

Relatório

Sistema de gerenciamento de locadora e loja de filmes **UltraFiction**

Raul Pelegriani Neto Nº USP: 9364906
Renan Rodrigues Nº USP: 9278132

Banco de Dados

O banco de dados utilizado na aplicação foi o PostgreSQL 9.4 pois é amplamente utilizado em diversas linguagens, por isso fácil de conseguir informações e ajuda na internet, além disso o grupo já possuía conhecimentos prévios sobre o uso desse banco de dados. A versão 9.4 foi escolhida pois era a mais recente com driver JDBC (*postgresql-9.4.1208.jre6*) disponível para download. O servidor do banco de dados roda localmente na máquina na porta 5433 (*localhost:5433*). O usuário e senha padrões utilizados por nós são respectivamente: “*postgres*” “*sqladmin*”.

Para gerenciamento, utilizamos a ferramenta pgAdmin III, um programa muito simples e eficiente que faz a comunicação com o PostgreSQL.

O script para geração do banco de dados é o seguinte:

```
CREATE DATABASE "LojaFilmes"
WITH OWNER = postgres
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'Portuguese_Brazil.1252'
LC_CTYPE = 'Portuguese_Brazil.1252'
CONNECTION LIMIT = -1;
```

Foi projetada apenas uma tabela que registra os filmes, não há uma tabela para vendas ou aluguel pois isso é registrado no log do sistema.

A tabela “filmes” é composta por 10 campos que armazenam os dados de cada filme no sistema, esses dados são:

- Código (*primary key*)
- Nome
- Duração
- Gênero
- Ano de lançamento
- Sinopse
- Diretor
- Preço de venda
- Preço de aluguel
- Quantidade de filmes disponíveis

Para gerar uma tabela no banco de dados que correspondesse a essas informações utilizamos o seguinte script SQL:

```
create table filmes (
    codigo serial not null,
    nome varchar(40),
    duracao int,
    genero varchar(20),
    ano_lancamento int,
    sinopse varchar(200),
    diretor varchar(40),
    preco float,
    preco_loc float,
    qntd int,
    primary key(codigo)
);
```

Representação gráfica da tabela gerada:

filmes
<input type="checkbox"/> codigo
<input type="checkbox"/> nome
<input type="checkbox"/> duracao
<input type="checkbox"/> genero
<input type="checkbox"/> ano_lancamento
<input type="checkbox"/> sinopse
<input type="checkbox"/> diretor
<input type="checkbox"/> preco
<input type="checkbox"/> preco_loc
<input type="checkbox"/> qntd

BancoDados.java

O arquivo BancoDados.java possui a classe “tfilmes” que molda a tabela do banco de dados em forma de classe java com seus campos de dados iguais ao da tabela.

A classe BancoDados faz a conexão com o banco de dados e define operações na tabela “filmes” por meio de cláusulas SQL e outras funções auxiliares. Conecta-se ao banco de dados LojaFilmes por meio que uma String de conexão, que contém o driver de conexão ao servidor, seu servidor, o nome do banco de dados e seu login e senha de acesso.

```
// String de conexão
Class.forName("org.postgresql.Driver");
con=DriverManager.getConnection(
    "jdbc:postgresql://localhost:5433/LojaFilmes",
    "postgres", "sqladmin");
```

Faz operações de incluir e excluir filmes, e alterar dados de um filme no banco. Para incluir um filme na tabela utiliza-se o comando *insert*, por exemplo:

```
insert into filmes (nome, duracao, genero, ano_lancamento, sinopse,diretor,
preco, preco_loc, qntd) values('Sangue Negro','158','Drama','2007','Breve
sinopse do filme','Paul Thomas Anderson',37.99,6.50,'10');
```

Para excluir e alterar são diferentes comandos, *delete* e *update* respectivamente. Essa classe basicamente envia esses comandos ao banco de dados que os executa fazendo as alterações que precisamos, se der errado lança uma exceção.

As bibliotecas utilizadas foram:

`import java.sql.*;` Essa é utilizada para as funções que envolvem manipulação do banco de dados e seus recursos.

