

T200-01 Proj Arquitetura de Sistemas

Cinemax

Leone de Araújo Fernandes – 2315740

Lucas Ferreira Santos Gomes - 2312638

Sistema de Cadastro de Filmes do Cinemax

Propósito do Sistema:

O sistema Cinemax foi desenvolvido com o objetivo de simplificar o gerenciamento de filmes e combos de alimentação em um cinema. Ele permite a realização de operações de cadastro, consulta, atualização e exclusão de registros, oferecendo uma interface intuitiva e funcional para usuários administrativos e clientes do cinema.

Principais Funcionalidades:

- **Cadastro de Filmes e Combos:** Permite aos administradores cadastrar novos filmes e combos informando detalhes como nome, gênero, duração, descrição e preço.
- **Consulta de Filmes e Combos:** Os usuários podem visualizar todos os filmes e combos cadastrados no sistema, obtendo informações detalhadas sobre cada item.
- **Atualização de Filmes e Combos:** Permite a atualização dos dados de filmes e combos existentes, como nome, gênero, duração, descrição e preço.
- **Exclusão de Filmes e Combos:** Oferece a funcionalidade de excluir filmes e combos da base de dados quando necessário.

Principais Usuários:

- **Administradores do Cinema:** São os principais usuários do sistema, responsáveis por gerenciar o catálogo de filmes e combos, realizando operações como cadastro, atualização e exclusão.
- **Clientes do Cinema:** São os usuários finais que desejam consultar os filmes e combos disponíveis antes de adquirir ingressos. Eles utilizam o sistema para visualizar as opções disponíveis e obter informações sobre os itens em cartaz.

Código Utilizado:

Classe Filme: Esta classe representa um filme no sistema. Ela possui atributos como nome, gênero e duração, além de métodos para acessar e modificar esses

atributos. A sobrescrita do método `toString()` permite a obtenção de uma representação textual do objeto filme, facilitando a exibição de informações.

Classe CadastroFilme: É uma classe abstrata que define as operações CRUD (Create, Read, Update, Delete) para os filmes. Ela possui métodos abstratos como `create()`, `readAll()`, `update()` e `delete()`. As subclasses concretas dessa classe (como `CadastroFilmeIndividual` e `CadastroFilmeDupla`) são responsáveis por implementar esses métodos de acordo com suas funcionalidades específicas.

Classe CadastroFactory: Implementa o padrão Factory Method, permitindo a criação de instâncias de `CadastroCombo` com base no tipo especificado (`Individual`, `ComboDupla`). O método `criarCadastro()` recebe o tipo de cadastro desejado e retorna uma instância correspondente da classe `CadastroCombo`.

Classe ComboDupla: Herda da classe abstrata `CadastroCombo` para implementar as funcionalidades específicas relacionadas aos combos duplos de alimentação. Possui métodos para realizar operações CRUD (Create, Read, Update, Delete) sobre os combos, como `create()`, `readAll()`, `update()` e `delete()`.

Classe Combo: Representa um item de combo que pode ser oferecido no cinema. Possui atributos como nome, descrição e preço, além de métodos para acessar e modificar esses atributos. A sobrescrita do método `toString()` permite obter uma representação textual do objeto combo.

Classe Individual: É uma subclasse de `CadastroCombo`, responsável por implementar as operações CRUD para o cadastro de combos individuais de alimentação. Implementa os métodos abstratos `create()`, `readAll()`, `update()` e `delete()` para manipulação dos combos individuais.

Classe CadastroCombo: É uma classe abstrata que define as operações CRUD para os combos de alimentação. As subclasses concretas dessa classe (como `ComboDupla` e `Individual`) são responsáveis por implementar esses métodos de acordo com suas funcionalidades específicas.

Classe ComboController: Atua como controladora das interações relacionadas aos combos de alimentação. Recebe uma lista de combos no construtor e possui métodos para cadastrar novos combos (`cadastrarCombo()`), listar todos os combos (`listarCombos()`), atualizar informações de combos (`atualizarCombo()`) e excluir combos (`excluirCombo()`).

Classe Main: É a classe principal do sistema, responsável por gerenciar a interação com o usuário por meio de um menu interativo. Utiliza objetos `Scanner` para receber as entradas do usuário e possui listas de filmes e combos para armazenar e manipular os dados.

O método `main()` contém o loop principal que exibe o menu de opções para o usuário e permite que ele realize diversas operações relacionadas aos filmes e combos.

As opções do menu correspondem a diferentes ações, como cadastrar filmes, cadastrar combos, listar filmes, listar combos, atualizar informações, excluir registros, entre outras.

Utiliza outros objetos e classes do sistema, como `ComboController`, `CadastroCombo`, `CadastroFilme`, `Filme`, `Combo`, `CadastroFactory`, `Individual` e `ComboDupla`, para executar as operações solicitadas pelo usuário

Padrões GoF (Gang of Four):

Abstract Factory: Implementado na classe `CadastroFactory`, permitindo a criação de instâncias de `CadastroCombo` com base no tipo especificado (`Individual`, `ComboDupla`). **Factory Method:** Presente na estrutura de criação de combos na classe `CadastroCombo`, onde cada subclasse (`Individual`, `ComboDupla`) implementa o método `create` para criar instâncias de `Combo`.

Padrões GRASP (General Responsibility Assignment Software Patterns):

Information Expert: A classe `CadastroCombo` é especialista nas operações relacionadas aos combos de alimentação, garantindo coesão e baixo acoplamento. **Controller:** A classe `ComboController` atua como controladora das interações relacionadas aos combos, mantendo a separação de responsabilidades e facilitando a manutenção do código.

Conclusão

O Sistema de Cadastro de Filmes e Combos do Cinemax foi desenvolvido com o intuito de oferecer uma ferramenta eficiente e intuitiva para o gerenciamento e consulta de informações relacionadas aos filmes em cartaz e combos de alimentação disponíveis. A adoção de padrões de projeto GoF e GRASP proporcionou uma estrutura sólida e flexível, permitindo a expansão e manutenção do sistema de forma organizada e eficaz.