

Aluno: Samuel Toledo

Relatório sobre os Padrões de Projeto MVC e DAO

Introdução

Os padrões de projeto de software são soluções generalizadas para problemas recorrentes no desenvolvimento de sistemas. Eles ajudam a melhorar a estrutura, manutenção e escalabilidade do código. Este relatório abordará dois padrões amplamente utilizados: MVC (Model-View-Controller) e DAO (Data Access Object). Ambos são importantes para a separação de responsabilidades, o que contribui para um design mais modular e robusto.

1. Padrão MVC (Model-View-Controller)

Funcionalidade

O padrão MVC organiza uma aplicação em três componentes principais:

Model (Modelo): Representa os dados da aplicação e as regras de negócios. O Model é responsável por gerenciar o estado da aplicação, armazenando as informações que podem ser manipuladas e acessadas. Ele também contém a lógica de validação dos dados.

View (Visão): É responsável por exibir a interface para o usuário. A View se conecta ao Model para obter as informações a serem apresentadas, mas ela não deve conter lógica de negócios. Sua função é simplesmente renderizar os dados ao usuário, por exemplo, exibindo um formulário ou uma página de resultado.

Controller (Controlador): Age como intermediário entre o Model e a View. Ele recebe as ações do usuário (como cliques ou entradas de formulário), processa essas ações, e, quando necessário, interage com o Model para alterar o estado dos dados. Depois, o Controller atualiza a View para refletir as mudanças.

Usabilidade

O padrão MVC é amplamente utilizado em aplicações web e aplicativos desktop. A separação clara de responsabilidades facilita a manutenção e a evolução do sistema. Algumas das principais vantagens incluem:

Facilidade de manutenção: Como a lógica de negócios, a apresentação e a interação do usuário estão separadas, é possível fazer alterações em uma camada sem afetar as outras.

Reutilização de código: O Model pode ser reutilizado em diferentes partes da aplicação ou em diferentes plataformas, sem a necessidade de modificar a lógica de apresentação ou controle.

Testabilidade: Cada componente pode ser testado individualmente. Por exemplo, a lógica de negócios pode ser testada sem envolver a interface gráfica, o que torna os testes mais rápidos e precisos.

MVC é a base de frameworks populares como Spring MVC (Java), ASP.NET MVC (C#), Ruby on Rails, e Django (Python). Cada um desses frameworks implementa o padrão MVC de uma maneira ligeiramente diferente, mas todos seguem os princípios fundamentais do padrão.

2. Padrão DAO (Data Access Object)

Funcionalidade

O padrão DAO (Data Access Object) é utilizado para abstrair e encapsular o acesso a dados. Ele fornece uma interface simples e isolada para interagir com o banco de dados ou outras fontes de persistência de dados, como arquivos e APIs. O DAO oculta a complexidade do código de banco de dados, concentrando as operações CRUD (Create, Read, Update, Delete) em uma classe específica.

Os principais componentes do padrão DAO incluem:

Entidade (Model): Representa uma linha ou objeto persistido no banco de dados. Cada entidade geralmente corresponde a uma tabela no banco de dados relacional.

DAO: Contém os métodos necessários para acessar os dados da entidade, como `addUser()`, `getUserById()`, `getAllUsers()`, `updateUser()`, e `deleteUser()`. O DAO é responsável por abrir a conexão com o banco de dados, realizar a consulta, processar os resultados e fechar a conexão.

Usabilidade

O padrão DAO é amplamente utilizado em aplicações empresariais que exigem interações constantes com bancos de dados. Ele é especialmente útil em sistemas que implementam persistência de dados usando frameworks como JDBC ou Hibernate em Java. Algumas das principais vantagens do padrão DAO incluem:

Separação de responsabilidades: O DAO isola a lógica de acesso aos dados da lógica de negócios, promovendo um design modular e limpo.

Reutilização e Manutenção: O DAO pode ser reutilizado por várias partes do sistema sem que seja necessário duplicar o código de acesso ao banco de dados. Isso facilita a

manutenção, pois mudanças na estrutura do banco de dados podem ser feitas em um único local (no DAO).

Facilidade de Teste: O padrão DAO facilita o teste unitário das operações de banco de dados. É possível implementar stubs ou mocks do DAO para simular o comportamento de acesso a dados durante os testes de lógica de negócios.

Um exemplo prático do uso de DAO seria um sistema que gerencia uma biblioteca de livros. O BookDAO seria responsável por todas as operações de acesso ao banco de dados relacionadas aos livros, enquanto a lógica de negócios, como calcular multas por atraso, ficaria em outra camada do sistema.

Comparação entre MVC e DAO

Ambos os padrões ajudam a melhorar a organização e a modularidade do código, mas atuam em diferentes áreas:

MVC está relacionado à estrutura geral da aplicação e à interação entre as camadas de apresentação, controle e dados. Ele é mais focado no fluxo de dados dentro da aplicação, separando a lógica de negócios da interface do usuário.

DAO, por outro lado, se concentra na persistência de dados, fornecendo uma interface limpa para interagir com o banco de dados e permitindo que a lógica de negócios não dependa de como os dados são armazenados ou recuperados.

Embora diferentes, os dois padrões podem ser usados em conjunto. Por exemplo, em um sistema MVC, o Model pode delegar as operações de persistência para um DAO, mantendo a lógica de negócios e a lógica de acesso a dados bem separadas.

Conclusão

Tanto o padrão MVC quanto o padrão DAO são fundamentais para o desenvolvimento de sistemas escaláveis e de fácil manutenção. O MVC facilita a separação de responsabilidades entre interface, controle e dados, enquanto o DAO isola a lógica de acesso a dados, tornando o sistema mais modular e desacoplado. A combinação desses dois padrões em um único projeto pode proporcionar uma arquitetura robusta e flexível, pronta para crescer e se adaptar às necessidades de negócio e tecnológicas.