

***Universidade Federal de Sergipe***

***Campus Universitário Prof. Alberto Carvalho***

***Departamento de Sistemas do Campus de Itabaiana***

***Itabaiana – Sergipe***

**Docente: Doutor Alcides Xavier Benicasa.**

**Discentes: Erick Oliveira Santos**

**Luiz Carlos Moitinho**

**Rafael Silveira de Andrade**

**Wesley Andrade Silva**

**Wedson Teixeira de Almeida**

**Disciplina: Estrutura de Dados I**

**Manual do usuário**

***i*Films**

**Itabaiana - SE, 30 de Março de 2019**

**1 - INTRODUÇÃO**

Estruturas de dados, baseando-se em uma abstração do mundo real, permitem diversas formas de manipular dados, ou seja, organizá-los da melhor maneira de acordo com a necessidade para que seja aplicada em uma situação específica. Sendo assim, a forma com que os dados estão dispostos em um software permite a otimização de processos e melhorias em desempenho, por isso, as estruturas encontram-se em muitas aplicações de desenvolvimento de sistemas, sendo que algumas são altamente especializadas e utilizadas em tarefas específicas.

Por meio dessa forma de manipular informações podemos administrar uma grande quantidade de dados de forma eficiente, como aplicações em bancos de dados ou serviços de busca e indexação de dados, e no desenvolvimento de algoritmos eficientes.

O presente trabalho, o qual permite a utilização da Estruturas de Dados e Mecanismo de Ordenação, foi proposto pelo Dr. Alcides Xavier Benicasa, professor da disciplina Estrutura de Dados, da Universidade Federal de Sergipe, campus Professor Alberto Carvalho.

**2 - OBJETIVO**

Utilizar os conceitos adquiridos na disciplina de Estrutura de Dados I, como estruturas de dados e métodos de ordenação dos mesmo, e utilizá-los em uma aplicação funcional, onde foi possível visualizar o poder computacional que possuem, de forma a analisar as melhores formas de produzir algoritmos em relação ao custo de processamento, em que é possível gerenciar e ordenar os dados de um catálogo de filmes variados. Estes que, por sua vez, estão distribuídos de forma desordenada com base em suas características, como, gênero, ano, título, popularidade e nota.

O foco da aplicação é utilizar as estruturas e os métodos de ordenação de dados para organizar e realizar as pesquisas que o usuário necessitar na base de dados. Esta foi desenvolvida na linguagem de programação Java, já que é bastante difundida no mercado de trabalho, por possuir uma extensa variedade de ferramentas úteis a resolução dos problemas.

**3 - CONCEITOS**

**3.1 - Lista Duplamente Encadeada**

Elas são estruturas de dados semelhantes às listas simplesmente encadeadas e sua alocação da memória é feita durante a execução. No entanto, em comparação com as listas simplesmente encadeadas, a conexão entre os elementos é feita através de dois ponteiros (um que aponta para o elemento anterior e o outro para o seguinte): o ponteiro anterior ao primeiro elemento deve apontar para NULL (identificando o início da lista); o ponteiro seguinte ao último elemento deve apontar para NULL (o fim da lista).

**3.2 - Métodos de Ordenação**

Quando trabalhamos com listas, existem ocasiões em que necessitamos ordená-las para facilitar as pesquisas. Podemos ordenar os valores de uma matriz (ou banco de dados) do mais baixo para o mais alto (ordem crescente) ou ainda mais alto para o mais baixo (ordem crescente). Sem esse tipo de ordenação toda e qualquer pesquisa em uma estrutura de dados seria muito difícil e demorada.

**3.2.1 - Bubble Sort**

O algoritmo Bubble Sort possui um custo de tempo e processamento - fatores que programadores devem atentar-se para que sua aplicação se torne mais eficiente - que em situações onde possuem pouca quantidade de dados. Apesar de simples, não deve ser utilizado com matrizes ou listas muito extensas para evitar lentidão no processamento.

Seu funcionamento é muito simples.

