

**CENTRO UNIVERSITÁRIO FEI**

**JOÃO PEDRO ROSA CEZARINO**

**HUGO LINHARES OLIVEIRA**

**VITOR MARTINS OLIVEIRA**

**THALES DE OLIVEIRA LACERDA**

**BV - SIMULATION: RELATÓRIO DE PROJETO**

São Bernardo do Campo

2023

## **ABSTRACT**

Este relatório detalha a implementação de um sistema automatizado para negociação de ações, englobando três componentes: a Bolsa de Valores (BV), o Home Broker (HB) e os Robôs de Negociação. Esses componentes interagem para simular a dinâmica de um mercado financeiro, com os robôs executando transações de forma autônoma. O relatório descreve a uma breve introdução ao projeto, a estrutura utilizada, imagens de alguns testes realizados e uma conclusão para fechamento da simulação.

**Keywords:** Negociação automatizada, Bolsa de Valores, Home Broker, Robôs de Negociação, RabbitMQ, Python, Sincronização de componentes, Logging, Gestão de erros

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>4</b>
<b>2</b>	<b>ESTRUTURA</b>	<b>5</b>
2.1	BV - Bolsa de Valores	5
2.2	HB - Home Broker	5
2.3	RB - Robôs de Negociação	6
2.4	Comunicação	6
2.5	Erros e Logs	6
2.6	Automatização	6
2.7	Sincronização	7
<b>3</b>	<b>TESTES</b>	<b>8</b>
<b>4</b>	<b>CONCLUSÃO</b>	<b>12</b>

## 1 INTRODUÇÃO

Este relatório descreve o projeto de implementação do Algoritmo de Berkeley, um algoritmo que permite a sincronização de diversos computadores em um sistema distribuído. O algoritmo de Berkeley é amplamente reconhecido por sua eficácia na sincronização de relógios em sistemas distribuídos.

Neste projeto, o algoritmo é aplicado a um sistema distribuído responsável pelo controle de compra/venda de ações em uma bolsa de valores (BV). A BV detém uma lista de ações, a quantidade dessas ações disponíveis para compra/venda e o valor atual de cada ação. As informações são propagadas para as instituições intermediárias e seus sistemas de home-brokers (HB) sempre que a quantidade ou o preço de uma ação é modificado.

Os sistemas de HB são responsáveis pelo envio dos pedidos de compra/venda para a BV. Ao analisar o pedido de compra/venda do HB e a hora em que ele foi emitido, a BV pode decidir por acatar o pedido ou pode decidir por iniciar um processo de sincronização usando o algoritmo de Berkeley.

Os robôs no sistema desempenham o papel de negociadores autônomos, cada um operando com um conjunto de ações e tendo a capacidade de executar operações de compra e venda. Eles operam por meio de solicitações aleatórias de transações, escolhendo aleatoriamente uma ação do seu conjunto, decidindo aleatoriamente se irão comprar ou vender, e selecionando uma quantidade aleatória para a transação. Eles também são responsáveis por monitorar suas próprias ações, fazendo solicitações periódicas para os Homebrokers, para manter suas listas de ações atualizadas. É importante ressaltar que cada robô realiza transações de forma independente e paralela aos outros.

Este relatório detalhará o processo de implementação do projeto, incluindo os desafios encontrados, as soluções implementadas e os resultados obtidos.

O projeto encontra-se disponível no repositório a seguir: <<https://github.com/akajhon/BV-Simulation>>

## 2 ESTRUTURA

Este projeto consiste em uma arquitetura distribuída, que inclui três componentes principais que interagem entre si através do protocolo de mensagens AMQP, implementado usando RabbitMQ. A seguir, descreveremos em detalhes cada componente e o papel que desempenham no sistema.

O sistema é organizado em uma estrutura de containeres Docker, o que proporciona isolamento e independência para cada componente, além de permitir fácil escalabilidade e implantação. A estrutura de containers é gerenciada através do “Docker Compose”, que automatiza o processo de criação e gerenciamento de containers. Isso facilita a configuração do sistema, permitindo a criação de múltiplos contêiners com apenas um comando. A configuração do “Docker Compose” também facilita a ligação entre os contêiners, permitindo que eles se comuniquem entre si com facilidade.

