

**IFES - Campus Serra**

**30/10/2022**

## **Relatório da Primeira Etapa do Trabalho de Grafos**

Arthur Santos Miguel

Cleber de Jesus Salustiano

Serra - ES  
2022

## Estrutura

A estrutura escolhida para representar os gráficos foi a Lista de Arestas, devido ao fato de ser uma estrutura que facilita o acompanhamento das operações realizadas com o grafo, dado que seus elementos são divididos explicitamente entre arestas e vértices.

Essa divisão de classes é semelhante à Lista de Adjacências, mas a Lista de Arestas conta com a vantagem de que os dados referentes aos caminhos estão alocados nas arestas em si, que já são naturalmente relacionadas aos caminhos. Por isso consideramos que facilita o entendimento.

Além disso, acreditamos que a Matriz Adjacência não seria tão interessante devido a sua robustez, e grande quantidade de zeros armazenados. E por ver diversas estruturas anteriores, observamos na Lista de Aresta uma similaridade com elas, bem como uma facilidade de implementação e modificação da mesma. Sendo assim, a escolha foi estratégica devido ao melhor entendimento, facilidade de utilização e modificação.

## Estratégia

Para implementar a estrutura de Lista de Arestas, começamos com os códigos apresentados nos slides da matéria. Assim, criamos as classes 'Grafo', 'Aresta' e 'Vertice'. Optamos também por criar uma classe 'Cidades', que armazena o nome e o 'código' da cidade, para melhorar a aplicação do Menu - assim deixamos o código mais limpo e legível.

O programa principal cria um grafo, e, lendo o arquivo de entrada, cria instâncias da classe 'Cidade' que serão armazenadas no grafo como Vértices. Após isso, lê a matriz de adjacências do arquivo para definir as arestas, seus pesos, origem e destino.

Para a opção de obter as cidades vizinhas, percorremos a lista de arestas, analisando aquelas que possuem como origem a cidade escolhida e possuem peso maior que zero (o que indica que há conexão entre as cidades). A medida que obtemos essas arestas, imprimimos as cidades destino de cada uma.

Na segunda opção, criamos uma função que recebe uma vértice (Cidade) como ponto inicial. Utilizamos então o processo da primeira opção para encontrar as cidades vizinhas, que são guardadas em uma fila. Para cada vértice da fila, encontramos as vértices adjacentes e guardamos na fila aquelas que já não estejam lá e que não foram visitadas ainda (definidas como 'marcadas'). Com isso, conseguimos visualizar as cidades adjacentes à cidade escolhida, as adjacências dessas cidades adjacentes e assim por diante.