Esse método faz um laço pelos valores da matriz, comparando-os e movendo o maior para a posição anterior. Este método cria uma ordenação decrescente. Para criar uma ordenação crescente, o algoritmo deverá mover o maior valor para a posição posterior, após o elemento testado.

**3.2.2 - MergeSort (“Dividir e conquistar”)**

A ideia básica do MergeSort é criar uma sequência ordenada a partir de duas outras também ordenadas. Para isso, o algoritmo Merge Sort divide a sequência original em pares de dados, agrupa estes pares na ordem desejada. Depois as agrupa as sequências de pares já ordenados, formando uma nova sequência ordenada de quatro elementos, e assim por diante, até ter toda a sequência ordenada.

Algoritmo implementado por meio de uma chamada recursiva em que é passado os parâmetros início, meio e fim da lista, nas duas chamadas os parâmetros passados são alterados a cada chamada.

O Merge é mais rápido e eficiente quando é utilizado sobre uma grande quantidade de dados, porém como ele é um método de natureza recursiva, há um alto consumo de memória.

**3.3 - Métodos de Busca**

Se uma base de dados está ordenada, nada melhor que aplicar os métodos corretos de pesquisa a mesma. E os algoritmos para pesquisa são muitos. Porém é possível destacar os dois principais e mais utilizados: Busca/Pesquisa Sequencial e Busca/Pesquisa Binária.

**3.3.1 - Busca Sequencial**

Neste algoritmo de pesquisa é comparado a chave pesquisada com os elementos da lista partindo do início até o fim da lista. Quando o elemento é encontrado a pesquisa é finalizada e o elemento retornado, se toda a lista for percorrida e o elemento não for encontrado é retornado um valor para representar esse caso, por exemplo: -1.

Quanto a complexidade no melhor caso a quantidade de comparações é 1, o pior caso é n (quantidade de elementos) e a média é definida por (melhorCaso + piorCaso)/2.

**3.3.2 - Busca Binária**

Caso você possua uma lista com os valores ordenados de forma numérica ou alfabética, é possível fazer uma busca muito mais rápida. Ao invés de pesquisar a lista em sequência, uma busca binária começará examinando o item do meio. Se esse item for aquele que estamos procurando, a busca termina.

Se não for o item correto, podemos usar a natureza ordenada da lista para eliminar a metade dos itens restantes. Se o item que procuramos for maior do que o item do meio, sabemos que toda a metade inferior da lista, bem como o item do meio, podem ser eliminados de uma consideração mais aprofundada. O item, se estiver na lista, deve estar na metade superior.

Podemos então repetir o processo com a metade superior. Comece no item do meio e compare-o contra o que estamos procurando. Novamente, encontramos ou dividimos a lista pela metade, eliminando assim uma grande parte do nosso possível espaço de busca

A complexidade é de ordem O(log2 n), em que n é o tamanho do vetor de busca. Assim ela apresenta-se mais eficiente que a busca sequencial cuja ordem é O(n).

**4 - DETALHES DA APLICAÇÃO**

iFilms foi desenvolvida em Java uma [linguagem de programação](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o) [orientada a objetos](https://pt.wikipedia.org/wiki/Orienta%C3%A7%C3%A3o_a_objetos) desenvolvida na [década de 90](https://pt.wikipedia.org/wiki/D%C3%A9cada_de_1990) por uma equipe de programadores chefiada por [James Gosling](https://pt.wikipedia.org/wiki/James_Gosling), na empresa [Sun Microsystems](https://pt.wikipedia.org/wiki/Sun_Microsystems). Diferente das linguagens de programação que são [compiladas](https://pt.wikipedia.org/wiki/Compilador) para [código nativo](https://pt.wikipedia.org/wiki/C%C3%B3digo_nativo), a linguagem Java é compilada para um [*bytecode*](https://pt.wikipedia.org/wiki/Bytecode_Java) que é interpretado por uma [máquina virtual](https://pt.wikipedia.org/wiki/M%C3%A1quina_virtual_Java) (Java Virtual Machine, mais conhecida pela sua abreviação JVM).

