

Trabalho Prático Individual 01

Desenvolvimento de Sistemas Distribuídos Simples

Emanuely Carvalho
2021039905

BELO HORIZONTE
2024

Introdução

Neste projeto, foi desenvolvido um sistema simples de gerenciamento de corridas de táxi, onde um cliente pode solicitar uma corrida para um motorista que está em uma localização fixa. No caso, utilizei a latitude e a longitude da Universidade Federal de Minas Gerais - UFMG (-19.871904, -43.966248). O sistema utiliza o protocolo TCP/IP para comunicação entre o cliente e o servidor, que gerencia as solicitações de corrida. O servidor calcula a distância entre o cliente e o motorista com base nas coordenadas geográficas fornecidas pelo cliente, utilizando a fórmula de Haversine. O motorista pode aceitar ou recusar a corrida, e se aceitar, a distância até o cliente é atualizada a cada 2 segundos até o motorista chegar ao destino. Após a conclusão da corrida, o motorista volta ao estado de “Aguardando solicitação”, pronto para receber uma nova corrida.

O código enviado está documentado de acordo com o raciocínio desenvolvido para a implementação do mesmo. É de bom tom destacar que, em alguns momentos, por questões puramente estéticas, o terminal é limpo, simulando o máximo possível uma interface amigável para os usuários (cliente e motorista).

Sumário

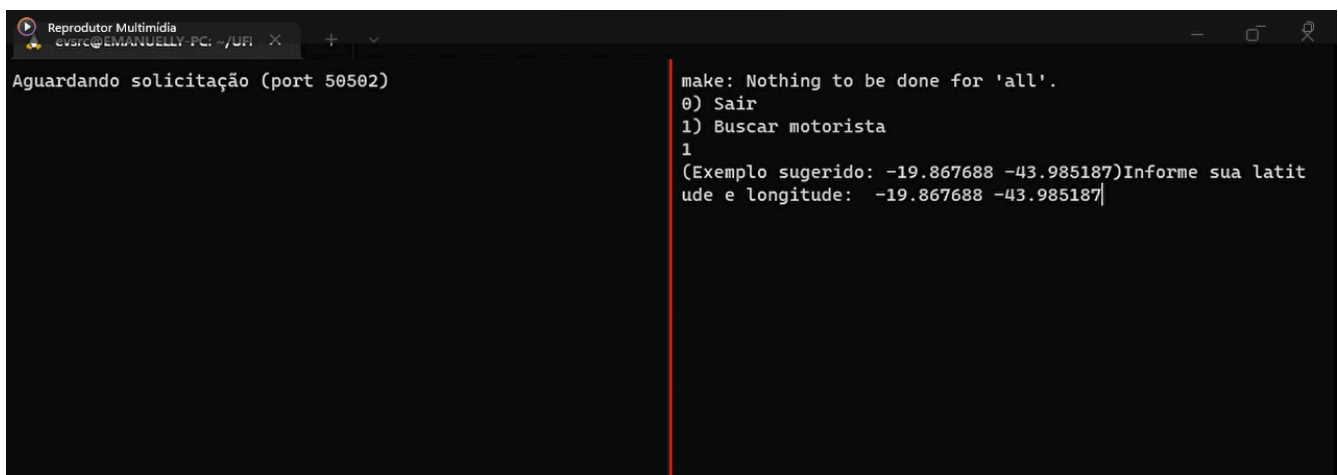
Introdução	2
Sumário	2
Demonstração	3
Instruções de Uso	5

Demonstração

Abaixo, está o link de um vídeo de demonstração do uso do código gerados a partir de vídeos que foram acelerados em 2,5x. No primeiro momento, o motorista rejeita a corrida e volta para o estado de “Aceitando solicitações”, o cliente por sua vez volta para o Menu Inicial. No segundo momento, o motorista aceita a corrida, envia a cada 2 segundos mensagens atualizando a distância do cliente e ambos recebem simultaneamente na interface o aviso de que o motorista chegou ao destino. Para o cliente, o programa é encerrado e o motorista novamente volta a aceitar solicitações.

[Link do vídeo no drive.](#)

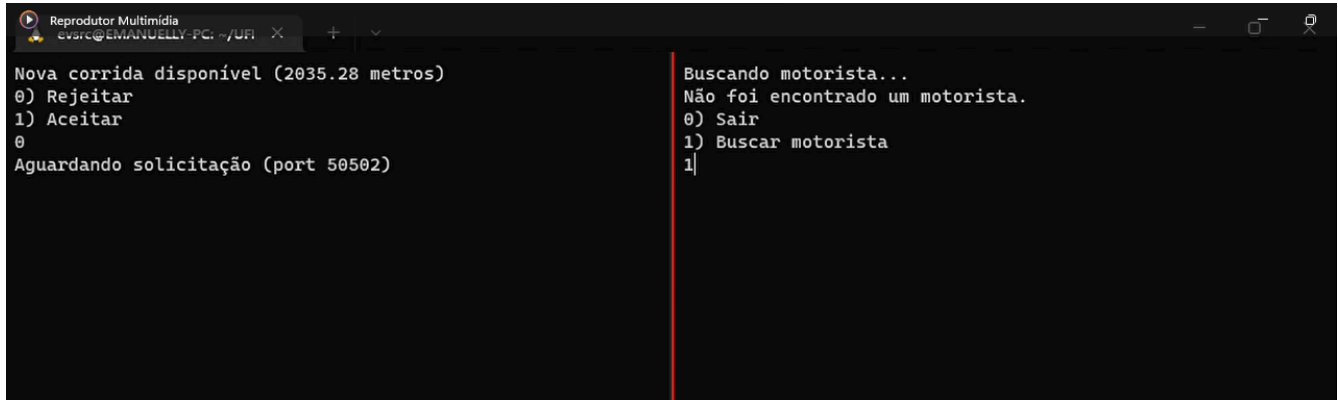
A seguir, screenshots do processo.



```
Reprodutor Multimidia
evsrc@EMANUELLY-PC: ~/UR  X  +  -
Aguardando solicitação (port 50502)

make: Nothing to be done for 'all'.
0) Sair
1) Buscar motorista
1
(Exemplo sugerido: -19.867688 -43.985187)Informe sua latitude e longitude: -19.867688 -43.985187|
```