`import javax.swing.*;` Essa é necessária para lançarmos as possíveis exceções da classe num *MessageBox*.

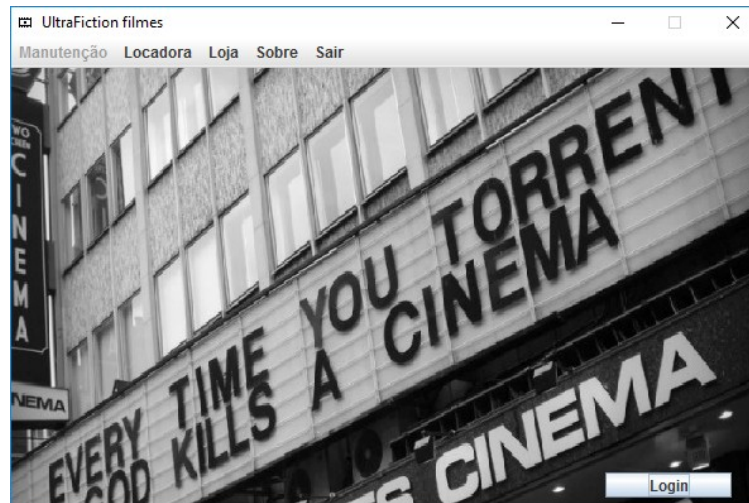
`import java.util.List;`

`import java.util.ArrayList;` As duas são utilizadas em uma função que pega todos os filmes do banco em forma de objeto *tfilmes* e retorna uma *ArrayList* deles.

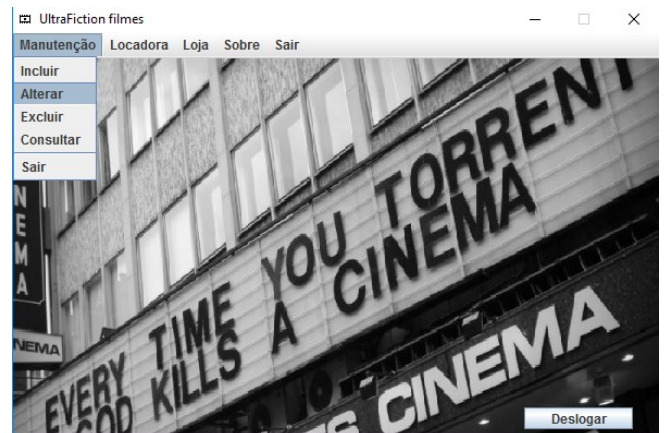
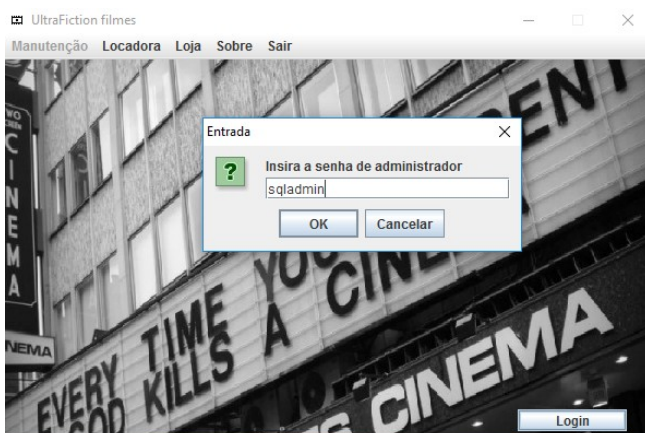
UltraFiction.java

A classe UltraFiction é a tela inicial do sistema, ela dá acesso a todas as demais telas, possui uma configuração para funcionário e para administrador, este último requerendo uma senha de acesso. Há um botão *Login* que clicado dá acesso a uma entrada de dado, onde deve ser digitada a senha de administrador.

O funcionário tem acesso apenas a tela de aluguel e devolução de filmes, venda de filmes e sobre os desenvolvedores. Possui as opções de manutenção desabilitadas.



