

Sistema de Transações de Carteira e Loja

Aluno: Gabriel Lima Nunes
Matrícula: 2021032048

Este projeto implementa um sistema simples de transações entre uma carteira (wallet) e uma loja (store) usando comunicação RPC. Os serviços de carteira e loja permitem a criação de ordens de pagamento, transferências e consultas de saldo, além de permitir o desligamento ordenado dos servidores.

Estrutura do Projeto

- **store.proto**: Definição do serviço e das mensagens RPC para o servidor da loja.
- **wallet.proto**: Definição do serviço e das mensagens RPC para o servidor da carteira.
- **store_server.py**: Implementação do servidor da loja.
- **wallet_server.py**: Implementação do servidor da carteira.
- **store_client.py**: Cliente para interagir com o servidor da loja.
- **wallet_client.py**: Cliente para interagir com o servidor da carteira.
- **Makefile**: Arquivo para automatizar a geração dos stubs e a execução dos servidores e clientes.

Como Executar

1. Gerar os Stubs

Primeiro, gere os stubs necessários para o gRPC usando o Makefile.

(Opcional) Execute o comando:

```
make stubs
```

Este comando gera os arquivos necessários a partir dos arquivos `.proto`.

2. Executar o Servidor de Carteiras (Wallet)

Para executar o servidor da carteira, use o seguinte comando:

```
make run_serv_banco arg1=5555
```

Aqui, **arg1** é o número da porta onde o servidor da carteira será iniciado. Você pode especificar outra porta, se desejar.

Após iniciar o servidor ele deve ler da entrada padrão uma lista de carteiras: linhas cada uma com o identificador (string) e o valor (inteiro positivo) de uma carteira, separados por um espaço. Um exemplo de conteúdo para essa entrada seria:

Dorgival 10

Bezos 1000000

Papai_Noel 100000

Após receber a lista de carteiras o servidor espera uma linha vazia ou EOF e então ele tornará sua porta disponível.

3. Executar o Cliente de Carteira (Wallet)

Para interagir com o servidor da carteira, use o comando:

```
make run_cli_banco arg1=carteira_cliente arg2=localhost:5555
```

Aqui, **arg1** é o ID da carteira do cliente e **arg2** é o endereço do servidor de carteira no formato **host:porta**.

4. Executar o Servidor da Loja (Store)

Para executar o servidor da loja, use o comando:

```
make run_serv_loja arg1=10 arg2=6666 arg3=carteira_loja  
arg4=localhost:5555
```

- **arg1** é o preço do item (neste caso, 10 unidades).
- **arg2** é o número da porta onde o servidor da loja será iniciado.
- **arg3** é o ID da carteira da loja.
- **arg4** é o endereço do servidor da carteira no formato **host:porta**.

5. Executar o Cliente da Loja (Store)

Para interagir com o servidor da loja, use o comando:

```
make run_cli_loja arg1=carteira_cliente arg2=localhost:5555  
arg3=localhost:6666
```

- `arg1` é o ID da carteira do cliente.
- `arg2` é o endereço do servidor da carteira no formato `host:porta`.
- `arg3` é o endereço do servidor da loja no formato `host:porta`.

6. Limpar Arquivos Intermediários

Para remover todos os arquivos intermediários, exceto os necessários para entrega, execute:

```
make clean
```

Este comando remove todos os arquivos gerados durante a compilação e execução, deixando apenas os arquivos de código-fonte e o Makefile.