Relatório do Projeto 3 de Estrutura de Dados: “Caminhos de Marte”

Danilo de Oliveira Nunes - 18203

João Victor Javitti Alves - 18207

Introdução:

Sei lá oq escrever chico, só preciso tirar 1 nesse projeto dá uma nota ai que nois agradesse kkkkkkkkkkk

Desenvolvimento:

Para o início do projeto começamos com as lógicas mais simples e já conhecidas como implementar o método LerArquivo() para ser realizado enquanto o formulário windows se inicializa. Após isso partimos para a parte em que desenhamos no mapa os nomes das cidades, encontramos várias dificuldades por não conhecermos o método paint e não termos muita noção de como usá-lo, mas com apoio da monitoria e pesquisas intensivas na internet conseguimos entender seu funcionamento o que também facilitou o método de desenhar a matriz que começamos a implementar em seguida.

Backtracking Para a Procura do Caminho:

Ficamos com muitas dúvidas de como deveria ser feito esta parte do projeto, primeiramente consideramos apenas ir guardando os caminhos percorridos numa pilha e ir tentando caminho por caminho até alcançar a cidade destino, essa ideia nos ocorreu tanto pelo Backtracking já trabalhado conosco em sala de aula quanto pelo projeto Labirinto efetuado na disciplina de Programação Orientada a Objetos no ano anterior.

Porém, devido a escolha de uma matriz esparsa para armazenar os dados de caminhos fornecidos pelo arquivo, percebi que ao inserir os valores de IdCidadeDestino (atributo da classe CaminhoEntreCidades usada para guardar os valores do arquivo) como coluna, possuíamos na coluna todos os caminhos que levavam a uma determinada cidade, essa que o Id seria o número da própria coluna (não exatamente, na verdade, os caminhos fornecidos no arquivo eram unilaterais e decidimos no projeto admitir que sim eram utilizáveis para ida ou volta o que nos deu a necessidade de inserir “caminhos invertidos” na matriz para facilitar a lógica que usamos).

A lógica que efetuei foi a seguinte: Pesquisando pela coluna de código igual a cidade que o usuário buscou, teríamos os caminhos que levavam a ela e por estes, receberíamos também o código da cidade que deveríamos visitar para ir até lá, caso não fosse exatamente a cidade origem, apenas teríamos que pesquisar como chegaríamos a esta cidade usando o mesmo método que descrevemos anteriormente, usando assim uma recursão que era ativada toda vez que queríamos chegar a uma cidade, e como deveríamos checar todos os caminhos possíveis que levariam a esta cidade também colocamos no método um while que percorria todos os caminhos possíveis, assim, teríamos acesso a todos os caminhos que levariam de uma cidade para a outra.

Encontramos alguns problemas devido ao uso de uma lista ligada para se armazenar os caminhos, primeiramente, como a lista usa ponteiros para se construir, mesmo que a recursão retornasse ao método, os valores editados da lista seguiam editados, para resolver isso tivemos que mandar para a próxima instância do método um clone da lista de cidades para que a atual não fosse modificada. Também tivemos problemas com o método Existe já que este presumia que a lista estivesse ordenada, para resolver apenas criamos um método para a ListaLigada que verifica se existe independente se está ordenada ou não.