

## ROTAS AÉREAS ENTRE CIDADES MARCIANAS

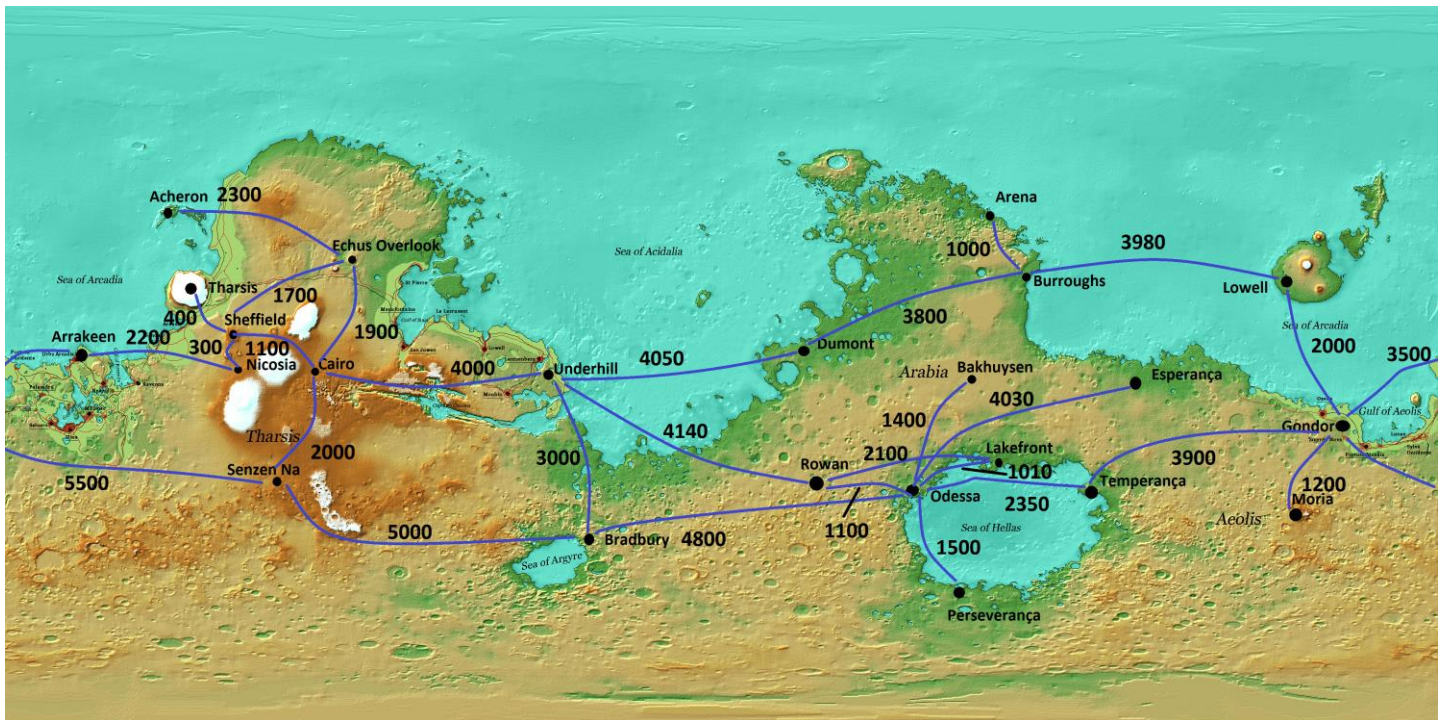
Depois da colonização e terraformação de Marte, em 2187 a Federação Marciana declarou-se independente da Terra e proporcionou aos seus habitantes uma malha aeroviária realizada por dirigíveis ligando os principais centros populacionais do planeta.

A Federação deseja fornecer aos seus usuários um aplicativo que permita verificar os caminhos entre essas principais cidades, através de viagens aéreas.