### 2.1 BV - BOLSA DE VALORES

A Bolsa de Valores (BV) é o componente central do sistema, encarregado de gerenciar e atualizar a lista de ações disponíveis e seus respectivos valores. É aqui que as ordens de compra e venda são recebidas e processadas. Para cada ação, a BV mantém um registro do número total de ações disponíveis, o número de ações disponíveis para venda e o preço atual, o que é importante para manter a lógica de negociação e evitar a venda de ações que não foram compradas.

A função “processar\_pedido” é o núcleo da lógica da BV. Quando um pedido de compra ou venda é recebido de um HB, essa função é chamada para processar o pedido. A função verifica a operação solicitada (compra ou venda) e realiza a operação correspondente nas ações especificadas. Além disso, a função verifica se o pedido é válido - por exemplo, se existem ações suficientes disponíveis para venda.

### 2.2 HB - HOME BROKER

O Home Broker (HB) atua como intermediário entre a Bolsa de Valores (BV) e os Robôs. Ele é responsável por encaminhar as ordens de compra e venda dos robôs para a BV e enviar as

atualizações da lista de ações para os robôs. O HB mantém um registro dos pedidos pendentes e assegura que cada robô receba uma resposta a cada pedido enviado.

### 2.3 RB - ROBÔS DE NEGOCIAÇÃO

Os Robôs de Negociação são os agentes ativos no sistema. Eles enviam ordens de compra e venda para a BV através do HB, baseando suas decisões em uma estratégia de negociação simples que consiste em escolher aleatoriamente uma ação e uma operação (compra ou venda) e uma quantidade para a operação a ser realizada.

Cada robô tem uma função “realizar\_operacao”, que contém toda essa lógica. Quando uma operação é escolhida, o robô envia um pedido ao HB através de uma fila de mensagens RabbitMQ. O robô então espera receber a lista atualizada de ações do HB para prosseguir com a próxima operação.

### 2.4 COMUNICAÇÃO

A comunicação entre os componentes é feita através de um broker de mensagens RabbitMQ. O RabbitMQ permite a comunicação assíncrona e robusta entre os componentes, com suporte para filas de mensagens, tópicos e outros recursos de gerenciamento de mensagens. Neste projeto, cada componente é conectado ao broker RabbitMQ e publica ou consome mensagens de uma ou mais filas ou tópicos.

### 2.5 ERROS E LOGS

As mensagens de log são coloridas de acordo com seu tipo (informação, sucesso, sincronização ou erro), facilitando a leitura e a interpretação das mensagens. Além disso, estruturas do tipo “try/catch” foram implementadas visando o tratamento de possíveis erros e bugs ao longo da execução do código.

### 2.6 AUTOMATIZAÇÃO

Visando facilitar a implantação e o gerenciamento do sistema, um script shell foi desenvolvido para automatizar as tarefas envolvidas na simulação. Ele possui funções para construir e iniciar os serviços Docker (start\_services), parar os serviços Docker (stop\_services),

visualizar logs dos serviços em execução (view\_logs) e verificar/instalar o Docker e Docker Compose, caso não estejam presentes do sistema em que a simulação será executada (install\_requirements).

O script pode ser executado com quatro parâmetros diferentes: “start”, “stop”, “requirements” e “logs”. Cada parâmetro desencadeia a execução de uma função específica. Por exemplo, se o script for executado com o parâmetro “start”, ele exibirá um banner, construirá e iniciará os contêineres Docker, e acompanhará os logs de todos os serviços em execução.

Ao usar este script, os usuários podem iniciar e monitorar o sistema facilmente, o que facilita a implantação e o desenvolvimento do projeto como um todo

## 2.7 SINCRONIZAÇÃO

A sincronização do relógio é fundamental em sistemas distribuídos para garantir a ordenação correta das operações e para evitar problemas como a inconsistência de dados devido ao desalinhamento de tempo. A função “sincronizar\_relogio” é responsável pela sincronização do relógio da BV com o relógio do HB envolvido no pedido. A função recebe o tempo atual do BV e o tempo recebido do HB, calcula a média desses dois valores e ajusta o relógio do BV e do HB com esse valor médio. Após a sincronização, todas as operações futuras realizadas pelo HB serão registradas com o novo tempo, garantindo que as transações estejam corretamente alinhadas com o tempo da Bolsa de Valores.

