

# Diário Cultural: Sistema de Acompanhamento de Leitura e Audiovisual

Enzo Cauã da S. Barbosa

<sup>1</sup>Universidade Estadual de Feira de Santana (UEFS)

Av. Transnordestina, s/n, Novo Horizonte

Feira de Santana – BA, Brasil – 44036-900

caua7uefs@gmail.com

**Abstract.** *This third phase of the Diário Cultural application focused on implementing a graphical user interface using JavaFX. The transition from a text-based interface to a visual one aimed to improve usability and accessibility for the end user. Object-oriented programming concepts and the MVC architectural pattern were applied, along with tools such as SceneBuilder, CSS for visual styling, and Maven for dependency management. The interface includes features such as media registration, rating, listing, and searching. Manual testing confirmed the integrity of the interactions and their integration with the existing data persistence layer. As a result, the application became more user-friendly and functionally complete, ready for future enhancements.*

**Resumo.** *Esta terceira fase do desenvolvimento da aplicação Diário Cultural teve como foco a implementação da interface gráfica utilizando JavaFX. A substituição da interface textual por uma interface visual teve como objetivo tornar a navegação mais intuitiva e acessível ao usuário final. Para isso, foram utilizados conceitos de orientação a objetos, o padrão arquitetural MVC, além de ferramentas como SceneBuilder, CSS para estilização visual e Maven para gerenciamento de dependências. A interface gráfica contempla funcionalidades como cadastro, avaliação, listagem e busca de mídias. Foram realizados testes manuais que confirmaram a integridade das interações e a integração com o modelo de dados já persistente. O resultado foi uma aplicação mais amigável e funcional, pronta para evoluções futuras.*

## 1. Introdução

Na terceira fase do desenvolvimento do *Diário Cultural*, foi implementada a interface gráfica do sistema, com o objetivo de substituir os menus baseados em terminal por uma experiência mais visual, intuitiva e acessível ao usuário final. Essa evolução atende às limitações identificadas nas fases anteriores, especialmente no que se refere à usabilidade e navegabilidade da aplicação.

Com base nos princípios da programação orientada a objetos (POO) e mantendo a arquitetura previamente estabelecida pelo padrão Model-View-Controller (MVC), a interface gráfica foi construída utilizando a biblioteca JavaFX, amplamente reconhecida por sua integração com a linguagem Java e sua capacidade de construção de interfaces ricas.

Além disso, esta fase passou a utilizar ferramentas específicas para o desenvolvimento frontend, como o SceneBuilder, responsável pela modelagem visual das telas

através de arquivos FXML, e o CSS, utilizado para a estilização dos componentes visuais. O gerenciamento de dependências do projeto se deu por meio do Apache Maven, garantindo uma organização eficiente das bibliotecas e módulos envolvidos.

Este relatório detalha a construção da interface gráfica, abordando desde os conceitos técnicos utilizados até a descrição da usabilidade do sistema, os testes realizados, os desafios enfrentados e as perspectivas de melhorias futuras.

## **2. Fundamentação Teórica**

A implementação da interface gráfica na aplicação *Diário Cultural* fundamenta-se em conceitos de programação orientada a objetos, padrões de projeto e tecnologias específicas voltadas ao desenvolvimento de interfaces ricas em Java.

### **2.1. JavaFX e FXML**

O JavaFX é uma biblioteca gráfica moderna da linguagem Java, voltada para o desenvolvimento de interfaces gráficas ricas (Rich Internet Applications). Sua estrutura modular e sua compatibilidade com o padrão MVC tornam-no uma escolha apropriada para aplicações que exigem separação de responsabilidades entre interface, lógica e dados [Oracle 2025].

Uma de suas principais vantagens é o suporte ao formato FXML, um arquivo XML que permite descrever a estrutura visual da interface de forma declarativa. O uso de FXML facilita a separação entre o código de apresentação e a lógica de controle, permitindo que desenvolvedores e designers trabalhem de maneira mais independente.

### **2.2. Padrão MVC aplicado à Interface**

A arquitetura do projeto segue o padrão Model-View-Controller (MVC), que já havia sido adotado na fase anterior. Na interface gráfica, esse padrão é reforçado:

- **Model:** representa os dados persistentes do sistema, como mídias, avaliações e acervos.
- **View:** implementada com arquivos FXML e estilizada com CSS, é responsável pela apresentação visual ao usuário.
- **Controller:** classes Java associadas a cada tela (FXML) que gerenciam eventos, ações de botões e integração com o modelo.

Essa estrutura promove manutenibilidade, modularização e facilita a realização de testes isolados por camada, conforme descrito por Gamma et al. [Gamma et al. 1995].