O administrador pode incluir, alterar dados e excluir qualquer filme, além de ter acesso as telas que o funcionário tem.



As bibliotecas utilizadas foram:

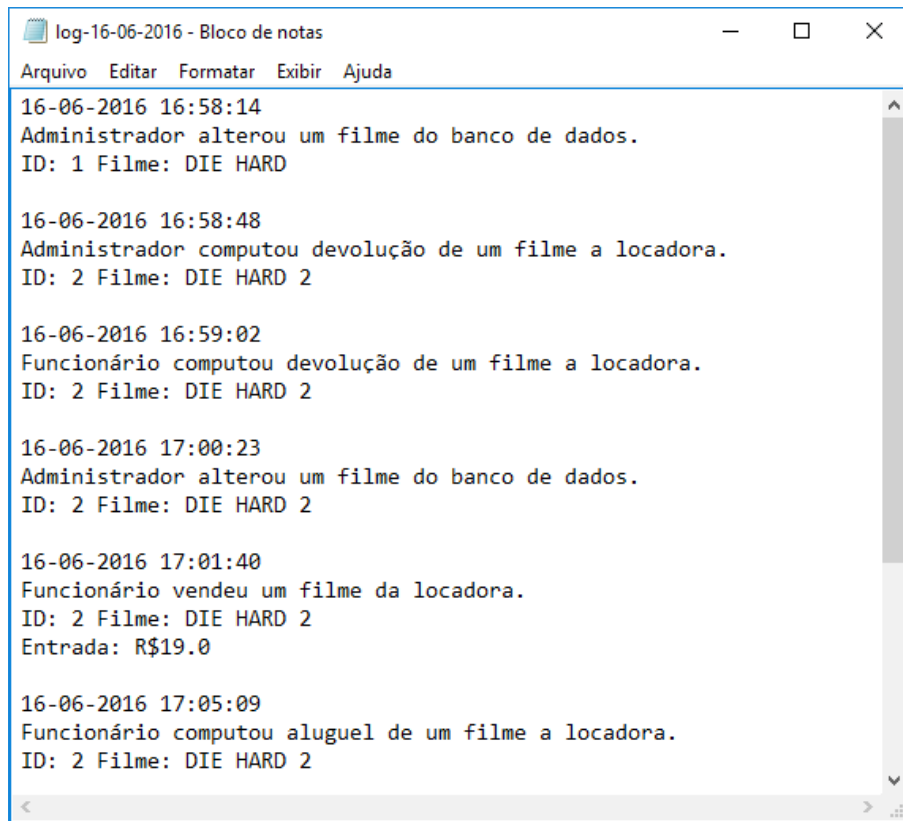
`import java.awt.*;` Utilizado na construção da interface gráfica.

`import java.awt.event.*;` Essa biblioteca é usada para usar eventos dos elementos de interface gráfica na tela, como o clique em botões, no menu e em suas barras.

`import javax.swing.*;` Utilizado na construção da interface gráfica.

Log.java

Essa classe manipula arquivos registrando toda e qualquer atividade do banco de dados feita pelo sistema. A cada dia gera um arquivo chamado log-(data do dia).txt na pasta do projeto e nele serão registrados todas as atividades do dia com a data e hora exatas da atividade, se foi o administrador ou o funcionário que o fez e o que foi feito. Por exemplo:



```
log-16-06-2016 - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda

16-06-2016 16:58:14
Administrador alterou um filme do banco de dados.
ID: 1 Filme: DIE HARD

16-06-2016 16:58:48
Administrador computou devolução de um filme a locadora.
ID: 2 Filme: DIE HARD 2

16-06-2016 16:59:02
Funcionário computou devolução de um filme a locadora.
ID: 2 Filme: DIE HARD 2

16-06-2016 17:00:23
Administrador alterou um filme do banco de dados.
ID: 2 Filme: DIE HARD 2

16-06-2016 17:01:40
Funcionário vendeu um filme da locadora.
ID: 2 Filme: DIE HARD 2
Entrada: R$19.0

16-06-2016 17:05:09
Funcionário computou aluguel de um filme a locadora.
ID: 2 Filme: DIE HARD 2
```

As bibliotecas utilizadas foram:

`import java.util.*;` Necessário para utilização do objeto *Formatter* e *Date*.

