



UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA - SÃO CARLOS  
DEPARTAMENTO DE COMPUTAÇÃO (DC)

---

## SIMULAÇÃO DE UMA BOLSA DE VALORES DISTRIBUÍDA

**Nome dos membros:**

*Henrique Cordeiro Frajacomio, 726536*

*Pedro Henrique Dumont Mauad, 619736*

São Carlos, SP, 2019

---

# 1 Introdução

Este relatório diz respeito ao primeiro trabalho desenvolvido na disciplina *Sistemas Distribuídos* da Universidade Federal de São Carlos.

O sistema desenvolvido simula uma bolsa de valores, com comunicação usando Sockets ZMQ, contendo 5 camadas de componentes:

- **Stock Managers:** sistemas que gerenciam o preço de uma única ação.
- **Broker:** componente que recebe todas as ações dos Stock Managers e distribui para seus Workers.
- **Workers:** componentes gerados pelo Broker. Cada Worker recebe uma ação do Broker, podendo ser instanciados mais Workers de acordo com a demanda do sistema.
- **Monitor:** componente que recebe as ações dos Workers e as envia para o clients.
- **Clients:** os programas de acesso dos usuários. Possui suporte para RPCs, apesar de nenhuma ter sido totalmente implementada.

## 2 Arquitetura

O sistema desenvolvido possui a arquitetura representada pela Figura 1.

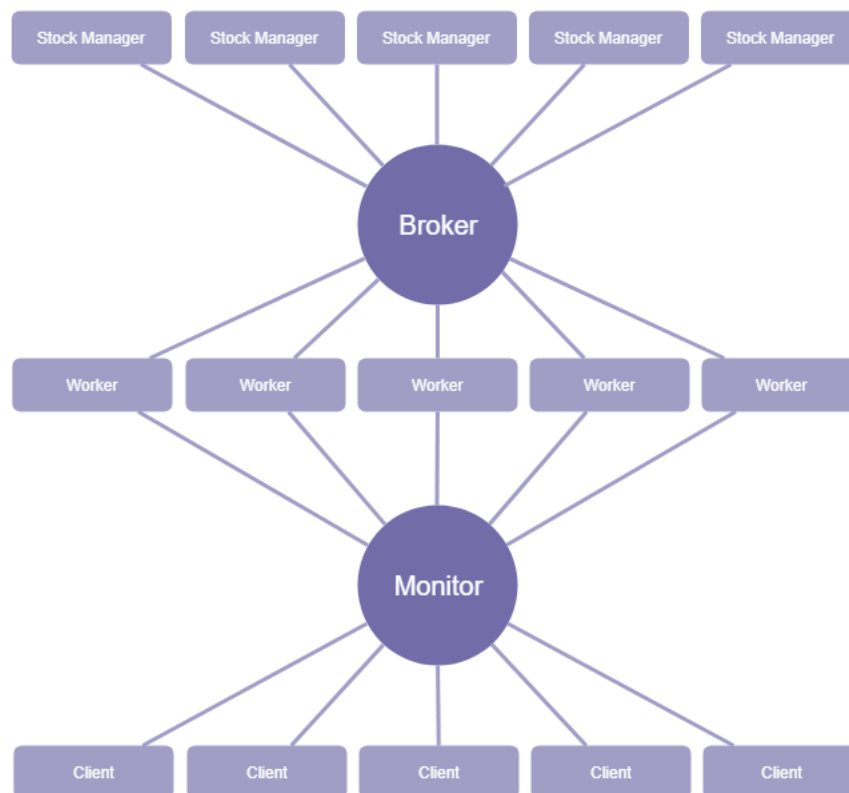


Figura 1: Arquitetura do sistema distribuído

---

Os Stock Managers criam e gerenciam o preço das ações, que por sua vez, são enviadas para o Broker. O Broker instancia Workers que carregam uma ação e a disseminam para o Monitor. O Monitor mantém os preços das ações atualizados e os envia para os clientes conectados. O cliente tem o preço das ações atualizadas a cada segundo em sua tela.

## 3 Utilização

para a execução do sistema, os códigos *StockManager.py*, *Broker.py*, *Monitor.py* e *Client.py* devem ser executados. Para facilitar o processo, o script *run.py* foi criado para executar os três primeiros códigos, que atuam como servidores, automaticamente como três processos separados. Basta o usuário rodar o *client.py* depois disso.

```
python run.py
python Client.py
```

### 3.1 Requisitos

Para a execução do programa, é necessário:

- Python 3 ou superior
- Python 3 como instalação padrão do Python
- Biblioteca zmq