

## Trabalho de comunicação indireta Em dupla

### 1. Introdução

Middlewares orientado a mensagens (MOM – *Message Oriented Middlewares*) são sistemas que permitem o envio de mensagens entre entidades de um sistema distribuído. São uma forma de comunicação indireta que provê um serviço de comunicação baseado em filas de mensagens, promovendo o desacoplamento de tempo e espaço. A Seção 6.4 da 5ª edição do livro *Distributed Systems*, do Coulouris (2013), descreve o modelo de programação de MOM.

O *Java Message Service* – JMS e o *Advanced Message Queuing Protocol* – AMQP são APIs que especificam modelos de *middleware* orientado a mensagens. Esses *middlewares* suportam pelo menos dois modelos básicos de troca de mensagens: ponto a ponto e o modelo *publish/subscribe*. Para que uma aplicação possa utilizar o *middleware*, deve haver um provedor que possa gerenciar as sessões e filas. Existem opções comerciais e livres de sistemas MOM. Dentre as opções FOSS tem-se: ActiveMQ, JBossMQ, HornetQ, Joram, MantaRay, OpenJMS, RabbitMQ.

### 2. Especificação

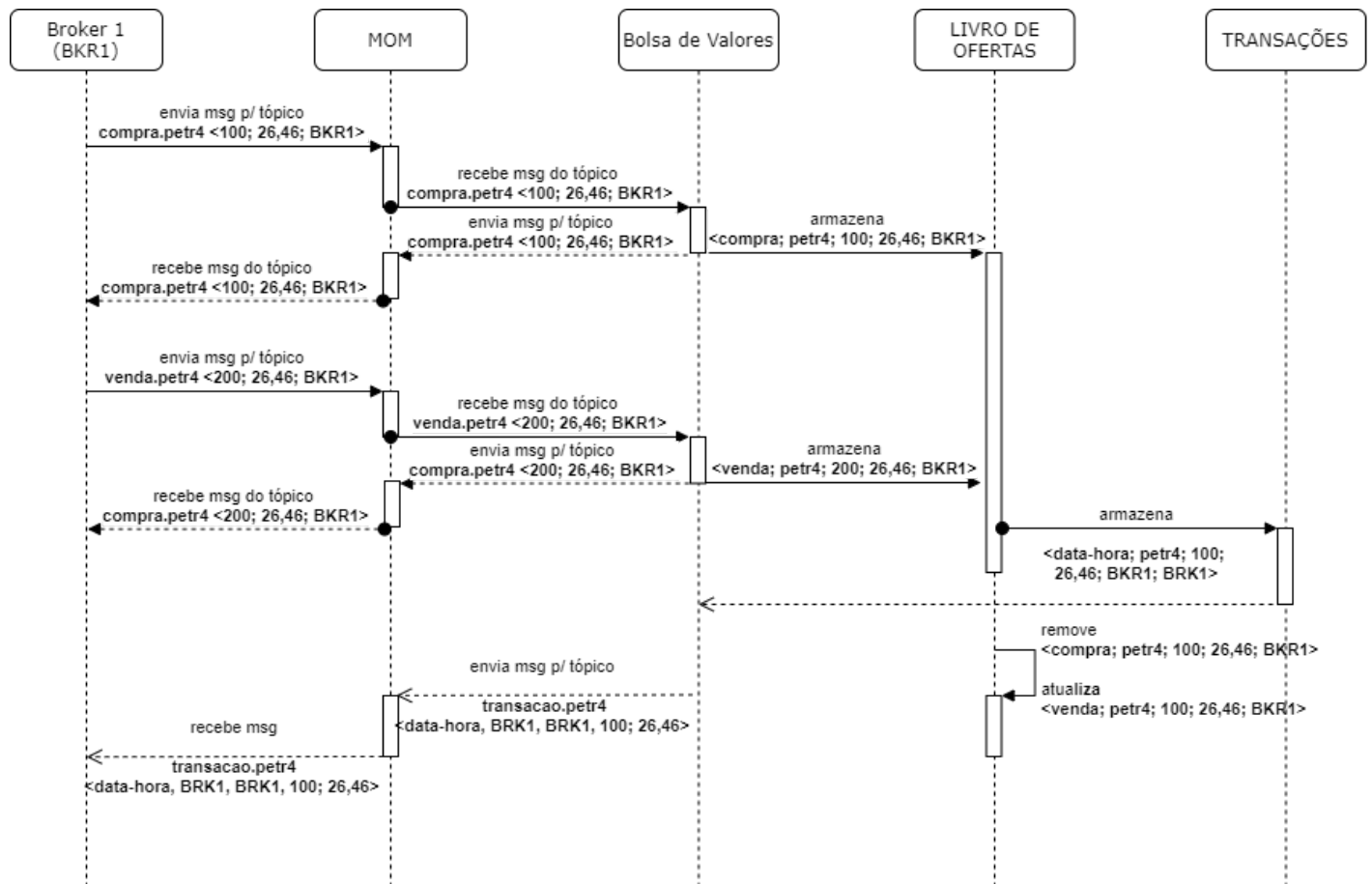
Neste trabalho deverá ser desenvolvido um sistema para uma bolsa de valores qualquer, como a Bovespa, utilizando o RabbitMQ. Quem quiser conhecer um pouco mais sobre o funcionamento da bolsa de valores, visite o site da BOVESPA em [http://www.b3.com.br/pt\\_br/market-data-e-indices/servicos-de-dados/market-data/cotacoes/](http://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/cotacoes/).