`import javax.swing.JOptionPane;` Utilizado para lançamento de possíveis exceções em *MessageBox*.

`import java.io.File;` Manipulação de arquivo, uso do objeto *File*.

`import java.io.FileWriter;` Uso do objeto *FileWriter*, necessário para adicionar mas informações ao final do arquivo em vez de sobrescrever.

`import java.text.SimpleDateFormat;` Utilizado para formatar data e hora que são emitidos no log.

Incluir.java

Essa classe é de acesso apenas de administrador, é acessado por um item do menu Manutenção. Mostra todos os filmes do banco de dados e inclui novos filmes enviando um comando SQL *insert* para o banco de dados executar. Na tela além da tabela há *JTextFields* para entrada de dados para registrar novos filmes.

Processo de inserção:

The screenshots show a Java Swing window titled "Registrar filmes". It contains a table with columns: Código, Nome, Duração, Gênero, Ano, Sinopse, Diretor, Preço, Aluguel, and Em estoque. To the right of the table are input fields for each column. The first screenshot shows the initial state with empty input fields. The second screenshot shows the input fields filled with data for "Toy Story". The third screenshot shows the same data after a click, with the input fields now containing the values from the table.

Código	Nome	Duração	Gênero	Ano	Sinopse	Diretor	Preço	Aluguel	Em estoque
1	Homem Fô...	117	Ficção Cien.	2015	Hank Pym...	Peyton Reed	30.5	21.2	1
2	Os Vingado...	143	Ação	2012	Nick Fury, d...	Joss Whed...	30.99	19.99	1
3	O Regresso...	156	Fantasia / ...	2015	Um explora...	Alejandro G...	35.9	25.9	1
4	Toy Story	81	Aventura / C...	1995	Woody é o ...	John Lasse...	24.99	15.50	1

As bibliotecas utilizadas foram:

import java.awt.event.*; Para tratar os eventos da tela como cliques em no botão ou na tabela.

import javax.swing.*; Necessário para uso dos objetos que compõe a tela como *JButton*, *JTextField* e outros.

import javax.swing.table.*; Necessário para criação e manipulação da tabela onde são mostrados os dados do banco de dados.

import java.util.List; É utilizado pois há uma *List<tfilmes>* que pega todas as linhas da tabela filmes do banco de dados e escreve na *JTable*.

Alterar.java

A classe alterar é a apenas acessada pelo administrador mas sua lógica é a base para as classes de venda e aluguel de filmes. Ao ser inicializada todos os filmes e suas informações são exibidas na tabela a qual o usuário pode navegar e selecionar qualquer filme que será alterado, podendo alterar qualquer informação do filme exceto o código (ID). As alterações são feitas por meio do comando SQL *update*.

The screenshot shows a Java Swing window titled "Alterar filmes". It contains a table with the following columns: Codigo, Nome, Duração, Gênero, Ano, Sinopse, Diretor, Preço, Aluguel, and Em estoque. The first row is selected, showing details for "Homem Formiga" (Ant-Man).

Codigo	Nome	Duração	Gênero	Ano	Sinopse	Diretor	Preço	Aluguel	Em estoque
1	Homem Fo...	117	Ficção Cien...	2015	Hank Pym, ...	Peyton Reed	30.5	21.2	1
2	Os Vingado...	143	Ação	2012	Nick Fury, d...	Joss Whed...	30.99	19.99	1
3	O Regresso	156	Faroeste / ...	2015	Um explora...	Alejandro G...	35.9	25.9	1
4	Toy Story	81	Aventura / C...	1995	Woody é o ...	John Lasse...	24.99	15.5	1

Below the table, there are four buttons: "Alterar", "Salvar", "Cancelar", and "Sair". To the right of the table, there is a form with labels and input fields for each column of the selected row:

- Código: 1
- Nome: Homem Formiga
- Duração: 117
- Gênero: Ficção Científica
- Ano: 2015
- Sinopse: sua tecnologia de encolhimento
- Diretor: Peyton Reed
- Preço: 30.5
- Preço aluguel: 21.2
- Em estoque: 1

As bibliotecas utilizadas foram:

`import java.awt.event.*;` Para tratar os eventos da tela como cliques em no botão ou na tabela.