A aplicação conta com 13 classes desenvolvidas na linguagem de programação Java, entre elas as mais importantes para o manipulação dos dados são: Arquivo, ConverteTipo, TLista, TNodo, TInfo.

A classe Arquivo possui métodos para carregar a base de dados em uma lista, salvar uma lista em um arquivo txt e carregar uma lista já salva, também há métodos para carregar e gravar o arquivo de usuários. Possui três métodos públicos, o primeiro para carregar os dados dos filmes em uma Lista, esse método é responsável por escolher qual arquivo irá carregar o arquivo original com os dados não tratados ou um arquivo com os dados tratados, a escolha acontece da seguinte maneira se o arquivo com os dados tratados existe ele é carregado senão é carregado o arquivo original e ao final do carregamento do arquivo original é chamando um método que salva os dados tratados, assim criando o arquivo com os dados tratados. O segundo método público é utilizado para carregar o arquivo de pessoas para uma lista e o terceiro para salvar uma lista de pessoas em um arquivo.

A classe ConverteTipo possui métodos utilizados para conversão de Strings em tipos de dados primitivos (int, long e float).

A classe TInfo tem seus objetos utilizados para armazenar os dados referentes a um filme, estes dados são: Id, Budget, Genres, HomePage, Keywords, Original Language, Original Title, Overview, Popularity, Production Companies, Production Countries, Release Date, Revenue, Runtime, Spoken Languages, Status, Tagline, Title, Vote Average, Vote Count.

A classe TNodo instancia objetos que são nodos da lista, possuindo o atributo próximo que armazena o endereço do próximo nodo, anterior que armazena o endereço do nodo anterior e um ponteiro para um TInfo, no qual os objetos armazenam os dados dos filmes conforme citado anteriormente.

A classe TLista como atributo, o nodo Primeiro que é um TNodo responsável por armazenar o endereço do primeiro elemento da Lista e último que também é um TNodo e é responsável por armazenar o endereço do último elemento da Lista. Os métodos disponíveis na classe são os referentes a realizar operações de inserção, remoção, impressão, ordenação e pesquisa. Os algoritmos de ordenação utilizados são o BubbleSort e o MergeSort. Quanto a pesquisa estão disponíveis métodos que permitem realizar pesquisas tanto sequencial quanto binária.

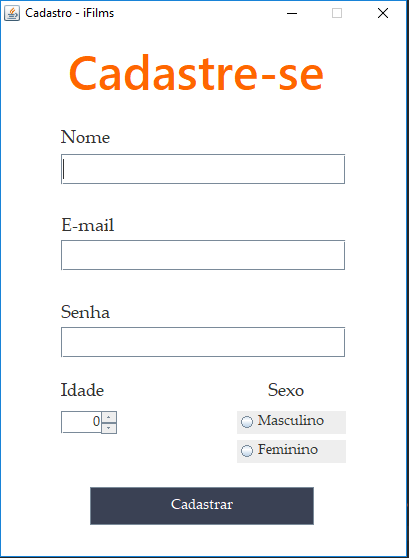
**5 - TELAS**

O *i*Films é uma base de dados disponibilizada pela IMDB com mais de 5 mil filmes disponíveis para pesquisa. Para ter acesso basta apenas inserir o email e senha de sua posse cadastrados no sistema na tela de login. (Imagem 1).