Cliente solicita a corrida informando latitude e longitude

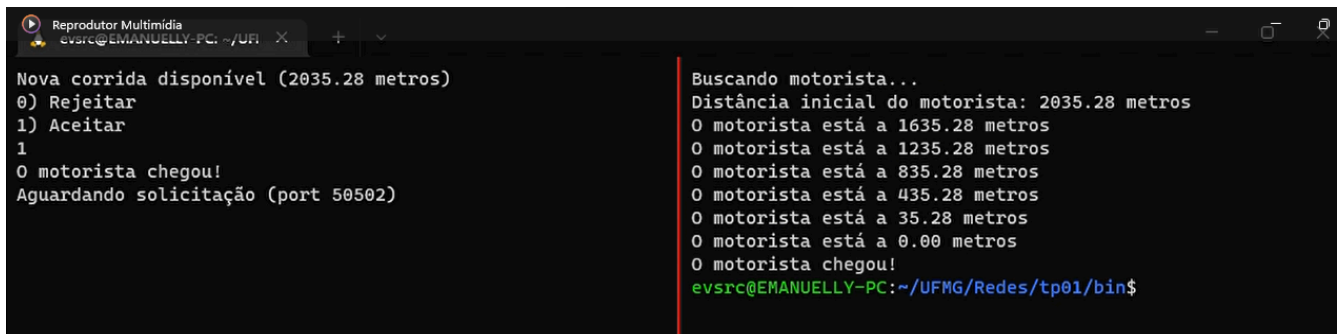


```
Reprodutor Multimidia
evsrc@EMANUELLY-PC: ~/URI X + v

Nova corrida disponível (2035.28 metros)
0) Rejeitar
1) Aceitar
0
Aguardando solicitação (port 50502)

Buscando motorista...
Não foi encontrado um motorista.
0) Sair
1) Buscar motorista
1
```

Motorista nega a corrida e volta a buscar solicitações. Cliente recebe a negativa e volta ao menu inicial.



```
Reprodutor Multimidia
evsrc@EMANUELLY-PC: ~/URI X + v

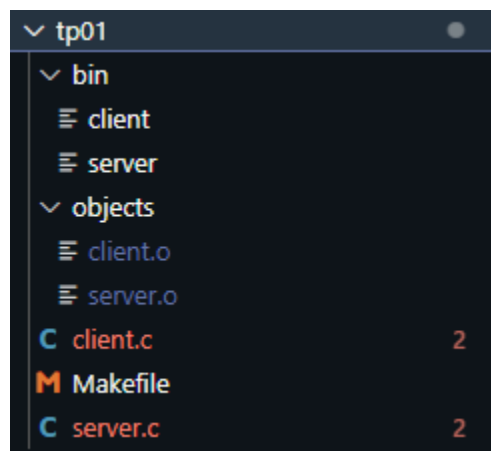
Nova corrida disponível (2035.28 metros)
0) Rejeitar
1) Aceitar
1
O motorista chegou!
Aguardando solicitação (port 50502)

Buscando motorista...
Distância inicial do motorista: 2035.28 metros
O motorista está a 1635.28 metros
O motorista está a 1235.28 metros
O motorista está a 835.28 metros
O motorista está a 435.28 metros
O motorista está a 35.28 metros
O motorista está a 0.00 metros
O motorista chegou!
evsrc@EMANUELLY-PC: ~/UFMG/Redes/tp01/bin$
```

Motorista aceita a corrida e envia atualizações a 400m/s. O aviso da chegada é enviado para ambos simultaneamente. O motorista volta a receber solicitações enquanto para o cliente o programa é encerrado.

Instruções de Uso

Após descompactar a pasta do envio, basta acessá-la no terminal e rodar o comando make, sem nenhum parâmetro adicional. Ao fazer isso, você verá uma estrutura de pastas semelhante a esta:



Para rodar o código, você tem as seguintes opções

- Na pasta raiz
 - `./bin/server ipv4 50501`
 - `./bin/server ipv6 50501`
 - `./bin/client 127.0.0.1 50501`
 - `./bin/server ::1 50501`
- Na pasta bin
 - `./client ipv4 50501`
 - `./client ipv6 50501`
 - `./client 127.0.0.1 50501`
 - `./client ::1 50501`

Lembrando que, para haver a conexão bem sucedida, é fundamental que os parâmetros sejam os mesmos para ambas as solicitações (cliente e servidor).