

Rede de trens entre cidades

Estrutura de Dados II
Professor Francisco da Fonseca Rodrigues

Integrantes:

Lorennna Joanna Leal Nunes - 18343

Maria Eduarda Elias Rocha - 18190

Campinas, 2019

Índice de conteúdo

Objetivo.....	03
Desenvolvimento.....	03
Conclusão.....	05

Objetivo

O projeto tem como objetivo fornecer aos usuários um aplicativo móvel que permita verificar caminhos entre cidades (no caso, cidades entre Portugal e Espanha), através de viagens de trem com as especificações constatadas no enunciado do projeto. Ele deverá encontrar um roteiro entre duas cidades escolhidas pelo usuário e fornecer a distância total entre elas e o tempo estimado do percurso e deverá ter a opção de incluir novas cidades às já existentes, adicionando também a distância e tempo de cada ligação.

O programa foi desenvolvido na linguagem de programação C#, com a utilização de uma matriz de adjacência representando o grafo com as cidades e seus dados, o método de Dijkstra para realizar algumas operações, backtracking e pilhas para encontrar as possíveis soluções. As mesmas foram apresentadas na disciplina Estrutura de Dados II do 4º semestre do curso técnico de Informática do Colégio Técnico de Campinas (COTUCA).

Desenvolvimento

- **Na escola, 12 de novembro de 2019;**
Criamos o repositório do projeto no GitHub e começamos a desenvolver seu design no Android Studio. E em seguida, descobrimos que não havia o componente comboBox e seria necessário utilizarmos o Spinner.
- **Na escola, 13 de novembro de 2019;**
Os itens que serão adicionados nos Spinners de origem e destino serão as cidades que estão no arquivo texto. Portanto, criamos uma classe Cidade, que será um objeto, com seus getters e setters, de Cidade que possui os campos de índice, nome e coordenadas. E em seguida começamos a desenvolver o método que faz tal leitura do arquivo e cria uma lista de cidades.
- **Na escola, 14 de novembro de 2019;**
Finalizamos o método dito anteriormente que lê as cidades presente no arquivo texto. Separamos as cidades por linhas, pelo fato do arquivo estar nesta formatação, e em cada linha havia o nome da cidade, sua coordenada do eixo x e sua coordenada do eixo y.
- **Na escola, 21 de novembro de 2019;**
Transferimos o projeto para a linguagem C#, por acharmos que seria uma linguagem melhor já que muitas classes já estavam feitas.
- **Na escola, 22 de novembro de 2019;**
Decidimos retornar o projeto para Java após passar muito tempo tentando usar o visual (Xamarin) no Visual Studio 2017. As seguintes classes foram

refeitas para Java: ListaSimples(List), NoLista(Node), Fila(Queue), Pilha(Stack), Grafo(Graph), BucketHash e as classes Vertice e DistOriginal(Distance) foram feitas como classes internas de Graph. Para treino decidimos fazer as classes na língua inglesa, mas todas as classes de aplicação foram feitas em português.

- **Na escola, 29 de dezembro de 2019;**

O algoritmo de Dijkstra foi implementado e os caminhos foram encontrados a partir do algoritmo.

Além disso foi usado um objeto da classe Canvas para desenhar os pontos das cidades e as linhas entre elas.

- **Em casa, 30 de dezembro de 2019;**

Foi pesquisado como salvar as alterações nos arquivos texto dentro da pasta assets do projeto Android. Descobrimos que não seria possível utilizar os arquivos dessa pasta e que seria necessário o uso de armazenamento interno do projeto. Para tal, necessitamos do uso do próprio programa para criar esse arquivo, então para manter as alterações dos arquivos do armazenamento interno decidimos colocar um botão para que se o usuário queira recrie os arquivos.

Também foi descoberto que para cada computador que a aplicação rodasse seria necessário recriar os arquivos, então avisamos o usuário toda vez que iniciar a tela principal.

- **Em casa, 01 de dezembro de 2019;**

Criamos a tela que será exibida ao usuário quando o mesmo selecionar a opção de adicionar uma nova cidade na tela principal. E para acrescentar a nova cidade, deverá ser informado seu nome e suas coordenadas do eixo x e do eixo y.

Posteriormente criamos a classe correspondente à tela em questão. Classe que faz com que quando o usuário clicar no botão adicionar (presente na tela desenvolvida) adicione no arquivo texto "Cidades" interno.

- **Na escola, 02 de dezembro de 2019;**

Parecido com a tela de adicionar cidades, criamos a tela que será exibida ao usuário quando o mesmo selecionar a opção de adicionar um novo caminho na tela principal. Para adicionar o novo caminho será necessário informar a cidade de origem e de destino, que o usuário poderá escolher entre as cidades possíveis que já foram adicionadas anteriormente, a distância entre as duas cidades escolhidas e o tempo que gasto pelo trem percorrendo esse trajeto.

Em seguida desenvolvemos a classe correspondente à tela comentada acima. Tal classe que adiciona o caminho no arquivo texto "GrafoTremEspanhaPortugal". Mas tivemos um problema, nosso construtor

da classe Caminho estava recebendo apenas parâmetros com o código das cidades de origem e destino e não como string como precisávamos.

- **Na escola, 03 de dezembro de 2019;**

Fizemos a verificação para fazer com que a pessoa não adicionasse um caminho com a cidade destino igual a cidade origem e nem escolher uma rota com as mesmas características. Também foram feitas todas as verificações de inserção como: tamanho das informações e como elas deveriam ser adicionadas.

- **Na escola, 04 de dezembro de 2019;**

Realizamos os testes finais. comentamos e documentamos todo o código.

Conclusão

O projeto foi extremamente importante pois com ele colocamos em prática toda nossa aprendizagem da matéria de Estrutura de Dados II. Concluimos que do mesmo modo que os números podem ser representados em sistemas diferentes, dados podem ser armazenados em estruturas diferentes, ou seja, há diversos métodos que podem ser utilizados para guardar tais dados na memória e no disco para serem processados.

O projeto aprimorou nosso conhecimento a respeito da técnica Bucket Hash que é uma forma de guardar informações para acesso rápido, o método Dijkstra para determinar o menor caminho em um grafo ponderado e a utilização de pilha, fila e lista ligadas que aprendemos na matéria Estrutura de Dados I.