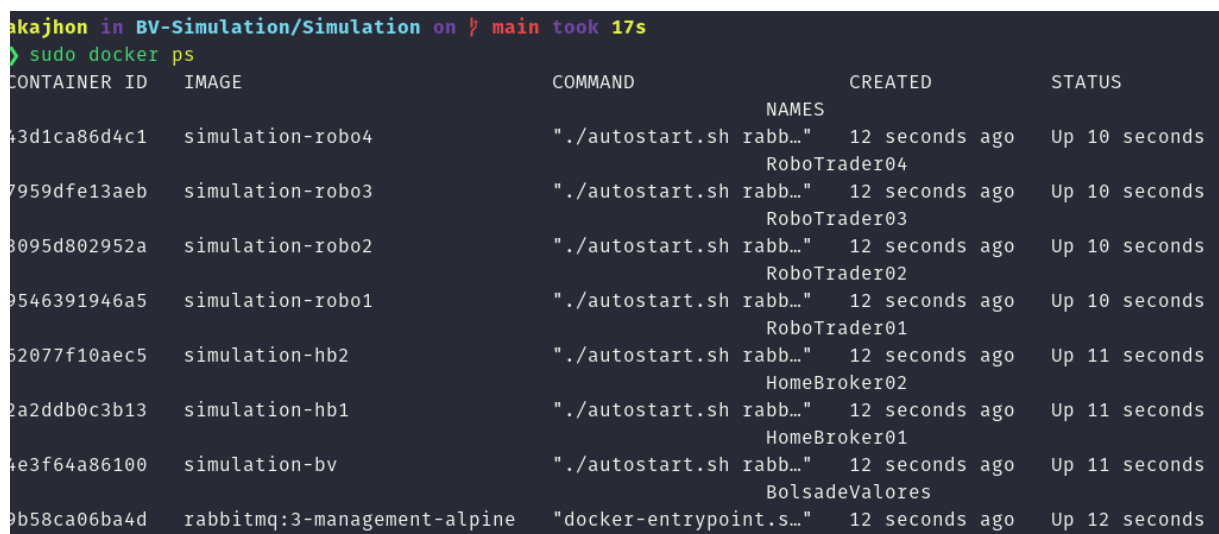
### 3 TESTES

Este capítulo apresenta os testes realizados no sistema. Serão apresentados os logs das operações de compra de ações, o recebimento da lista de ações pelos robôs, erros relacionados à compra de quantidades maiores que as disponíveis, a inicialização do RabbitMQ e o processo de sincronização entre a Bolsa de Valores (BV) e o Home Broker (HB).

Os testes foram realizados com o objetivo de verificar o correto funcionamento do sistema em diferentes cenários e identificar possíveis problemas ou falhas. Foram gerados logs detalhados que permitem uma análise minuciosa do fluxo de execução e das interações entre os componentes.

Para evidenciar a execução em ambientes completamente independentes, segue abaixo uma a saída do comando “docker ps” após o início da simulação. Nela fica evidente a presença de todos os Contêineres descritos no arquivo docker-“compose”.

