



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Instituto de Ciências Exatas e Informática  
Prática sobre Listas Duplamente Encadeadas

Curso : *Engenharia de Software*  
Disciplina : *Algoritmos e Estruturas de Dados II*  
Professora : *Eveline Alonso Veloso*

### **Regras Básicas:**

1. Estude bastante cada par de entrada/saída fornecido.
2. Todos os programas deverão ser desenvolvidos na linguagem de programação Java.
3. Essas práticas poderão ser desenvolvidas em grupos de, no máximo, três integrantes.
4. Cópias, se existirem, serão encaminhadas ao colegiado de coordenação didática do curso.
5. Fique atento ao *charset* dos arquivos de entrada e saída. Recomenda-se a utilização dos métodos da classe MyIO.java para **leitura de dados do teclado**. É necessário definir o *charset* a ser utilizado antes de começar a leitura de dados do teclado, da seguinte forma: **MyIO.setCharset("UTF-8")**.
6. As saídas esperadas, cadastradas no VERDE pela professora, foram geradas empregando-se: **System.out.println()**.
7. Em cada submissão, vocês devem enviar apenas um arquivo (.java). Essa regra será necessária para a submissão de exercícios no VERDE e no identificador de plágios utilizado na disciplina.
8. A resolução (código) de cada exercício deverá ser submetida ao VERDE.
9. A execução do código submetido será realizada automaticamente pelo VERDE, mas o código será analisado e validado pela professora.

### **Base de Dados:**

A *National Basketball Association* (em português: Associação Nacional de Basquetebol; abreviação oficial: NBA) é a principal liga de basquetebol profissional da América do Norte. Com 30 franquias como membros (29 nos Estados Unidos e 1 no Canadá), a NBA também é considerada a principal liga de basquete do mundo. É um membro ativo da *USA Basketball* (USAB), que é reconhecida pela FIBA (a

Federação Internacional de Basquetebol) como a entidade máxima e organizadora do basquetebol nos Estados Unidos. A NBA é uma das 4 '*major leagues*' de esporte profissional na América do Norte. Os jogadores da NBA são os esportistas mais bem pagos do mundo, por salário médio anual.

A liga foi fundada na cidade de Nova Iorque, em 6 de junho de 1946, como



redor do mundo, além de vários dos

a *Basketball Association of America* (BAA). Adotou o nome de *National Basketball Association* em 1949, quando se fundiu com a rival *National Basketball League* (NBL). A liga tem diversos escritórios ao

próprios clubes fora da sede principal na *Olympic Tower*, localizada na Quinta Avenida, 645. Os estúdios da *NBA Entertainment* e da *NBA TV* estão localizados em Secaucus, New Jersey.

Esse semestre, você precisará ler, organizar e armazenar os dados de jogadores da liga de basquete norte-americana – NBA – extraídos do [site https://www.kaggle.com/drgilermo/nba-players-stats](https://www.kaggle.com/drgilermo/nba-players-stats). Nossa base contém registros de jogadores desde 1950, um total de 67 temporadas da NBA. O arquivo original sofreu algumas adaptações para ser utilizado neste e nos próximos trabalhos práticos da disciplina.

## **Exercício:**

### **1. Lista duplamente encadeada**

Utilize a classe *Jogador* especificada e desenvolvida em prática anterior.

Crie uma **lista duplamente encadeada** de objetos da classe *Jogador*. Neste exercício, faremos **inserções** e **remoções** de itens na lista e, após o processamento de todas as operações, mostraremos seus elementos.

Os métodos de sua lista duplamente encadeada devem operar conforme descrito a seguir, respeitando-se parâmetros e tipos de retorno:

- Sua classe *ListaDuplamenteEncadeada* deverá ter, pelo menos, **um construtor**.
- *void inserirInicio (Jogador jogador)*: insere um objeto do tipo *Jogador* na primeira posição da lista.
- *void inserir (Jogador jogador, int posicao)*: insere um jogador na posição da lista indicada pelo parâmetro *posicao*, desse método; onde  $0 \leq posicao \leq n$ , sendo  $n$  o número de jogadores já inseridos na estrutura.
- *void inserirFim (Jogador jogador)*: insere um objeto da classe *Jogador* na última posição da lista.
- *Jogador removerInicio()*: remove e retorna o primeiro jogador da lista.
- *Jogador remover(int posicao)*: remove e retorna o objeto *Jogador* armazenado na posição da lista indicada pelo parâmetro *posicao*, desse método; onde  $0 \leq posicao < n$ , sendo  $n$  o número de jogadores já inseridos na estrutura.

- *Jogador* **removerFim**(): remove e retorna o último *Jogador* da lista.
- *void* **mostrar** (): para todos os objetos do tipo *Jogador* presentes na lista, exibe a posição do objeto na lista seguida dos valores de seus atributos (observe o formato de cada linha da saída esperada).

Seu programa deve ler um arquivo-texto chamado **jogadores.txt** que, no VERDE, **localiza-se na pasta /tmp**. Você deve preencher um vetor de objetos da classe *Jogador* com os dados dos diversos jogadores da NBA informados nesse arquivo. Atenção para os dados de entrada, pois em alguns registros faltam valores e esses devem ser substituídos pela *string* "nao informado", na saída padrão.

Cada uma das linhas presentes no arquivo indica os dados de um jogador, separados pelo símbolo `,'. Esses dados são, nessa ordem:

- *id* do jogador;
- nome do jogador;
- sua altura;
- seu peso;
- universidade que o jogador representa;
- ano de nascimento do jogador;
- nome da cidade em que o jogador nasceu;
- estado em que o jogador nasceu.

Depois, seu programa deve processar a entrada padrão, que é dividida em **duas partes**. A **primeira** contém, em cada linha, uma *string* indicando o ***id*** do jogador que deve ser **inserido no final** da lista de jogadores, na ordem em que são apresentados.

Após a palavra FIM, inicia-se a segunda parte da entrada padrão.

A primeira linha dessa **segunda parte da entrada padrão** apresenta um número inteiro *n* indicando a **quantidade de jogadores que serão** em seguida **inseridos ou removidos** da lista. Nas próximas *n* linhas, tem-se *n* **comandos de inserção ou remoção** que devem ser processados neste exercício. Cada uma dessas linhas tem uma palavra de comando, conforme descrito a seguir:

- II: inserir no início;
- I\*: inserir em uma determinada posição;
- IF: inserir no final;
- RI: remover do início;
- R\*: remover de uma determinada posição; e
- RF: remover do final.

No caso dos comandos de **inserção**, temos também uma *string* indicando o ***id*** do jogador que **deve ser inserido** na lista de jogadores.

No caso dos comandos de **inserção e remoção** “em uma determinada **posição**”, temos também um inteiro indicando essa **posição**. No comando de inserção, a posição fica imediatamente após a palavra de comando. Lembre-se que o primeiro item da lista encontra-se na posição 0.

A **saída padrão** deve apresentar uma linha para **cada jogador removido**, sendo que essa informação será constituída pela *string* “(R)” seguida do atributo *nome* do jogador retirado da lista.

Em seguida, teremos, ainda na saída padrão, os atributos relativos aos jogadores **presentes na lista duplamente encadeada após o processamento de todas as operações de inserção e remoção** (observe o formato de cada linha da saída esperada).