### **2.3. SceneBuilder e CSS**

Para facilitar a criação das interfaces visuais, utilizou-se o SceneBuilder, ferramenta que permite montar graficamente os arquivos FXML. Essa abordagem reduz erros manuais na criação da interface e agiliza o processo de prototipação visual.

A estilização dos componentes foi realizada com CSS, permitindo personalização de cores, fontes, espaçamentos e estilos dos elementos gráficos. O uso de CSS contribui para a consistência visual da aplicação e facilita a manutenção da identidade visual do sistema.

## 2.4. Maven como Gerenciador de Dependências

O projeto continuou utilizando o Apache Maven como gerenciador de dependências e estruturação do projeto. O Maven permite configurar todas as bibliotecas (como JavaFX) no arquivo `pom.xml`, garantindo portabilidade e reprodutibilidade do ambiente de desenvolvimento em diferentes máquinas [Foundation 2025].

Esses fundamentos teóricos e tecnológicos compõem a base para o desenvolvimento eficiente da interface gráfica do sistema, assegurando sua integração com as camadas já existentes e promovendo uma experiência de usuário mais rica.

## 3. Metodologia

A metodologia adotada para a implementação da interface gráfica da aplicação *Diário Cultural* seguiu uma abordagem incremental e iterativa, estruturada com base nos princípios da engenharia de software e nas diretrizes do padrão arquitetural MVC.

### 3.1. Organização do Projeto

O projeto foi estruturado em módulos conforme os princípios do padrão MVC. As camadas foram organizadas da seguinte forma:

- **Modelo (Model):** reutilizou as classes desenvolvidas nas fases anteriores, responsáveis por representar livros, filmes, séries, avaliações, temporadas e o acervo.
- **Visualização (View):** composta por arquivos FXML desenvolvidos no SceneBuilder, representando as telas de cadastro, avaliação, busca e listagem de mídias. Esses arquivos foram estilizados com CSS para melhorar a estética da interface.
- **Controle (Controller):** formado por classes Java que manipulam eventos da interface (cliques, entradas de texto, seleção de filtros), acessam os dados do modelo e atualizam a visualização de acordo com as ações do usuário.

### 3.2. Construção da Interface

A construção da interface foi realizada com o apoio da ferramenta SceneBuilder, o que possibilitou a criação visual das telas de forma mais ágil e organizada. Cada componente da interface (botões, caixas de texto, rótulos, áreas de avaliação) foi posicionado de forma intuitiva, priorizando a usabilidade.

O controle da interface foi realizado por meio da anotação `@FXML`, que permite vincular elementos visuais a métodos manipuladores nos arquivos de controle Java. Isso garantiu uma integração fluida entre a camada de visualização (FXML) e a lógica de controle.

### 3.3. Estilização com CSS

A personalização visual das telas foi feita com o uso de arquivos `.css`, permitindo configurar cores, espaçamentos, fontes, bordas e efeitos visuais de forma centralizada. Essa separação da lógica de estilo favorece a manutenção e promove uma interface mais moderna e agradável.

### 3.4. Gerenciamento com Maven

O gerenciamento do projeto foi mantido com o uso do Apache Maven, o que permitiu definir e controlar as dependências necessárias, como o JavaFX SDK e bibliotecas auxiliares, através do arquivo `pom.xml`. Isso garantiu reprodutibilidade e agilidade na configuração do ambiente de desenvolvimento.

### 3.5. Iterações e Validações

A implementação foi conduzida em ciclos curtos de desenvolvimento, com testes manuais intermediários a cada nova funcionalidade incorporada. Esse processo iterativo possibilitou identificar falhas de usabilidade, ajustes visuais e incoerências de navegação ainda durante o desenvolvimento, promovendo melhorias contínuas na interface.

## 4. Resultados e Discussões

A implementação da interface gráfica na aplicação *Diário Cultural* resultou em uma experiência significativamente mais amigável e funcional para o usuário final. A seguir, são apresentados os principais aspectos observados após a finalização da interface, destacando funcionalidades implementadas, usabilidade, testes realizados e limitações encontradas.

### 4.1. Funcionalidades Implementadas

A nova interface gráfica abrangeu as principais funcionalidades do sistema, anteriormente disponíveis apenas por meio de menus em terminal. Foram implementadas as seguintes telas:

- Tela Inicial: menu principal com botões para acesso direto às funcionalidades centrais.
- Cadastro de Mídias: formulários para inserção de dados de livros, filmes e séries.
- Avaliação: sistema de avaliação com seleção de estrelas (1 a 5), comentários e marcação como lido/assistido.
- Busca e Filtros: campos para pesquisa por título, autor, gênero e ano, com listagem dinâmica dos resultados.
- Listagem de Acervo: exibição ordenada de mídias cadastradas, com filtros por tipo e avaliação.