Figura 1 – Contêineres Docker



```

akajhon in BV-Simulation/Simulation on 7 main took 17s
> sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
43d1ca86d4c1   simulation-robo4                     "/autostart.sh rabb..." 12 seconds ago Up 10 seconds
7959dfe13aeb   simulation-robo3                     "/autostart.sh rabb..." 12 seconds ago Up 10 seconds
8095d802952a   simulation-robo2                     "/autostart.sh rabb..." 12 seconds ago Up 10 seconds
9546391946a5   simulation-robo1                     "/autostart.sh rabb..." 12 seconds ago Up 10 seconds
62077f10aec5   simulation-hb2                       "/autostart.sh rabb..." 12 seconds ago Up 11 seconds
2a2ddb0c3b13   simulation-hb1                       "/autostart.sh rabb..." 12 seconds ago Up 11 seconds
4e3f64a86100   simulation-bv                         "/autostart.sh rabb..." 12 seconds ago Up 11 seconds
9b58ca06ba4d   rabbitmq:3-management-alpine        "docker-entrypoint.s..." 12 seconds ago Up 12 seconds
  
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
43d1ca86d4c1	simulation-robo4	"/autostart.sh rabb..."	12 seconds ago	Up 10 seconds
7959dfe13aeb	simulation-robo3	"/autostart.sh rabb..."	12 seconds ago	Up 10 seconds
8095d802952a	simulation-robo2	"/autostart.sh rabb..."	12 seconds ago	Up 10 seconds
9546391946a5	simulation-robo1	"/autostart.sh rabb..."	12 seconds ago	Up 10 seconds
62077f10aec5	simulation-hb2	"/autostart.sh rabb..."	12 seconds ago	Up 11 seconds
2a2ddb0c3b13	simulation-hb1	"/autostart.sh rabb..."	12 seconds ago	Up 11 seconds
4e3f64a86100	simulation-bv	"/autostart.sh rabb..."	12 seconds ago	Up 11 seconds
9b58ca06ba4d	rabbitmq:3-management-alpine	"docker-entrypoint.s..."	12 seconds ago	Up 12 seconds

No primeiro teste, são apresentados os logs das operações de compra de ações realizadas pelos robôs. É possível observar as requisições feitas pelos robôs, os pedidos enviados ao HB e as respostas recebidas da BV.



Figura 2 – Compra de Ações

```

RoboTrader01 | 19:55:52 - INFO - [$] Pedido de compra de 63 ACA07 encaminhado ao HB1 com sucesso !
RoboTrader01 | 19:55:52 - INFO - [*] Robo1 solicitando lista de ações ao HB1...
HomeBroker01 | 19:55:52 - INFO - [$] Pedido de compra de 63 ACA07, realizado por robo1, encaminhado a BV com sucesso!
HomeBroker01 | 19:55:52 - INFO - [*] robo1 solicitou a lista de ações .
HomeBroker01 | 19:55:52 - INFO - [*] HB1 solicitando lista de ações a BV...
BolsadeValores | 19:55:52 - INFO - [$] Pedido de compra de 63 ACA07, realizado por hb1, processado com sucesso!
BolsadeValores | 19:55:52 - INFO - [*] Lista de ações enviada da BV ao hb1
HomeBroker01 | 19:55:52 - INFO - [*] Lista de ações recebida da BV no HB1 !
HomeBroker01 | 19:55:52 - INFO - [*] Lista de ações enviada pelo HB1 ao robo1 !
RoboTrader01 | 19:55:52 - INFO - [*] Lista de ações recebida no Robo1 !
RoboTrader01 | 19:55:52 - INFO - [*] Ações recebidas de hb1:
RoboTrader01 | 19:55:52 - INFO - [$] ACA01: {'quantidade': 69, 'valor': 18.87, 'disponivel_para_venda': 0}
RoboTrader01 | 19:55:52 - INFO - [$] ACA02: {'quantidade': 79, 'valor': 81.43, 'disponivel_para_venda': 0}
RoboTrader01 | 19:55:52 - INFO - [$] ACA03: {'quantidade': 98, 'valor': 89.93, 'disponivel_para_venda': 0}
RoboTrader01 | 19:55:52 - INFO - [$] ACA04: {'quantidade': 70, 'valor': 17.47, 'disponivel_para_venda': 0}
RoboTrader01 | 19:55:52 - INFO - [$] ACA05: {'quantidade': 98, 'valor': 37.03, 'disponivel_para_venda': 0}
RoboTrader01 | 19:55:52 - INFO - [$] ACA06: {'quantidade': 82, 'valor': 80.73, 'disponivel_para_venda': 0}
RoboTrader01 | 19:55:52 - INFO - [$] ACA07: {'quantidade': 18, 'valor': 98.09, 'disponivel_para_venda': 63}
RoboTrader01 | 19:55:52 - INFO - [$] ACA08: {'quantidade': 76, 'valor': 59.63, 'disponivel_para_venda': 0}

```

Em seguida, são exibidos os logs que demonstram o recebimento da lista de ações pelos robôs. Essa lista é enviada pela BV e repassada pelo HB e contém as informações atualizadas sobre as ações disponíveis para compra/venda.

Figura 3 – Lista de Ações