A corretora (ou *broker*) pode enviar as seguintes operações à bolsa de valores:

Mensagens da Corretora → Bolsa de valores		
OPERAÇÃO	FORMATO DA MENSAGEM	DESCRIÇÃO
compra	<quant: int, val: real, corretora: char[4]>	Envia à bolsa de valores uma ordem de compra com o tópico <b>compra.ativo</b> indicando que a corretora deseja comprar <i>quant</i> lotes de ações de um ativo pelo preço de <i>val</i> reais.
venda	<quant: int, val: real, corretora: char[4]>	Envia à bolsa de valores uma ordem de venda com o tópico <b>compra.ativo</b> indicando que a corretora deseja vender <i>quant</i> lotes de ações de um ativo pelo preço de <i>val</i> reais.

Mensagens da Bolsa de valores → Corretora		
OPERAÇÃO	FORMATO DA MENSAGEM	DESCRIÇÃO
compra	<quant: int, val: real, corretora: char[4]>	Envia ao tópico <b>compra.ativo</b> uma mensagem notificando que a bolsa de valores recebeu uma ordem de compra.
venda	<quant: int, val: real, corretora: char[4]>	Envia ao tópico <b>venda.ativo</b> uma mensagem notificando que a bolsa de valores recebeu uma ordem de compra.
transacao	<data-hora: "dd/mm/aaaa hh:mm", corr_vd: char[4], corr_cp: char[4], quant: int, val: real>	Envia ao tópico <b>transacao.ativo</b> uma mensagem notificando que a bolsa de valores realizou uma transação de compra e venda.

O diagrama de sequência a seguir ilustra um cenário de troca de mensagens entre uma Corretora e a Bolsa de valores. Linhas contínuas representam mensagens enviadas na estrutura de tópicos da Corretora para a Bolsa de Valores, enquanto as linhas tracejadas representam as mensagens enviadas da Bolsa de valores para as corretoras.



As regras de negócio do sistema podem ser descritas da seguinte forma:

- Brokers podem enviar ORDENS DE COMPRA, ORDENS DE VENDA para a bolsa de valores.
- Brokers podem assinar tópicos relativos aos ativos que desejam acompanhar.
- Sempre que a bolsa de valores recebe uma ordem de compra ou de venda, ela deve encaminhar essa ordem a todos os brokers interessados naquela ação específica através de um mecanismo de tópicos.
- Sempre que o valor de uma ORDEM DE COMPRA for maior ou igual ao valor de uma ORDEM DE VENDA para um mesmo ativo, a bolsa de valores deve gerar uma mensagem do tipo TRANSAÇÃO no tópico adequado, e atualizar/remover as ordens da fila.
- Os brokers deverão encaminhar mensagens para uma
- A bolsa de valores e os brokers deverão usar uma estrutura de tópicos, do tipo: *operacao.ativo*.