Para tanto, um arquivo texto contendo nomes de cidades, distância entre elas, tempo de percurso e preço da passagem é fornecido. Esse arquivo se chama CaminhosEntreCidadesMarte.txt.

Há também um arquivo texto chamado CidadesMarte.txt, o nome da cidade e a coordenada cartesiana (x, y) proporcional às dimensões de um mapa de Marte, onde a cidade deve ser localizada no mapa.

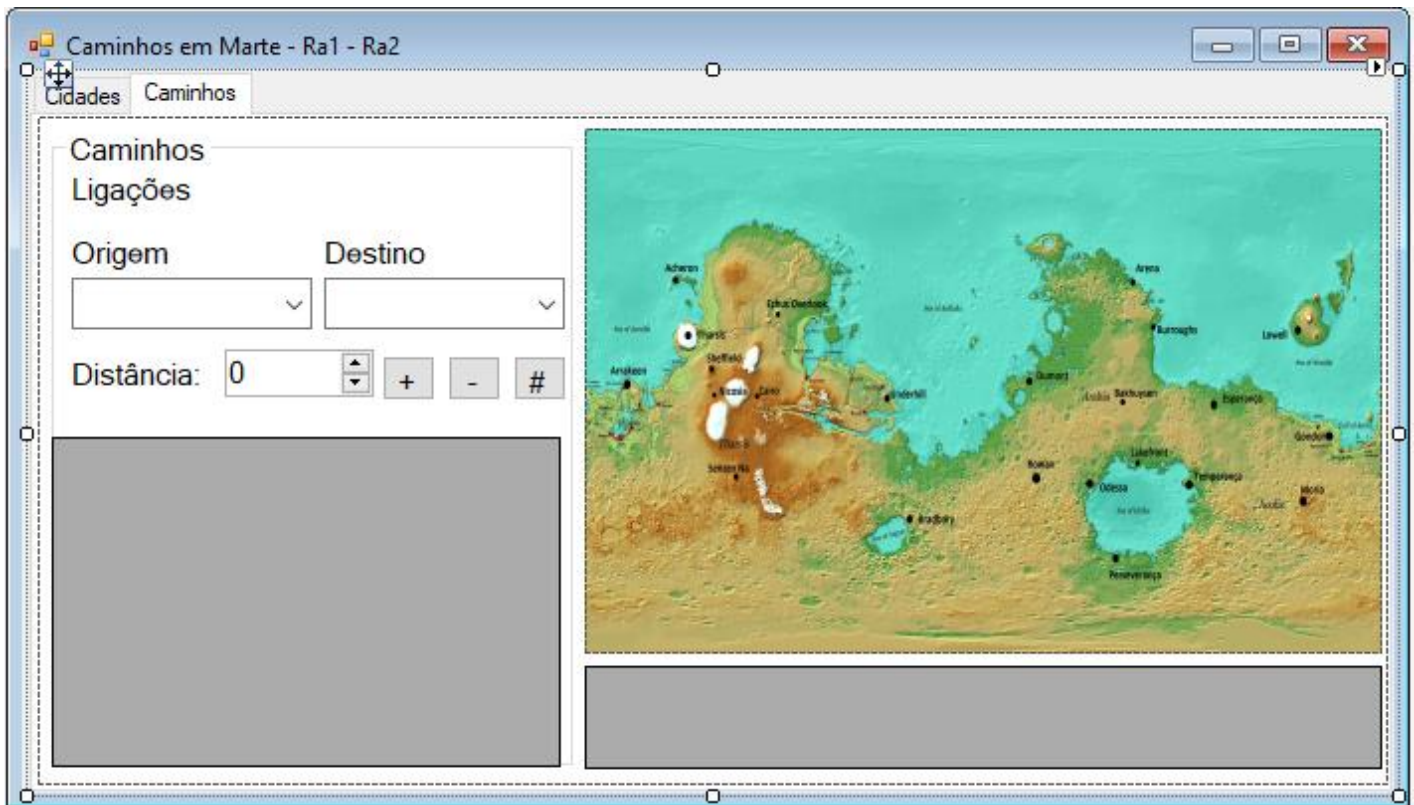
Continuando o projeto I já entregue, o aplicativo deverá encontrar todos os caminhos entre as cidades A e B indicadas e relacionar o mais vantajoso, de acordo com o critério de menor distância percorrida.