Todas as telas foram construídas em FXML e estilizadas com CSS, proporcionando uma aparência consistente e organizada. Os botões, caixas de texto, campos de seleção e áreas de comentários foram dispostos de forma a reduzir o número de cliques e facilitar a navegação.

### 4.2. Usabilidade e Navegação

Durante os testes manuais, observou-se que a interface proporcionou uma navegação mais intuitiva em comparação à versão textual. A utilização de elementos gráficos como botões, abas, estrelas de avaliação e caixas de seleção contribuiu para uma interação mais natural com o sistema, mesmo para usuários sem conhecimentos técnicos.

Mensagens de feedback foram implementadas para informar ao usuário o sucesso ou falha de cada ação, como cadastro concluído, campos obrigatórios não preenchidos ou erro na leitura de arquivos. Esses alertas contribuíram para a prevenção de erros e para a fluidez na utilização do sistema.

### 4.3. Testes Realizados

Foram realizados testes manuais com foco na verificação dos seguintes aspectos:

- Integridade da comunicação entre os controladores e a camada de modelo.
- Validação de entradas nos formulários (ex: notas válidas, campos obrigatórios).
- Comportamento esperado dos botões e elementos gráficos em diferentes fluxos de uso.
- Verificação de mensagens de erro e sucesso.
- Compatibilidade entre diferentes telas e navegação entre funcionalidades.

Os testes confirmaram o correto funcionamento da maioria das interações. Falhas pontuais foram identificadas e corrigidas durante o processo, como botões que não estavam vinculados corretamente aos métodos no controlador, formatação incorreta de datas e erros visuais em resoluções menores.

### 4.4. Limitações Identificadas

Apesar dos avanços obtidos, algumas limitações permanecem nesta fase:

- Validação de entradas: embora tenham sido incluídas validações básicas, ainda há margem para melhoria na verificação de formatos e preenchimentos obrigatórios.
- Feedback visual avançado: algumas ações carecem de feedback mais expressivo, como ícones animados ou transições suaves.
- Adição de funcionalidades complementares: funcionalidades como remoção de mídias, exportação do acervo e personalização do perfil do usuário ainda não foram implementadas.

Esses pontos representam oportunidades para evolução nas próximas fases do projeto.

## 5. Conclusão

A terceira fase do desenvolvimento da aplicação *Diário Cultural* representou um avanço significativo na experiência do usuário, ao introduzir uma interface gráfica moderna e funcional construída com JavaFX. A transição da interação via terminal para uma interface visual proporcionou maior acessibilidade, usabilidade e atratividade ao sistema.

A aplicação dos conceitos de programação orientada a objetos e do padrão arquitetural Model-View-Controller (MVC) garantiu a organização e escalabilidade do sistema, facilitando a integração entre as camadas de controle, visualização e persistência de dados. O uso do SceneBuilder e do CSS permitiu a construção de telas visualmente agradáveis e bem estruturadas, enquanto o Maven garantiu a gestão eficiente das dependências e do ambiente de desenvolvimento.

Dentre os objetivos estabelecidos para esta fase, destacam-se como cumpridos:

- Implementação das principais funcionalidades do sistema na interface gráfica;
- Navegação intuitiva com poucos cliques;
- Estilização da interface com CSS;
- Integração funcional com o backend e a persistência de dados;
- Realização de testes manuais e ajustes iterativos.

Como pendências e oportunidades de melhoria, permanecem:

- Inclusão de funcionalidades complementares, como exclusão de mídias;
- Validações mais robustas nos formulários;
- Adição de recursos de acessibilidade e responsividade;
- Melhorias nos feedbacks visuais (ex: ícones, animações, transições).

Para etapas posteriores, sugerem-se como trabalhos futuros:

- Implementar suporte a múltiplos usuários e autenticação;
- Exportar o acervo em formatos como PDF e CSV;
- Integrar APIs externas para preenchimento automático de dados culturais;
- Adaptar a interface para dispositivos móveis e diferentes resoluções.

A implementação da interface gráfica cumpre um dos marcos mais importantes do projeto, tornando o *Diário Cultural* mais próximo de um sistema completo e pronto para uso prático por usuários finais.

## References

Foundation, A. (2025). Maven project. <https://maven.apache.org>. Acesso em: jul. 2025.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Oracle (2025). Introduction to javafx. <https://openjfx.io>. Acesso em: jul. 2025.