### 3. Atividades

O sistema deve ser carregado com a lista de ativos da Bovespa. A tabela a seguir ilustra alguns exemplos de ativos da Bovespa.



## Lista de ativos resumida

NOME DE PREGÃO	CÓDIGO	ATIVIDADE PRINCIPAL
AMBEV S/A ON	ABEV3	Fabricação e Distribuição de Cervejas. Refrigerantes e Bebidas Não Carbonatadas e Não Alcoólicas.
PETROBRAS PN	PETR4	Petróleo. Gás e Energia
VALE PNA	VALE5	Mineração
ITAUNIBANCO PN	ITUB4	A Sociedade Tem por Objeto A Atividade Bancária.
BRABESCO PN	BBDC4	Prática de Operações Bancárias em Geral. inclusive Câmbio
BRASIL ON	BBAS3	Banco Múltiplo.
CIELO ON	CIEL3	Empresa Prestadora de Serviços de Adquirência e Meios de Pagamento.
PETROBRAS ON	PETR3	Petróleo. Gás e Energia.
HYPERMARCAS ON	HYPE3	Produção e Venda de Bens de Consumo e Medicamentos.
VALE ON	VALE3	Mineração
BBSEGURIDADE ON	BBSE3	Participação no Capital Social de Outras Sociedades. que Tenham por Atividade Operações de Seguros. Resseguros. Previdência Complementar ou Capitalização.
CETIP ON	CTIP3	Sociedade Administradora de Mercados de Balcão Organizados.
GERDAU PN	GGBR4	Participação e Administração.
FIBRIA ON	FIBR3	
RAIADROGASIL ON	RADL3	Comércio de Produtos Farmacêuticos. Perfumarias e Afins.

Inicialmente, o grupo deverá:

1. Instalar o RabbitMQ, disponível em <https://www.rabbitmq.com>
2. Executar os tutoriais 1 a 5.

O trabalho consiste em desenvolver um pequeno aplicativo para o Broker e outro aplicativo para a Bolsa de valores, utilizando filas de mensagens e estruturas de tópicos. Os requisitos do trabalho são:

1. O endereço do RabbitMQ server deve ser passado como parâmetro para que brokers e bolsas possam escolher a quem se conectar.
2. A bolsa deve abrir um canal do tipo pub/sub utilizando tópicos para publicar as atualizações no livro de ofertas e as operações realizadas em uma ação. O nome do canal deve ser BOLSADEVALORES.
3. O servidor abre uma fila de mensagens para receber as operações dos clientes. O nome da fila de mensagens deve ser BROKER.
4. Os clientes enviam operações para o servidor através da fila de mensagens BROKER.
5. Todos os clientes devem receber a notificação das operações através da fila BOLSADEVALORES.
6. O servidor deverá ser disponibilizado em uma máquina diferente de localhost.
7. O aplicativo deve funcionar nas máquinas Linux do laboratório de redes do curso de Engenharia de Software da PUC Minas.

Observações:



- Será necessário utilizar Threads em Java.
- O Tutorial 1 do RabbitMQ ensina como criar o canal para o recebimento da mensagem pelo servidor.
- O Tutorial 3 do RabbitMQ ensina como criar a exchange box para o Publish/Subscribe.
- O Tutorial 5 do RabbitMQ ensina como utilizar uma estrutura de tópicos.

## 4. Resultados Esperados

Deverá ser entregue:

- Todo o código, comentado.
- Um arquivo README.TXT contendo uma explicação sucinta do código e instruções para compilação e execução do programa. Preferencialmente, criar scripts que permitam executar o programa automaticamente.
- Uma documentação em PDF descrevendo a estrutura do sistema: classes, operações, etc.
- É obrigatória a apresentação de um diagrama de classes no formato UML para ilustrar o sistema e um diagrama de componentes para ilustrar a arquitetura