Nessa segunda parte do projeto, o arquivo de cidades deverá ser armazenado em um vetor de objetos da classe Cidade, vetor esse ordenado pelo nome das cidades, para busca de suas coordenadas para posicionamento dos extremos das linhas retas que determinam os caminhos entre cidades que façam parte dos percursos selecionados. O índice de cada cidade nesse vetor de cidades será usado, também, como índice de linha e de coluna dessa cidade numa matriz de adjacências, que será explicada mais à frente.

Como o mapa no seu tamanho original não caberá na tela, deve-se continuar permitindo que ele se ajuste ao tamanho da tela e lembre-se que isso mudará as coordenadas de exibição de cada cidade no mapa **proporcionalmente** à mudança da altura y e largura x do mapa apresentado na tela, numa proporção entre a largura e a altura da tela com a coordenada (X, Y) original da cidade.

O mapa deverá variar de tamanho caso se aumente ou diminua o tamanho do formulário durante a execução. Assim, como as coordenadas X e Y são proporcionais, para determinar a o pixel (x, y) exato da cidade no mapa que está sendo exibido, a largura do mapa deve ser multiplicada pela coordenada X da cidade e a altura do mapa deve ser multiplicada pela coordenada Y da cidade. Por exemplo, se o mapa estiver sendo exibido com tamanho de 1200 x 600 pixels (largura x altura), para a cidade de Acheron do exemplo acima as coordenadas onde o círculo deve ser exibido nesse mapa será (136, 173). Você pode escrever o nome da cidade abaixo do círculo.



Além de manter as funcionalidades já existentes no Projeto I, no início da execução do programa deverá também ser pedido o nome do arquivo de caminhos de ligação direta (CaminhoEntreCidadesMarte.txt), através de um OpenFileDialog. Esse arquivo deverá ser aberto e lido totalmente e as distâncias entre cidades armazenadas em uma matriz de adjacências cujos índices de acesso serão definidos pelo índice em que cada nome de cidade está no vetor de cidades citado no início do texto.

Os dados do arquivo de caminhos deverão gerar o conteúdo da matriz de adjacências, onde cada linha e cada coluna corresponde a uma das cidades (na ordem de armazenamento no vetor de objetos de cidades). Cada posição dessa matriz armazenará a distância que representa a ligação entre a cidade de origem (linha) e a cidade de destino (coluna).

#### Vetor de Cidades

	Nome	X	y
0	Acheron	0,11353	0,28857
1	Arena	0,68726	0,29248
2	Arrakeen	0,05298	0,48682
3	Bakhuisen	0,67456	0,52002
4	Bradbury	0,40747	0,74365
5	Burroughs	0,71265	0,37939
6	Cairo	0,21606	0,51025
7	Dumont	0,55737	0,48096
8	Echus Overlook	0,24194	0,35498
...			
N			

#### Matriz de Adjacências

	0	1	2	3	..	6	7	8	...	N
0								2300		
1										
2										
3										
4...										
5		1000								
6										
7					3800					
8						1900				
...										
N										

