## Estrutura de Dados Atividade I - 31/03/2025

Prof. Dr. Antonio Marcos Selmini

### Leia atentamente as instruções:

- 1. Atividade pode ser realizada em duplas. É necessário que todos os alunos estejam em algum grupo para que o feedback seja cadastrado corretamente. Os alunos que fizerem em dupla deverão obrigatoriamente estar em um grupo.
- 2. Na resolução dos exercícios só poderão ser utilizadas estruturas e conteúdo apresentados durante as aulas. Se for utilizado conteúdo não apresentado, o(s) exercício(s) terá(ão) valor ZERO.
- 3. A data limite para entrega é dia 15/04/2025 às 23h59.
- 4. Os arquivos deverão ficar em um repositório do github e você deverá enviar o link do repositório na entrega de atividades. Após a finalização da atividade os arquivos no repositório não poderão ser alterados.
- 5. Atividades não serão aceitas fora do prazo e fora do local de entrega (você leu bem essa parte?).

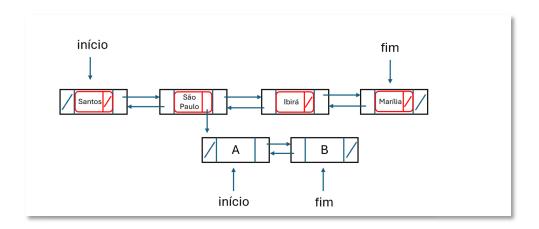
A empresa **ExpressLine** opera entregas entre cidades e deseja modelar sua rede de conexões diretas. Cada cidade poderá ter **zero ou mais conexões diretas** com outras cidades. Cada **ligação direta entre cidades** possui:

- ✓ Nome da cidade de destino (String)
- ✓ Distância em quilômetros (double)
- ✓ Fator de tráfego (double): valor entre 0 e 2 (0 indica que não tem tráfego e 2 tráfego muito intenso).
- ✓ Número de pedágios (int)

O tempo estimado de entrega é calculado como: tempo = (distância \* fator\_tráfego) + (número\_de\_pedágios \* 2)

O sistema a ser implementado deverá fazer uso de duas listas lineares duplamente encadeada e parametrizada (tipo genérico). A primeira lista também chamada de lista principal deverá armazenar em cada nó o nome de uma cidade e uma referência para a lista contendo as cidades que estão ligadas diretamente a cidade armazenada na lista principal. A figura a seguir mostra de forma ilustrativa a ideia geral do sistema.

Na figura, a lista principal é composta pelas cidades Santos, São Paulo, Ibirá e Marília. Cada elemento armazenado na lista deve ser obrigatoriamente um objeto composto pelo nome da cidade e uma referência para a lista de ligações diretas. As cidades: Santos, Ibirá e Marília não tem nenhuma lista de ligações diretas representadas, mas a cidade de São Paulo tem (cidades A e B ligadas diretamente).



#### Funcionalidades do sistema:

O sistema deve funcionar obrigatoriamente a partir de um menu de opções utilizando janelas gráficas (classe **JOptionPane**) para entrada e saída de dados. Você deverá adaptar a classe **ListaDupla** criada em sala de aula para essa nova aplicação e, em seguida, gerar o arquivo JAR que deverá ser utilizado no seu projeto. Lembrando que a classe **ListaDupla** deve ser totalmente genérica e não fazer referência a nenhum tipo de dado específico e a nenhuma funcionalidade específica de nenhum projeto. A classe deve ser útil para qualquer projeto.

O seguinte menu de opções deve ser exibido para o usuário da aplicação:

- 1. Cadastrar cidades na lista principal.
- 2. Cadastrar ligações diretas entre cidades.
- 3. Listar todas as cidades com suas ligações diretas.
- 4. Dado o nome de uma cidade de origem e uma cidade de destino, verificar se existe uma ligação direta e, se sim, exibir o tempo estimado da entrega.
- 5. Dado um tempo limite (em minutos), exibir todas as **ligações diretas em toda a malha** que possam ser realizadas dentro desse tempo.

#### Exemplo de uso (simulado no código):

Cidade: São Paulo

→ Campinas | Distância: 90.0 | Tráfego: 1.2 | Pedágios: 2 | Tempo: 110.00 min

Cidade: Campinas

→ Ribeirão Preto | Distância: 180.0 | Tráfego: 1.1 | Pedágios: 4 | Tempo: 206.00 min

#### Exemplo de consulta:

Existe ligação direta entre São Paulo e Campinas? Sim.

Tempo estimado: 110.0 minutos.

#### Consulta com tempo máximo:

Entregas possíveis com tempo ≤ 120 minutos:

São Paulo → Campinas (110.0 min)

# SPM | ESPM | ESP

#### Restrições:

- ✓ Não utilizar bibliotecas prontas como ArrayList, LinkedList, HashMap, etc.
- ✓ Toda a manipulação de dados deve ser feita utilizando **estruturas implementadas manualmente**.
- ✓ O projeto deve utilizar composição de listas: uma cidade *possui* uma lista de ligações.

#### Pontuação

Os itens serão avaliados como: "certo", "meio certo" ou "errado"

| Critério   | Pontuação |
|--|-----------|
| Cadastrar cidades na lista principal   | 2,0       |
| Cadastrar ligações diretas entre cidades.  | 2,0       |
| Listar todas as cidades com suas ligações diretas.   | 2,0       |
| Dado o nome de uma cidade de origem e uma cidade de destino, verificar se existe uma ligação direta e, se sim, exibir o tempo estimado da entrega. | 2,0       |
| Dado um tempo limite (em minutos), exibir todas as ligações diretas em toda a malha que possam ser realizadas dentro desse tempo.                  | 2,0       |

O não cumprimento dos itens básicos para a implementação da lista acarretará nota Zero:

- ✓ Não utilização de janelas gráficas.
- ✓ Não implementação da lista dupla.
- ✓ Não utilização do arquivo JAR gerado a partir da sua implementação de lista duplamente encadeada.
- ✓ Utilização de classes de estrutura de dados da linguagem Java (classes do pacote java.util).