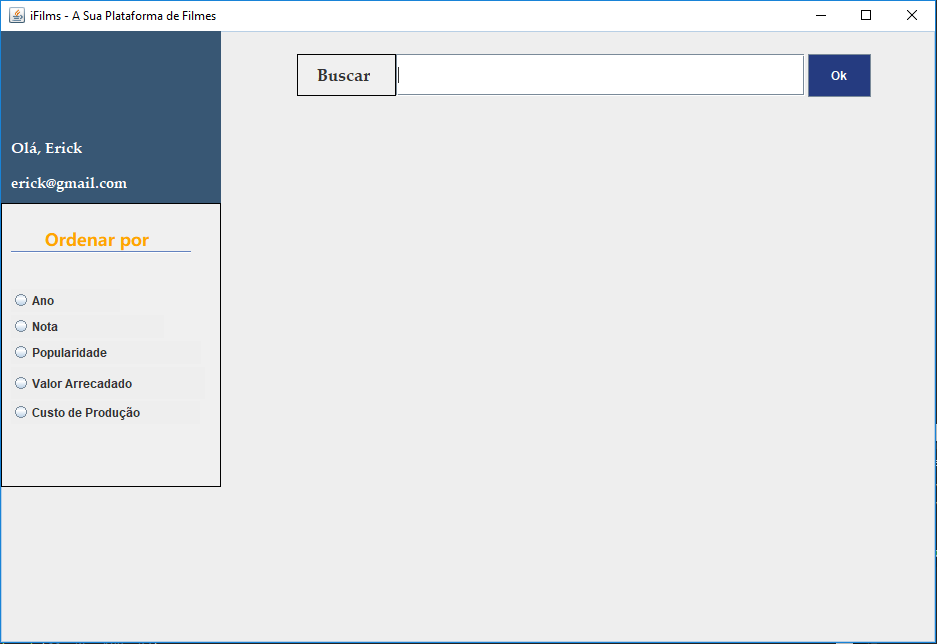
[Imagem 1. Tela de Login]

Caso você não possua uma conta, é necessário cadastrar-se na plataforma e inserir algumas informações de acesso na tela de cadastro, como nome, idade, email e senha (Imagem 2).



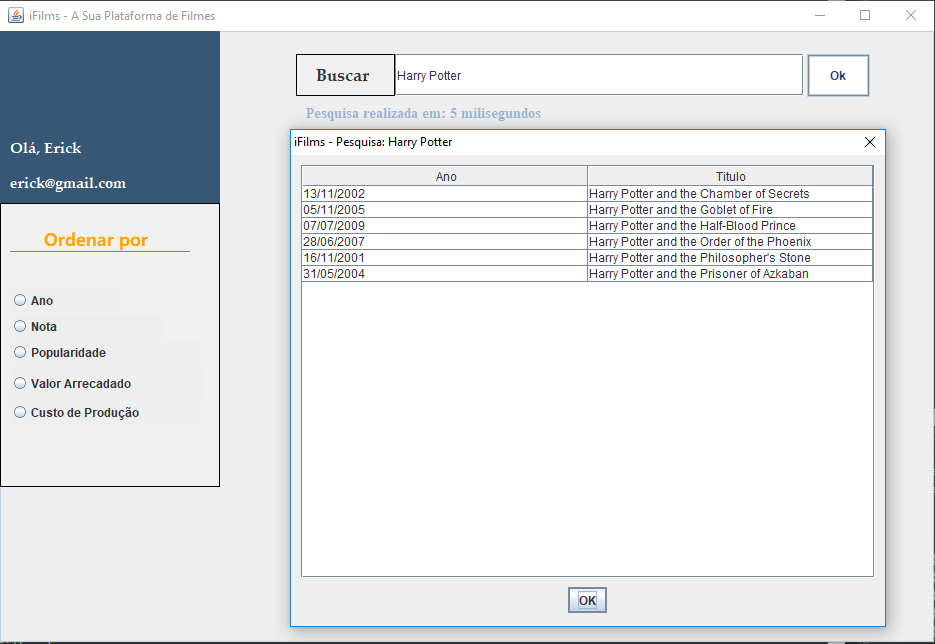
[Imagem 2. Tela de Cadastro]

Quando realizar o cadastro e seus dados forem válidos, você terá acesso a página principal da plataforma. Que contém alguns detalhes do seu cadastro, a barra de pesquisa de filmes, e os filtros de filmes disponíveis para uso (Imagem 3).