`import javax.swing.*;` Necessário para uso dos objetos que compõe a tela como *JButton*, *JTextField* e outros.

`import javax.swing.table.*;` Necessário para criação e manipulação da tabela onde são mostrados os dados do banco de dados.

`import java.util.List;` É utilizado pois há uma *List<tfilmes>* que pega todas as linhas da tabela filmes do banco de dados e escreve na *JTable*.

Excluir.java

Essa classe possui a funcionalidade de excluir do banco de dados o filme desejado. Tem acesso apenas do administrador pois é uma operação muito arriscada ao funcionário. O design da tela é bem semelhante as demais telas e há uma tela de confirmação se realmente deseja excluir o filme.

The application window titled "Excluir filmes" contains a table with the following data:

Código	Nome	Duração	Gênero	Ano	Sinopse	Diretor	Preço	Aluguel	Em estoque
1	Homem Fo...	117	Ficção Cien...	2015	Hank Pym, ...	Peyton Reed	30.5	21.2	1
2	Os Vingado...	143	Ação	2012	Nick Fury, d...	Joss Whed...	30.99	19.99	1
3	O Regresso	156	Faroeste / ...	2015	Um explora...	Alejandro G...	35.9	25.9	1
4	Toy Story	81	Aventura / C...	1995	Woody é o ...	John Lasse...	24.99	15.5	1

The right side of the window contains a form with the following fields:

- Código: 1
- Nome: Homem Formiga
- Duração: 117
- Gênero: Ficção Científica
- Ano: 2015
- Sinopse: Hank Pym, demite-se da S.H.I.E
- Diretor: Peyton Reed
- Preço: 30.5
- Preço aluguel: 21.2
- Em estoque: 1

The bottom screenshot shows a confirmation dialog box titled "Sistema informa:" with the message "Deseja Realmente Apagar?" and buttons "Sim" and "Não".

As bibliotecas utilizadas foram:

`import java.awt.event.*;` Para tratar os eventos da tela como cliques em no botão ou na tabela.

