

Vitor Ferreira França 2

## Trabalho 2



Git

## **Sumario**

**-Adições (pagina 3).**

**- Mudanças para obedecer os princips SOLID e a listagem dinamica (pagina 8).**

## -Adições

O material "**Filme**" foi adicionado ao sistema, estendendo a classe abstrata "**Conteúdo**", que reúne os atributos comuns a todos os materiais, como livros e revistas. Além disso, a classe implementa o método `getAtributos()`, conforme definido por uma interface.

Também foram desenvolvidos um **controller** e uma **view** para facilitar a adição de novos filmes à lista.

## -FilmeView

```
package view;

import controller.FilmeController;
import model.Conteudo;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import model.*;
import javax.swing.*;
import java.util.ArrayList;

public class FilmeView{
    public FilmeView(ArrayList<Conteudo> listaLivro, JFrame janelaMain) {
        // Criando a janela principal
        JFrame jframe = new JFrame("Adicionar Filme");
        jframe.setSize(500, 400);
        jframe.setLocationRelativeTo(null);
        jframe.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        jframe.setLayout(new BorderLayout(10, 10));

        // Painel para os campos de entrada
        JPanel painelCampos = new JPanel(new GridLayout(3, 2, 10, 10));
        painelCampos.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

        JLabel tituloLabel = new JLabel("Título:");
        JTextField tituloField = new JTextField();

        JLabel autorLabel = new JLabel("Autor:");
        JTextField autorField = new JTextField();
    }
}
```

```
JLabel anoLabel = new JLabel("Ano:");
JTextField anoField = new JTextField();

JLabel diretorLabel = new JLabel("Diretor:");
JTextField diretorField = new JTextField();

// Adicionando componentes ao painel
painelCampos.add(tituloLabel);
painelCampos.add(tituloField);
painelCampos.add(autorLabel);
painelCampos.add(autorField);
painelCampos.add(anoLabel);
painelCampos.add(anoField);
painelCampos.add(diretorLabel);
painelCampos.add(diretorField);

// Criando os botões
JButton adicionarBtn = new JButton("Incluir");
JButton bibliotecaBtn = new JButton("Biblioteca");
JButton abrirRevistaBtn = new JButton("Revista");

// Ajustando a fonte dos botões
Font botaoFonte = new Font("Arial", Font.BOLD, 14);
adicionarBtn.setFont(botaoFonte);
bibliotecaBtn.setFont(botaoFonte);
abrirRevistaBtn.setFont(botaoFonte);

// Definição do controlador
FilmeController controller = new FilmeController();

// Ação do botão Adicionar
adicionarBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String titulo = tituloField.getText();
            String autor = autorField.getText();
            int ano = Integer.parseInt(anoField.getText());
            String diretor = diretorField.getText();

            // Limpar os campos após adicionar
            tituloField.setText("");
```

```

        autorField.setText("");
        anoField.setText("");
        diretorField.setText("");

        controller.AdicionaFilme(titulo, autor, ano,diretor,listaLivro);

        JOptionPane.showMessageDialog(jframe, "Livro adicionado com sucesso!",
        "Sucesso", JOptionPane.INFORMATION_MESSAGE);

        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(jframe, "Erro: Ano deve ser um número.",
        "Erro", JOptionPane.ERROR_MESSAGE);
        }
    }
});

// Ação do botão Biblioteca
bibliotecaBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
        jframe.dispose();
        janelaMain.setVisible(true);
    }
});

// Ação do botão Abrir Revista
abrirRevistaBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
        jframe.dispose();
        new RevistaView(listaLivro, janelaMain);
    }
});

// Pannel de botões
JPanel painelBotoes = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));
painelBotoes.add(abrirRevistaBtn);
painelBotoes.add(bibliotecaBtn);
painelBotoes.add(adicionarBtn);

// Adicionando os componentes à janela
jframe.add(painelCampos, BorderLayout.CENTER);

```

```

jframe.add(painelBotoes, BorderLayout.SOUTH);

jframe.setVisible(true);
}

}

```

## -Modelo de Filme

```

package model;

public class Filme extends Conteudo{
    String diretor;
    public Filme(String titulo,String autor,int ano,String diretor){
        super(titulo, autor, ano);
        this.diretor=diretor;
    }

    public String getDiretor(){
        return this.diretor;
    }
    @Override
    public String getAtributos() {
        return "Livro " +
            "titulo='" + this.getTitulo() + "\" +
            ", autor='" + this.getAutor() + "\" +
            ", ano=" + this.getAno() +
            ", diretor=" +this.getDiretor()+
            '\n';
    }
}

```

## Controller de Filme

```

package controller;

import model.Conteudo;
import model.Filme;

```

```
import model.Revista;

import java.util.ArrayList;

public class FilmeController {
    public void AdicionaFilme(String titulo,String autor,int ano,String diretor
,ArrayList<Conteudo> Lista) {
        Filme filme = new Filme(titulo, autor, ano,diretor);
        Lista.add(filme);
    }
}
```

## -Mudanças para obedecer os princípios SOLID e a listagem dinamica

Houve a criação e adição da classe abstrata "**Conteúdo**" e de uma interface que declara métodos em comum. Dessa forma, as classes **Livro**, **Filme** e **Revista** estendem **Conteúdo**, garantindo uma estrutura unificada para os materiais. Assim sendo possível a aplicação de listagem dinamica.

Classe “conteudo”

```
package model;

public abstract class Conteudo implements Materiais {
    private String titulo;
    private String autor;
    private int ano;
    public String modelo="default";

    public Conteudo(String titulo, String autor,int ano){
        this.titulo=titulo;
        this.autor=autor;
        this.ano=ano;
    }

    public String getTitulo(){
        return this.titulo;
    }
}
```

```

}

public void setTitulo(String titulo){
    this.titulo = titulo;
}

public String getAutor(){
    return this.autor;
}

public void setAutor(String autor){
    this.autor = autor;
}

public int getAno(){
    return this.ano;
}

public void setAno(int ano){
    this.ano = ano;
}

public abstract String getAtributos();
}

```

-Interface

```

package model;

public interface Materiais {
    String getAtributos();
}

```

-Ao ser impresso

```

JButton btnAtualizar = new JButton("Atualizar Listagem");
btnAtualizar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
        txtLista.setText(""); // Limpa antes de listar
        for (Conteudo material : listaMateriais) {

```



```
txtLista.append(material.getAtributos());  
  
    }  
}  
});
```