

# Trabalho Prático

---

## Regras Básicas

1. extends Trabalho Prático 01
2. Fique atento ao Charset dos arquivos de entrada e saída.

## Classe + Registro



Você foi contratado para trabalhar em uma empresa que distribui *stream* de séries de TV na web. Sua tarefa é organizar as informações das séries disponíveis para exibição ao usuário. Entretanto, esses dados estão espalhados em vários arquivos no formato *html*, os quais foram obtidos através de consultas à base de dados Wikipedia. Todos esses arquivos estão agrupados no arquivo *series.zip*, e o mesmo deve ser descompactado na pasta */tmp/*.<sup>1</sup> Para isso, você deve ler, organizar e armazenar os dados de cada série em memória, utilizando as estruturas de dados em aula (Lista, Pilhas e Filas). Em seguida executar as operações descritas nos arquivos de entrada. Muito cuidado ao realizar o *parser* do texto. Fique atento a descrição dos dados que serão lidos e manipulados pelo seu sistema.

1. **Classe Séries em Java:** Crie uma classe *Séries* seguindo todas as regras apresentadas no slide *unidade01g\_conceitosBasicos\_introducaoOO.pdf*. Sua classe terá os atributos privados Nome (String), Formato (String), Duração (String), País de origem (String), Idioma original (String), Emissora de televisão original (String), Transmissão original (String), N.º de temporadas (Inteiro), N.º de episódios (Inteiro). Ela terá também pelo menos dois

---

<sup>1</sup>Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta */tmp/*.

construtores, e os métodos gets, sets, clone e imprimir e ler. O método imprimir mostra a String ‘nome formato duracao paisDeOrigem idiomaOriginal emissoraDeTelevisao transmissaoOriginal numeroTemporadas numeroEpisodio’, contendo todos os atributos da classe. O método ler deve efetuar a leitura dos atributos de um registro. Veja que os dados estão divididos em vários arquivos.

A entrada padrão é composta por várias linhas e cada uma contém o nome de um arquivo *.html*. A última linha da entrada contém *FIM*. A saída padrão também contém várias linhas, uma para cada registro contido em uma linha da entrada padrão.

2. **Registro em C:** Repita a anterior criando o registro *Série* na linguagem C.

## Pesquisa

3. **Pesquisa Sequencial em Java:** Faça a inserção de alguns objetos no final de uma Lista e, em seguida, faça algumas pesquisas sequenciais. A chave primária de pesquisa será o atributo **nome**. A entrada padrão é composta por duas partes onde a primeira é igual a entrada da primeira questão 1. As demais linhas correspondem a segunda parte. A segunda parte é composta por várias linhas. Cada uma possui um elemento que deve ser pesquisado na Lista. A última linha terá a palavra FIM. A saída padrão será composta por várias linhas contendo as palavras SIM/NÃO para indicar se existe cada um dos elementos pesquisados. Além disso, crie um arquivo de log na pasta corrente com o nome matrícula\_sequencial.txt com uma única linha contendo sua matrícula, tempo de execução do seu algoritmo e número de comparações. Todas as informações do arquivo de log devem ser separadas por uma tabulação ‘\t’.
4. **Pesquisa Binária em Java:** Repita a questão anterior, contudo, usando a Pesquisa Binária. A entrada e a saída padrão serão iguais às da questão anterior. O nome do arquivo de log será matrícula\_binaria.txt.