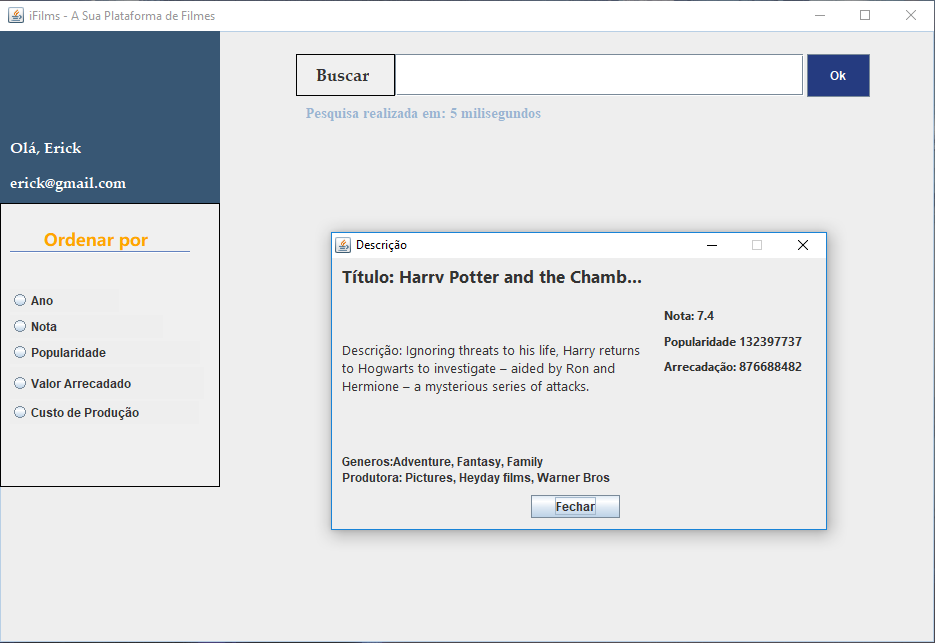
[Imagem 3. Tela de Início]

Através da barra de pesquisa podemos pesquisar filmes pelo seu nome, com isso, uma busca será realizada na base de dados e retornará filmes existentes com base no título do filme que foi digitado, além de informar também a velocidade da pesquisa (Imagem 4).



[Imagem 4. Pesquisa por Título]

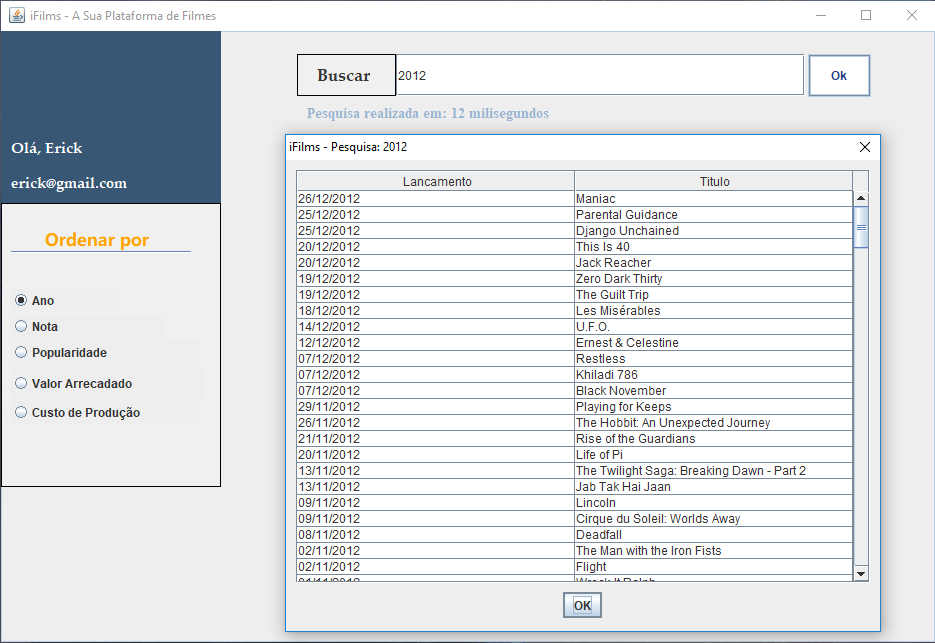
Com a tabela de filmes encontrados pela pesquisa disponível, agora, você poderá selecionar o filme específico ao qual deseja mais informações. Para isso, basta clicar no Título do filme escolhido, e uma nova janela surgirá contendo informações adicionais sobre o filme específico. Selecionando o primeiro filme por exemplo, teremos as informações disponíveis sobre ele, como nota, descrição, produtora entre outros (Imagem 5).