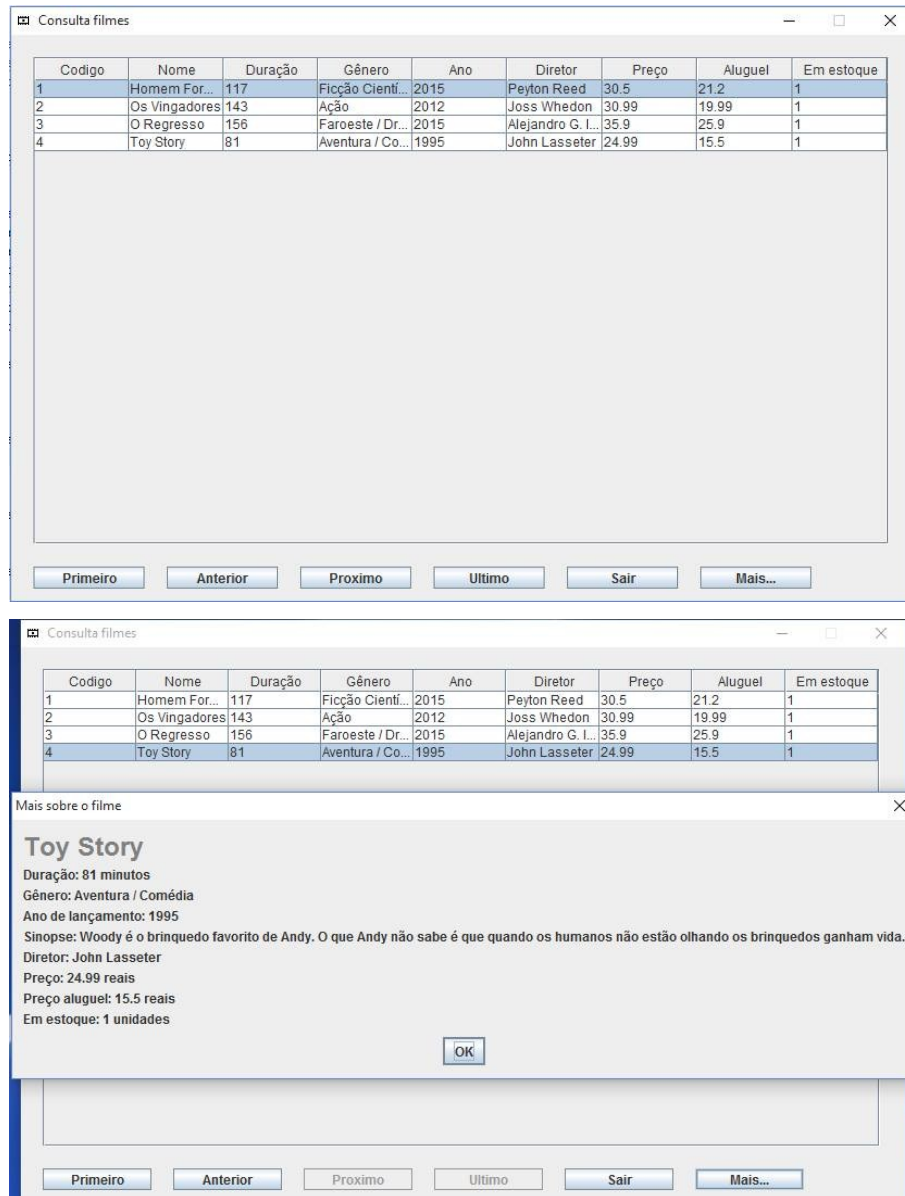
`import javax.swing.*;` Necessário para uso dos objetos que compõe a tela como *JBUTTON*, *JTextField* e outros.

`import javax.swing.table.*;` Necessário para criação e manipulação da tabela onde são mostrados os dados do banco de dados.

`import java.util.List;` É utilizado pois há uma *List<tfilmes>* que pega todas as linhas da tabela filmes do banco de dados e escreve na *JTable*.

Consulta.java

É uma classe que serve apenas para navegação e consulta entre os filmes, não faz operação alguma com os filmes além de exibir.



As bibliotecas utilizadas foram:

`import java.awt.event.*;` Para tratar os eventos da tela como cliques em no botão ou na tabela.

