

Backend Software Engineer Challenge

Rota de Viagem

Um turista deseja viajar pelo mundo pagando o menor preço possível independentemente do número de conexões necessárias. Vamos construir um programa que facilite ao nosso turista, escolher a melhor rota para sua viagem.

Para isso precisamos inserir as rotas através de um arquivo de entrada.

Input Example

```
GRU, BRC, 10  
BRC, SCL, 5  
GRU, CDG, 75  
GRU, SCL, 20  
GRU, ORL, 56  
ORL, CDG, 5  
SCL, ORL, 20
```

Explicando

Caso desejemos viajar de **GRU** para **CDG** existem as seguintes rotas:

1. GRU - BRC - SCL - ORL - CDG ao custo de **\$40**
2. GRU - ORL - CGD ao custo de **\$64**
3. GRU - CDG ao custo de **\$75**
4. GRU - SCL - ORL - CDG ao custo de **\$45**

O melhor preço é da rota **1** logo, o output da consulta deve ser:

GRU - BRC - SCL - ORL - CDG.

Execução do programa

A inicialização do teste se dará por linha de comando onde o primeiro argumento é o arquivo com a lista de rotas inicial.

```
$ mysolution input-routes.csv
```

Duas interfaces de consulta devem ser implementadas: - Interface de console deverá receber um input com a rota no formato "DE-PARA" e imprimir a melhor rota e seu respectivo valor.

Exemplo:

```
please enter the route: GRU-CGD  
best route: GRU - BRC - SCL - ORL - CDG > $40  
please enter the route: BRC-CDG  
best route: BRC - ORL > $30
```

Também será necessária a implementação de 2 endpoints Rest, um para registro de rotas e outro para consulta de melhor rota.

A interface Rest deverá suportar:

- Registro de novas rotas. Essas novas rotas devem ser persistidas no arquivo csv utilizado como input(input-routes.csv)
- Consulta de melhor rota entre dois pontos.

Recomendações

Para uma melhor fluidez da nossa conversa, atente-se aos seguintes pontos:

- Envie apenas o link do seu repositório de código fonte
- Estruture sua aplicação seguindo as boas práticas de desenvolvimento
- Evite o uso de frameworks ou bibliotecas externas à linguagem. Utilize apenas o que for necessário para a exposição do serviço
- Implementar testes unitários seguindo as boas práticas de mercado
- Sua aplicação deve ser fácil executada por outra pessoa, então é interessante termos sua aplicação executando em container
- Documentação em um arquivo Texto ou Markdown descreva:
 - Como executar a aplicação,
 - Documente as principais interfaces da sua aplicação (Ex: Apis ou CLI)
 - Estrutura dos arquivos/pacotes,
 - Explique as decisões de design adotadas para a solução
 - Descreva sua API Rest de forma simplificada