar un patrón de comunicaciones persistentes para garantizar el envío de los mensajes.

## Vista lógica: Diagrama de paquetes



Todos los programas usan RabbitMQ, y clases del Core que no se muestran por simplicidad (como ILogger y Configuration). Se usan interfaces para IOrders e IStock a fin de poder reemplazar fácilmente los archivos por bases de datos.

## Vista de procesos: Diagrama de secuencia



Este diagrama corresponde al escenario más común: un cliente hace un pedido, hay stock suficiente y se le entrega el pedido sin problemas. Se omitieron los mecanismos de comunicación por simplicidad; los mismos pueden verse en la vista de despliegue (diagrama de robustez).

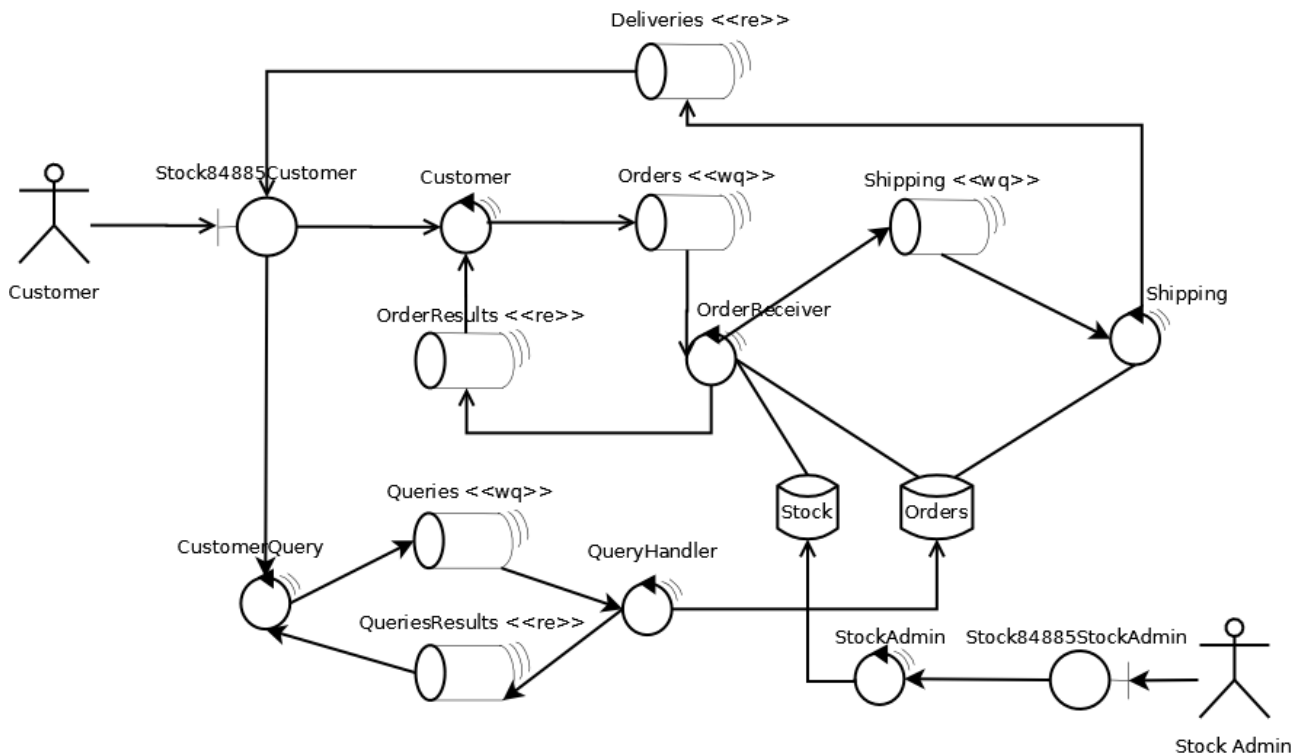
El cliente (Customer) manda su pedido a través de un objeto de borde de tipo **Stock84885Customer**, quien invoca a **CustomerController**. Éste instancia una **Order**, o, con los datos del pedido, y se la envía a **OrderReceiverController**. El OrderReceiver registra la orden con estado inicial RECEIVED. Luego, intenta decrementar el stock según el pedido. En este escenario, lo logra, y por ende cambia el estado del pedido a APPROVED, notifica al cliente que su pedido fue aprobado y envía la orden a un **ShippingController**.

En Shipping, la orden es despachada al cliente que la realizó, se marca la orden como enviada (DELIVERED), y se cierra el circuito.

El diagrama para el caso en que el pedido es rechazado sería similar, con la diferencia de que se marca la orden como rechazada (REJECTED), se notifica al cliente y se corta el circuito allí.