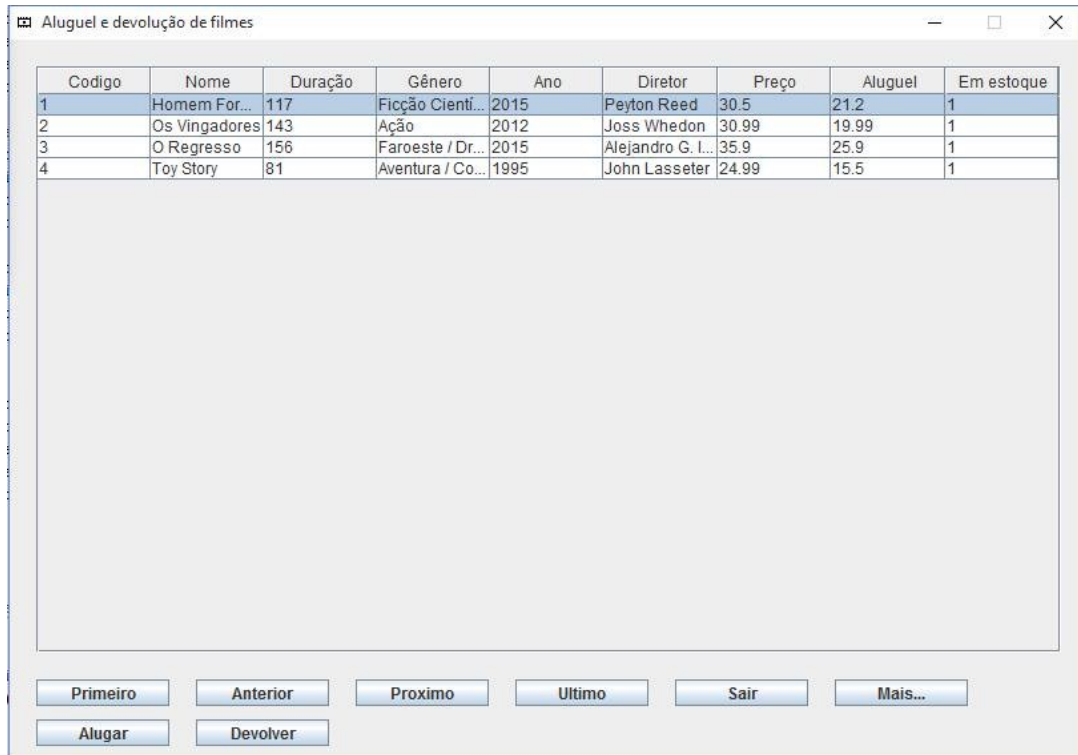
`import javax.swing.*;` Necessário para uso dos objetos que compõe a tela como *JButton*, *TextField* e outros.

`import javax.swing.table.*;` Necessário para criação e manipulação da tabela onde são mostrados os dados do banco de dados.

`import java.util.List;` É utilizado pois há uma *List<tfilmes>* que pega todas as linhas da tabela filmes do banco de dados e escreve na *JTable*.

Locadora.java

Essa classe é destinada a aluguel e devolução de filmes na locadora, basicamente alterar a quantidade de filmes em estoque, quando aluga subtrai um e quando devolve adiciona um; portanto realiza alterações no banco de dados sendo registrado no log do sistema. Essa tela é de acesso a funcionário e administrador, é diferenciado no log quem fez a alteração.



Codigo	Nome	Duração	Gênero	Ano	Diretor	Preço	Aluguel	Em estoque
1	Homem For...	117	Ficção Cientí...	2015	Peyton Reed	30.5	21.2	1
2	Os Vingadores	143	Ação	2012	Joss Whedon	30.99	19.99	1
3	O Regresso	156	Faroeste / Dr...	2015	Alejandro G. L...	35.9	25.9	1
4	Toy Story	81	Aventura / Co...	1995	John Lasseter	24.99	15.5	1

As bibliotecas utilizadas foram:

`import java.awt.Color;` Utilizado para dar cor a uma fonte.

`import java.awt.Font;` Cria fonte que escreve o título do filme na tela que se abre ao clicar no botão "Mais...".

`import java.awt.event.*;` Para tratar os eventos da tela como cliques em no botão ou na tabela.

