

Projeto 2 - Programação Funcional - Lambda Travel

<https://youtu.be/f9eWung9Aq8>

Matheus Ferreira Machado - 11201920804

Executando o projeto

O projeto utiliza o stack, para executá-lo, basta executar:

```
stack run          # Executa a aplicação
```

Aplicação

A aplicação simula um gerenciador de viagens como *Google Flights*, *Skyscanner*, etc. Ela permite buscar os melhores voos entre uma origem e um destino usando critérios de preço ou distância percorrida. A partir de uma lista pré determinada de localizações (localizada em *data/locations.txt*) e uma lista de rotas de diferentes companhias aéreas (arquivos no diretório *data/air-companies.txt*), a aplicação utiliza o algoritmo de Dijkstra para calcular as melhores rotas de acordo com o critério escolhido pelo cliente.

Detalhes Técnicos

Como já mencionado, a base do projeto é a implementação do algoritmo de Dijkstra em Grafos para encontrar o menor caminho entre nós, sendo cada nó uma localização nesse cenário. Comparado à primeira entrega, o algoritmo de Dijkstra foi completamente refatorado para utilizar a mônada state. Dessa forma, a implementação tornou-se muito mais clara, já que agora os nós visitados e os caminhos conhecidos ficam armazenados em um estado que é atualizado durante a execução.

Uso

Quando a aplicação é executada, mostra-se uma lista de localizações, deve ser escolhida uma localização que será a origem da busca. Em seguida, o usuário deve escolher se vai buscar um destino específico ou buscar todos os destinos disponíveis. Por fim, o usuário deve informar o critério de busca, preço ou distância percorrida. Após o envio de todas essas informações, a aplicação imprime as melhores rotas encontradas.

Desafios e destaques

O maior desafio foi realizar a refatoração do algoritmo de Dijkstra para utilizar a mônada state, pois ainda não havia utilizado ela na prática, resultando em algumas dificuldades para implementar as funções corretamente.

Um destaque da versão final do projeto é a implementação do algoritmo de Dijkstra com mônadas, acredito que o código ficou muito mais enxuto e compreensível. Além disso, a implementação ficou genérica o suficiente para agregar novas funcionalidades à aplicação sem a necessidade de realizar alterações neste código, resultando em uma aplicação com manutenção facilitada.