[Imagem 5. Janela de Descrição do Filme]

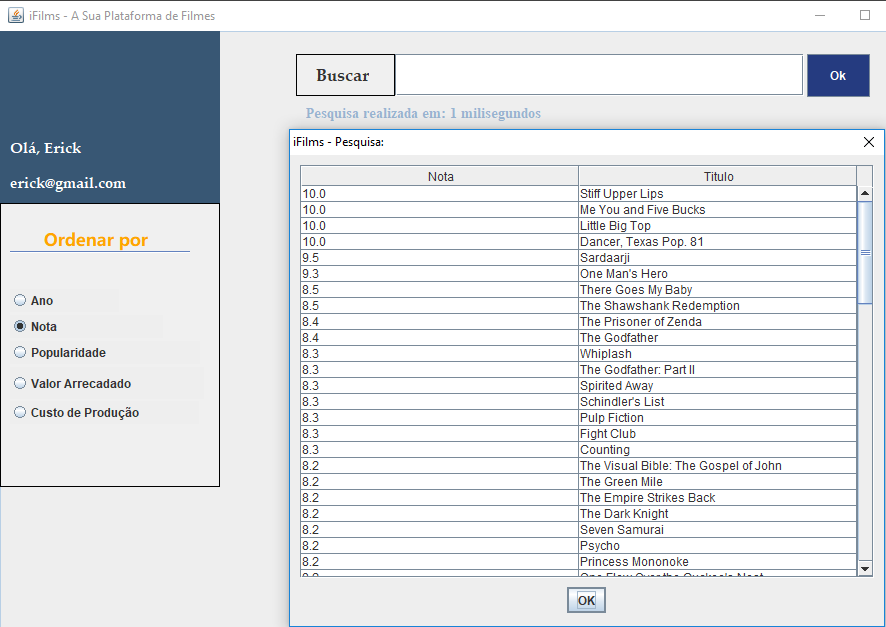
Caso você queria ver outro filme da sua lista pesquisada, apenas clique em “Fechar” e sua lista de filmes surgirá novamente.Porém, na plataforma, é possível também, além da pesquisa básica por nome, realizar também a ordenação dos filmes pesquisados, é possível ordenar por Ano de lançamento, Nota do filme, a Popularidade, o Valor Arrecadado (Bilheteria do filme) e o Custo de Produção.

Ao pesquisar por Nota por exemplo, você poderá selecionar o ano da lista de filmes que será mostrada. Por exemplo, ao pesquisar “2012” e selecionar a opção “Ano”, serão mostrados todos os filmes da base de dados que foram lançados em 2012, do último ao primeiro lançamento do ano (Imagem 6).



[Imagem 6. Método de Ordenação]

A ordenação por Nota seguirá um princípio parecido. Ao selecionar a opção nota, dois campos surgirão pedindo para informar o limite entre as notas. Por exemplo, ao definir um limite de notas entre 8.0 e 10.0, a tabela mostrará os filmes que estão entre as duas notas informadas (Imagem 7).



[Imagem 7. Ordenação por Nota]

Outros métodos como Popularidade, Valor Arrecadado e Custo de Produção seguem o mesmo princípio da pesquisa por nota. Bastando apenas definir o limite de valores entre eles.