Os botões [+], [-] e [#] serão usados para disparar, respectivamente, eventos que façam a inclusão de uma ligação entre duas cidades, a exclusão de uma ligação já existente e a alteração de uma ligação já existente. Cada ligação é identificada pelo nome da cidade de origem em conjunto com o nome de uma cidade de destino. O usuário indicará as cidades envolvidas em uma ligação (um caminho direto) através dos comboboxes cbxOrigem e cbxDestino. Nesses comboboxes devemos adicionar os nomes de todas as cidades armazenadas no vetor de cidades, já ordenado. Quando o usuário selecionar o nome de uma cidade

num dos comboboxes, deveremos procurar no vetor de cidades tal nome, para encontrar seu índice nesse vetor e usar tais índices na matriz de adjacências. Porém, se o vetor de cidades estiver ordenado por nome de cidade, e os comboboxes forem preenchidos seguindo a mesma ordem do vetor de cidades (em ordem alfabética do nome da cidade), o ato de selecionar um item num combobox já informará o índice desse item nesse combobox, que será igual ao índice que a cidade tem no vetor de cidades, de forma que a pesquisa da cidade também já está praticamente pronta. O índice do item selecionado do `cbxOrigem` será o número da linha na matriz de adjacências e o índice do item selecionado no `cbxDestino` será o número da coluna na matriz de adjacências. O índice do item selecionado pode ser acessado na propriedade **SelectedIndex**.

Ao final da execução do programa, além de manter o salvamento da tabela de hash de cidades, deveremos também salvar a matriz de adjacência no formato do arquivo de caminhos original.

Inclua um novo botão no formulário (`btnBuscar`) que, quando clicado, fará a busca de todos os caminhos entre a cidade de origem e a cidade de destino selecionadas nos respectivos comboboxes. Ao final da busca de todos os caminhos, o melhor caminho deverá ser mostrado no mapa. Ao final da busca de todos os caminhos, um `DataGridView` `dgvCaminhos` deverá conter todos os caminhos encontrados deverão ser mostrados, um por linha.

Outro `DataGridView` com apenas uma linha (chamado `dgvMelhorCaminho`), abaixo do anterior, deverá mostrar o melhor caminho dentre todos os encontrado.

Quando se clicar numa das linhas do `dgvCaminhos` ou na linha do `dgvMelhorCaminho`, o trajeto correspondente a essa linha deverá ser desenhado no mapa, com linhas ligando as cidades de cada ligação do trajeto.

## Descrição dos arquivos

### CidadesMarte.txt

NomeCidade – string, 15 posições  
 CoordenadaX – real, 7 posições  
 CoordenadaY – real, 7 posições

### CaminhoEntreCidadesMarte.txt

nomeCidadeOrigem - string, 15 posições  
 nomeCidadeDestino - string, 15 posições  
 distancia – inteiro, 5 posições  
 tempo – inteiro, 4 posições  
 custo – inteiro, 5 posições

## IMPORTANTE

- Trabalho feito **pela mesma dupla que fez a parte 1 do projeto**;
- Desenvolver em C# no Visual Studio;
- Comentar adequadamente o programa e o código programado;
- Nomear os identificadores de forma adequada;
- No início dos arquivos fonte, digitar comentário com os RAs e nomes dos alunos;
- **Relatório de desenvolvimento** deve ser feito num arquivo cujo nome é:  
 RA1\_RA2\_RelatorioProjeto2ED.PDF (exemplo: 23101\_23192\_RelatorioProjeto1ED.pdf).  
 Deve conter imagens da execução da busca de caminhos;
- O relatório **deve** ser entregue em formato **PDF**;
- Entrega: **18/06/2024 (29 dias)**, pelo Google Classroom
- Material a ser entregue: **pasta do projeto, arquivos de dados e PDF compactados em um único arquivo, cujo nome será RA1\_RA2\_Projeto2ED.rar** (23101\_23192\_Projeto2ED.rar, por exemplo).

## Fontes de informação:

<https://www.deviantart.com/axiaterraartunion/art/Mars-Political-Map-294675891>  
<http://www.teses.usp.br/teses/disponiveis/8/8147/tde-21102009-170528/pt-br.php>  
[https://en.wikipedia.org/wiki/Mars\\_trilogy](https://en.wikipedia.org/wiki/Mars_trilogy)