```

HomeBroker02 | [!] RabbitMQ is up -> Executing Command... [!]
HomeBroker02 | 19:55:42 - INFO - [...] HB2 aguardando mensagens...
RoboTrader04 | 19:55:47 - INFO - [*] Robo4 solicitando lista de ações ao HB2...
RoboTrader04 | 19:55:47 - INFO - [...] Robo4 aguardando para enviar mensagens...
HomeBroker02 | 19:55:47 - INFO - [*] robo4 solicitou a lista de ações .
HomeBroker02 | 19:55:47 - INFO - [*] HB2 solicitando lista de ações a BV...
BolsadeValores | 19:55:47 - INFO - [*] Lista de ações enviada da BV ao hb2
HomeBroker02 | 19:55:47 - INFO - [*] Lista de ações recebida da BV no HB2 !
HomeBroker02 | 19:55:47 - INFO - [*] Lista de ações enviada pelo HB2 ao robo4 !
RoboTrader04 | 19:55:47 - INFO - [*] Lista de ações recebida no Robo4 !
RoboTrader04 | 19:55:47 - INFO - [*] Ações recebidas de hb2:
RoboTrader04 | 19:55:47 - INFO - [$] ACA01: {'quantidade': 69, 'valor': 18.87, 'disponivel_para_venda': 0}
RoboTrader04 | 19:55:47 - INFO - [$] ACA02: {'quantidade': 79, 'valor': 81.43, 'disponivel_para_venda': 0}
RoboTrader04 | 19:55:47 - INFO - [$] ACA03: {'quantidade': 98, 'valor': 89.93, 'disponivel_para_venda': 0}
RoboTrader04 | 19:55:47 - INFO - [$] ACA04: {'quantidade': 70, 'valor': 17.47, 'disponivel_para_venda': 0}
RoboTrader04 | 19:55:47 - INFO - [$] ACA05: {'quantidade': 98, 'valor': 37.03, 'disponivel_para_venda': 0}
RoboTrader04 | 19:55:47 - INFO - [$] ACA06: {'quantidade': 82, 'valor': 80.73, 'disponivel_para_venda': 0}
RoboTrader04 | 19:55:47 - INFO - [$] ACA07: {'quantidade': 81, 'valor': 97.12, 'disponivel_para_venda': 0}
RoboTrader04 | 19:55:47 - INFO - [$] ACA08: {'quantidade': 76, 'valor': 59.63, 'disponivel_para_venda': 0}
RoboTrader01 | 19:55:47 - INFO - [*] Robo1 solicitando lista de ações ao HB1...
RoboTrader01 | 19:55:47 - INFO - [...] Robo1 aguardando para enviar mensagens...

```

Outro aspecto abordado nos testes é a detecção e tratamento de erros relacionados à compra de quantidades maiores do que as disponíveis. Os logs mostram casos em que os robôs tentam comprar uma quantidade superior ao estoque disponível na BV. Essas situações são identificadas e tratadas pelo sistema, sendo registradas nos logs para análise posterior.

Figura 4 – Erro de Compra

```

HomeBroker02 | 19:55:57 - INFO - [$] Pedido de compra de 49 ACA07, realizado por robo4, encaminhado a BV com sucesso!
HomeBroker02 | 19:55:57 - INFO - [*] robo4 solicitou a lista de ações .
HomeBroker02 | 19:55:57 - INFO - [*] HB2 solicitando lista de ações a BV...
BolsadeValores | 19:55:57 - INFO - [$] A quantidade a ser comprada 49 é maior do que a quantidade disponível 18 para ACA07 !

```

Além disso, os logs referentes à inicialização do RabbitMQ e ao aguardo de mensagens pelos componentes seguem abaixo. Esses logs são úteis para verificar se a comunicação assíncrona entre os componentes está ocorrendo corretamente e se os pedidos e atualizações estão sendo transmitidos de forma eficiente.

Figura 5 – RabbitMQ

