

Relatório do Projeto A3

Sistemas distribuídos

Componentes da equipe:

Nalbert Bastos dos Santos (RA:1272117627)

João Marcos de Carvalho dos Santos (RA:1272118596)

Fabiana Amorim Lopes (RA: 1272116875)

Rafael Nogueira André (RA: 1272117401)

Deivid Allan Santos Da Silva (RA 1272119169)

Lucca Serra de melo Reis (RA 1272116626)

Introdução:

A equipe desenvolveu um projeto que simule a captação de dados de venda em uma rede de lojas. No projeto foi implementado a possibilidade dos vendedores informarem suas vendas, os gerentes podem ter acesso a informações dessa venda e tudo isso estará em um servidor onde terá os dados processados e hospedados. O projeto foi implementado em Python e utilizamos sockets para comunicação entre os clientes (vendedores e gerentes) e o servidor. E o SQLite como banco de dados para armazenar as informações de vendas.

Requisitos de Software:

Linguagem de Programação: Python 3.x

Bibliotecas: socket, json, sqlite3

Banco de Dados: SQLite

Instruções de Instalação e Execução:

Deverá ter o Python 3.x instalado.

Instale as bibliotecas necessárias executando o seguinte comando no terminal:

```
pip install sqlite3
```

Abra terminais separados para cada parte do projeto (servidor, vendedores e gerentes).

O terminal deverá estar na pasta onde estão os arquivos

No terminal do servidor, execute o seguinte comando para iniciar o servidor:

```
C:\Users\Desktop\SD A3 Definitiva> servidor.py
```

No terminal dos vendedores, execute o seguinte comando para iniciar os vendedores:

```
C:\Users\Desktop\SD A3 Definitiva> vendedor.py
```

No terminal dos gerentes, execute o seguinte comando para iniciar os gerentes:

```
C:\Users\Desktop\SD A3 Definitiva> gerente.py
```

Siga as instruções exibidas na interface para interagir com o aplicativo.

Arquitetura e Estratégia:

Com pedido nos requisitos o projeto está baseado em uma arquitetura cliente-servidor, onde o servidor central hospeda as informações de vendas e responde a consultas dos gerentes. Os vendedores enviam mensagens ao servidor informando suas vendas, enquanto os gerentes enviam mensagens solicitando diferentes tipos de consultas.

Essa comunicação entre os clientes e o servidor ocorre pelos sockets TCP/IP. Cada cliente se conecta ao servidor usando o endereço IP e a porta correspondente. A troca de mensagens é realizada no formato JSON, permitindo uma estrutura de dados flexível para transmitir as informações.

O servidor tem um banco de dados SQLite para guardar as informações de vendas. Cada venda é registrada com o nome do vendedor, id da loja, data da venda e valor vendido. As consultas dos gerentes são processadas pelo servidor, faz as consultas no banco de dados e retorna as respostas.

Além disso, o projeto inclui um processo de eleição de servidor temporário, caso o servidor principal fique inacessível. Os clientes se comunicam entre si para selecionar um servidor temporário que assumirá as responsabilidades do servidor principal. Quando o servidor principal se recupera, ele informa aos clientes e assume novamente a responsabilidade.

Algoritmos:

Utilizamos Algoritmo de Comunicação Cliente-Servidor: O servidor aguarda solicitações dos clientes usando sockets TCP/IP. Os clientes estabelecem conexão com o servidor e enviam mensagens contendo operações específicas, como informar uma venda ou solicitar uma consulta. O servidor

recebe as mensagens, identifica o tipo de operação e processa a solicitação correspondente.

Utilizamos também o Algoritmo de Armazenamento de Dados: O servidor utiliza o banco de dados SQLite para armazenar as informações de vendas. Cada venda é inserida como um registro no banco de dados, contendo os detalhes da venda. Para consultas, o servidor realiza consultas SQL no banco de dados e retorna os resultados correspondentes.

E por fim Algoritmo de Eleição de Servidor Temporário: Quando o servidor principal fica indisponível, os clientes iniciam um processo de eleição para selecionar um servidor temporário. Os clientes se comunicam entre si para tomar a decisão com base em algum critério, como a disponibilidade ou ordem de chegada. Uma vez selecionado, o servidor temporário assume as responsabilidades do servidor principal até que ele se recupere.

Conclusão:

Bom como foi proposto fizemos um projeto relacionado a captação de dados de vendas. Usamos sockets como foi pedido e também uma das linguagens que foi abordada em aulas e que foi pedida nos requisitos. Os vendedores podem enviar suas vendas e os gerentes podem consultar as informações, e usamos o banco recomendado que foi o SQLite. A implementação do processo de eleição de servidor temporário garante que as funcionalidades do servidor estejam sempre disponíveis, mesmo em caso de falhas temporárias.