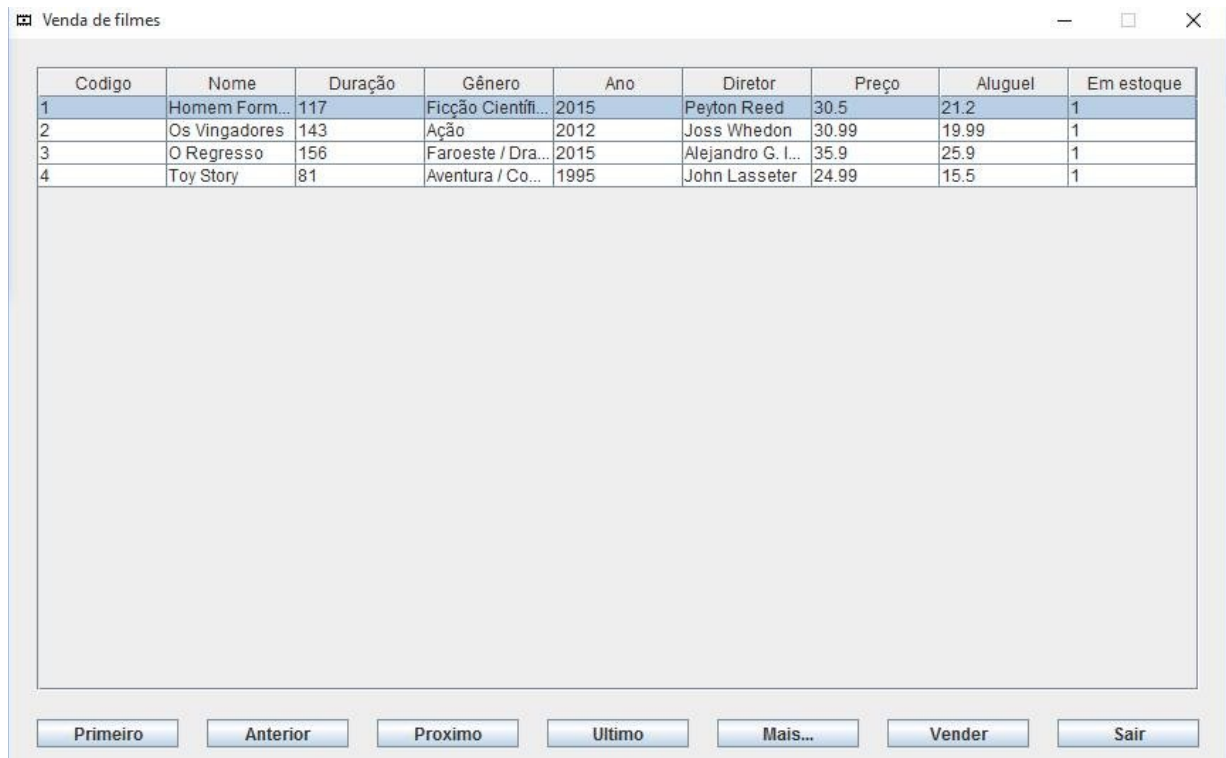
`import javax.swing.*;` Necessário para uso dos objetos que compõe a tela como *JButton*, *JTextField* e outros.

`import javax.swing.table.*;` Necessário para criação e manipulação da tabela onde são mostrados os dados do banco de dados.

`import java.util.List;` É utilizado pois há uma *List<tfilmes>* que pega todas as linhas da tabela filmes do banco de dados e escreve na *JTable*.

Loja.java

Essa classe é destinada apenas a venda de filmes no sistema, basicamente altera a quantidade de filmes em estoque, subtraindo um quando vende um filme; portanto realizando alterações no banco de dados sendo registrado no log do sistema. Essa tela é de acesso a funcionário e administrador, é diferenciado no log quem fez a alteração.



Codigo	Nome	Duração	Gênero	Ano	Diretor	Preço	Aluguel	Em estoque
1	Homem Form...	117	Ficção Cientifi...	2015	Peyton Reed	30.5	21.2	1
2	Os Vingadores	143	Ação	2012	Joss Whedon	30.99	19.99	1
3	O Regresso	156	Faroeste / Dra...	2015	Alejandro G. I...	35.9	25.9	1
4	Toy Story	81	Aventura / Co...	1995	John Lasseter	24.99	15.5	1

As bibliotecas utilizadas foram:

`import java.awt.Color;` Utilizado para dar cor a uma fonte.

`import java.awt.Font;` Cria fonte que escreve o título do filme na tela que se abre ao clicar no botão “Mais...”.

`import java.awt.event.*;` Para tratar os eventos da tela como cliques em no botão ou na tabela.

`import javax.swing.*;` Necessário para uso dos objetos que compõe a tela como *JButton*, *JTextField* e outros.

`import javax.swing.table.*;` Necessário para criação e manipulação da tabela onde são mostrados os dados do banco de dados.

`import java.util.List;` É utilizado pois há uma *List<tfilmes>* que pega todas as linhas da tabela filmes do banco de dados e escreve na *JTable*.