```

BolsadeValores | [...] RabbitMQ is unavailable -> Sleeping... [!]
RoboTrader02   | [...] RabbitMQ is unavailable -> Sleeping... [!]
HomeBroker01   | [...] RabbitMQ is unavailable -> Sleeping... [!]
RoboTrader03   | [...] RabbitMQ is unavailable -> Sleeping... [!]
HomeBroker02   | [...] RabbitMQ is unavailable -> Sleeping... [!]
rabbitmq       | Starting broker...2023-05-29 22:55:41.926447+00:00 [warning] <0.700.0> M
RoboTrader04   | [!] RabbitMQ is up -> Executing Command... [!]
rabbitmq       | completed with 4 plugins.
RoboTrader01   | [!] RabbitMQ is up -> Executing Command... [!]
BolsadeValores | [!] RabbitMQ is up -> Executing Command... [!]
RoboTrader02   | [!] RabbitMQ is up -> Executing Command... [!]
BolsadeValores | 19:55:42 - INFO - [...] BV aguardando mensagens...
HomeBroker01   | [!] RabbitMQ is up -> Executing Command... [!]
HomeBroker01   | 19:55:42 - INFO - [...] HB1 aguardando mensagens...
RoboTrader03   | [!] RabbitMQ is up -> Executing Command... [!]
HomeBroker02   | [!] RabbitMQ is up -> Executing Command... [!]
HomeBroker02   | 19:55:42 - INFO - [...] HB2 aguardando mensagens...
RoboTrader04   | 19:55:47 - INFO - [*] Robo4 solicitando lista de ações ao HB2...
RoboTrader04   | 19:55:47 - INFO - [...] Robo4 aguardando para enviar mensagens...
HomeBroker02   | 19:55:47 - INFO - [*] robo4 solicitou a lista de ações .

```

Por fim, o processo de sincronização entre a BV e o HB utilizando o Algoritmo de Berkeley é demonstrado. Os logs exibem os valores de tempo utilizados na sincronização e o ajuste dos relógios para garantir a correta ordenação das transações. Essa parte dos testes é crucial para validar a eficácia do algoritmo na sincronização dos relógios dos componentes distribuídos presentes na simulação.

Figura 6 – Sincronização

```
HomeBroker02 | 19:55:52 - INFO - [*] HB2 solicitando lista de ações a BV...
BolsadeValores | 19:55:52 - INFO - [#] Sincronizar enviado da BV ao hb2 .
BolsadeValores | 19:55:52 - INFO - [$] Pedido de compra de 46 ACA01, realizado por hb2, processado com sucesso!
BolsadeValores | 19:55:52 - INFO - [*] Lista de ações enviada da BV ao hb2
HomeBroker02 | 19:55:52 - INFO - [#] Iniciando sincronização com a BV...
HomeBroker02 | 19:55:52 - INFO - [#] Tempo do HB2 (antes de sincronizar): 19:55:42
HomeBroker02 | 19:55:52 - INFO - [#] Tempo do HB2 (após sincronizar): 19:55:41
BolsadeValores | 19:55:52 - INFO - [#] Sincronizando com hb2...
BolsadeValores | 19:55:52 - INFO - [#] Tempo da BV (antes de sincronizar): 19:55:40
BolsadeValores | 19:55:52 - INFO - [#] Tempo da BV (após sincronizar): 19:55:41
BolsadeValores | 19:55:52 - INFO - [#] Finalizada sincronização com o hb2!
HomeBroker02 | 19:55:52 - INFO - [*] Lista de ações recebida da BV no HB2 !
```

## 4 CONCLUSÃO

Este projeto demonstrou a implementação de um algoritmo de sincronização de tempo - Algoritmo de Berkeley - em um ambiente de sistemas distribuídos. Este ambiente, um simulador de bolsa de valores, evidenciou a importância crítica da sincronização de tempo para garantir a operação coerente e eficiente dos sistemas como um todo.

O modelo simulado representa um exemplo prático e escalável de um sistema de negociação automatizado. A utilização do Broker de mensagens RabbitMQ ilustrou uma abordagem robusta e eficiente para a comunicação entre os componentes do sistema, enquanto a implementação baseada em containers Docker ressaltou a importância da flexibilidade e escalabilidade na arquitetura de sistemas distribuídos modernos.