



Trabalho

Filмотeca - Coleção de Filmes

Segunda-feira, 1º de junho de 2015.

Desenvolva um aplicativo Java GUI para gerenciar a coleção de filmes do usuário. Este programa deve fornecer os seguintes serviços:

- 1. Cadastrar:** cadastra todos os dados de um filme segundo os dados apresentados na Seção 1 - Modelo lógico do banco de dados.
- 2. Consultar Filme:** permite ao usuário digitar o título do filme, pesquisa na filмотeca e, caso encontre, exibe todos os dados do filme (ver Seção 1). Se o filme não estiver cadastrado deve ser exibido a mensagem Filme não cadastrado.
- 3. Consultar Artista:** permite ao usuário digitar o nome do artista, pesquisa na filмотeca e, caso encontre, exibe os nomes de todos filmes em que o artista trabalha. Se o artista não estiver cadastrado deve ser exibido a mensagem Artista não cadastrado.
- 4. Consultar Diretor:** permite ao usuário digitar o nome do diretor, pesquisa na filмотeca e, caso encontre, exibe os nomes de todos filmes dirigidos por este diretor. Se o diretor não estiver cadastrado deve ser exibido a mensagem Diretor não cadastrado.
- 5. Consultar Autor:** permite ao usuário digitar o nome do autor, pesquisa na filмотeca e, caso encontre, exibe os nomes de todos filmes cujo roteiro foi escrito por este autor. Se o autor não estiver cadastrado deve ser exibido a mensagem Autor não cadastrado.
- 6. Consultar Gênero:** permite ao usuário digitar o gênero do filme, pesquisa na filмотeca e, caso encontre, exibe os nomes de todos filmes deste gênero. Se não existir nenhum filme cadastrado com este gênero deve ser exibido a mensagem Não há nenhum filme cadastrado neste gênero.
- 7. Os melhores filmes:** exibe a relação de todos os filmes cadastrados na filмотeca em ordem decrescente pela classificação pessoal.

1. Modelo lógico do banco de dados

Notação: chave primária sublinhada; chave estrangeira precedida por cerquilha (#).

1. Filme(CódigoFilme, Título, Duração, Ano, DataLançamento, Sinopse, ClassificaçãoEtária, ClassificaçãoIMDB, ClassificaçãoPessoal, Mídia, Pôster)

Valores válidos

- **Duração:** duração do filme em minutos.
- **Classificação Etária:** Livre, inadequado para menores de 10, 12, 14, 16 e 18.
- **Classificação IMDB:** nota de 1 a 10 do filme na avaliação dos usuários do site IMDB (www.imdb.com).
- **Classificação Pessoal:** nota de 1 a 10 do filme na avaliação do usuário do aplicativo.
- **Mídia:** DVD, Blu-Ray ou Arquivo Digital.
- **Pôster:** imagem de divulgação do filme.

2. Artista(CódigoArtista, Nome)
3. Elenco(#CódigoFilme, #CódigoArtista)
4. Diretor(CódigoDiretor, Nome)
5. DiretorFilme(#CódigoFilme, #CódigoDiretor)
6. Autor(CódigoAutor, Nome)
7. AutorFilme(#CódigoFilme, #CódigoAutor)
8. Gênero(CódigoGênero, Descrição)
9. GêneroFilme(#CódigoFilme, #CódigoGênero)
10. País(CódigoPaís, Nome)
11. PaísFilme(#CódigoFilme, #CódigoPaís)

2. Critérios de avaliação

1. O trabalho será avaliado considerando:

- 1.1 - tratamento dos dados fornecidos pelo usuário e dos cálculos que possam abortar a execução do programa via **tratamento de exceções**;
- 1.2 - a lógica empregada na solução do problema;
- 1.3 - o funcionamento do programa;
- 1.4 - a usabilidade¹ do programa e da interface gráfica com o usuário;

¹ A usabilidade está diretamente ligada a interface e a capacidade do *software* em permitir que o usuário alcance suas metas de interação com o sistema. Ser de fácil aprendizagem, permitir uma utilização eficiente e apresentar poucos erros, são os aspectos fundamentais para a percepção da boa usabilidade por parte do usuário. Mas a usabilidade pode ainda estar relacionada com a facilidade de ser memorizada e ao nível de satisfação do usuário.

- 1.5 - o conhecimento da linguagem de programação;
- 1.6 - o uso do paradigma de orientação a objetos;
- 1.7 - código fonte Java sem erros e sem advertências (*warnings*) do compilador;
- 1.8 - código fonte Java legível, indentado, organizado e comentado;
- 1.9 - documentação do sistema em HTML gerada através da ferramenta *javadoc*.

2. Todos os componentes Swing listados abaixo devem ser utilizados na criação da interface gráfica do programa. Outros componentes Swing poderão ser usados, porque a lista abaixo representa uma relação mínima de componentes GUI.

JDialog, JFrame, JLabel, JButton, JPanel, JTextField, JTextArea, JRadioButton, JCheckBox, JList, JMenu, JMenuItem, JMenuBar, JProgressBar, JPopupMenu, JTabbedPane.

3. O trabalho pode ser desenvolvido em equipe composta por no máximo 2 (dois) membros.
4. A implementação das tabelas e seus relacionamentos deve ser feita usando os recursos do banco de dados **PostgreSQL via JDBC**.
5. Não é permitido o uso de *frameworks*, API, bibliotecas de classes ou qualquer ferramenta que permita, de maneira automática para o programador - sem escrever comandos SQL -, a persistência e a recuperação de objetos em banco de dados relacionais.

3. Artefatos de software

Os seguintes artefatos de *software* devem ser entregues:

- a) o projeto Java desenvolvido na IDE Eclipse;
- b) o arquivo JAR executável da aplicação;
- c) a documentação HTML do sistema Java gerado com o *javadoc*.

Compactar todos os artefatos criando um arquivo ZIP com o nome da equipe.

4. Data e forma de apresentação

- **Segunda-feira, 6 de julho de 2015.**
- Entrevista sobre o desenvolvimento e o funcionamento do sistema.

5. Valor do trabalho

30,0 (trinta) pontos.

Prof. Márlon Oliveira da Silva
marlon.silva@ifsudestemg.edu.br