El acceso a IOrders e IStock es en exclusión mutua para toda objeto que los acceda, sea para leer o escribir.

## Vista de despliegue: Diagrama de robustez



Estereotipos:

- re: Routing Exchange. Usa routing para enviar el mensaje a un receptor específico. Por ejemplo, Shipping lo usa para enviarle el pedido al cliente que lo pidió y a ningún otro.
- wq: Workers Queue. El productor pone el mensaje en el exchange, y lo toma cualquier consumidor (se usa el esquema por defecto, o sea Round Robin). Por ejemplo, lo usa el Customer para enviar su pedido y que lo tome cualquier OrderReceiver.

Sobre el diagrama:

Un cliente (**Customer**) envía un pedido (order) por la cola **Orders**, de donde la toma algún **OrderReceiver**. El mismo crea una instancia de Order y la guarda en **Orders** con estado inicial recibido. Luego, revisa el **Stock**, garantizando exclusión mutua, y le responde al **Customer** si el pedido fue rechazado porque no había stock, o aprobado. En ambos casos, se actualiza el estado del pedido.

Si el pedido es aprobado, es enviado a la cola **Shipping**, de donde la toma algún **ShippingController** y la envía por el exchange **Deliveries** al cliente que la pidió. Shipping sabe quién es ese cliente porque dicha información llegó en el pedido que recibió.

Por otro lado, un Customer puede consultar el estado de sus pedidos. Lo hace a través de **CustomerQuery**, quien envía el nombre del cliente por **Queries**, de donde lo toma **QueryHandler**. Éste revisa Orders, y manda la lista de pedidos con su estado al cliente vía **QueriesResults**.

Por último, el actor StockAdmin, a través del **controller homónimo**, puede consultar el stock de cualquier producto en cualquier momento, con exclusión mutua garantizada.

## Vista de casos de uso

<b>Caso de uso</b>	Realizar pedido
<b>Actores</b>	Primario: Cliente (Customer)
<b>Pre-condiciones</b>	Stock definido y válido
<b>Descripción</b>	<p>Flujo principal</p> <ol style="list-style-type: none"> <li>1. Cliente envía su pedido especificando tipo de producto y cantidad.</li> <li>2. Sistema registra pedido como recibido.</li> <li>3. Sistema consulta stock de producto y cantidad solicitados.</li> <li>4. Sistema marca pedido como aprobado.</li> <li>5. Sistema responde a Cliente que su pedido fue aprobado.</li> <li>6. Sistema envía pedido a Shipping</li> <li>7. Sistema envía el pedido desde Shipping a Cliente.</li> </ol> <p>Flujo alternativo A1 (si no hay stock para el pedido)</p> <p>A1.1. Sistema marca pedido como rechazado.</p> <p>A1.2. Sistema responde a Cliente que su pedido fue rechazado.</p>
<b>Post-condiciones</b>	<p>Stock actualizado (decrementado)</p> <p>Pedido registrado en el sistema como rechazado o enviado</p>

<b>Caso de uso</b>	Realizar consulta de estado de pedido
<b>Actores</b>	Primario: Cliente
<b>Pre-condiciones</b>	Registro de pedidos activo y válido
<b>Descripción</b>	<p>Flujo principal:</p> <ol style="list-style-type: none"> <li>1. Cliente solicita a Sistema el estado de su(s) pedido(s).</li> <li>2. Sistema devuelve el listado de todos los pedidos realizados por Cliente, incluyendo su estado actual al momento de la consulta.</li> </ol> <p>Flujo alternativo A1:</p> <p>(si el cliente nunca hizo un pedido)</p> <p>A1.1. Sistema responde que no hay pedidos registrados</p>
<b>Post-condiciones</b>	<p>El Cliente tiene la información solicitada.</p> <p>No hay cambios en el Sistema.</p>

<b>Caso de uso</b>	Modificar stock
<b>Actores</b>	Primario: Administrador de Stock (StockAdmin)
<b>Pre-condiciones</b>	Registro de stock válido
<b>Descripción</b>	<p>Flujo principal:</p> <ol style="list-style-type: none"> <li>1. Admin solicita un cambio en el stock, especificando tipo de producto y variación en stock (positiva o negativa).</li> <li>2. Sistema realiza modificación.</li> </ol> <p>Flujo alternativo A1:</p> <p>(no se puede hacer el cambio porque daría stock negativo o por encima del máximo)</p> <p>A1.1. Sistema informa a Admin que no se puede efectuar el cambio</p>
<b>Post-condiciones</b>	Stock actualizado.