Aluno: Bruno Batista Ferreira
Matéria: Padrões de Projetos
Curso: TSI Data: 11/09/2024

Relatório

Em Padrões de Projetos, existem dois padrões amplamente utilizados para a organização e separação eficientes para aplicações complexas, esses padrões são o **MVC(Model View Controller)** e o **DAO(Data Access Object)**.

O padrão MVC divide a aplicação em três componentes principais:

- **Modelo**: Responsável pela lógica de negócios e gerenciamento dos dados.
- **Visão**: Encarregada da apresentação das informações para o usuário.
- **Controlador**: Intermediário entre o Modelo e a Visão.

O padrão DAO é focado na camada de acesso a dados, definindo uma interface para operações de banco de dados, separando a lógica de negócios do acesso ao banco de dados.

A seguir, exploraremos as funcionalidades e usabilidades dos padrões MVC e DAO. Como eles são essenciais para o desenvolvimento de software mais organizado, flexível e de fácil manutenção.

MVC

Sendo uma das arquiteturas de software mais amplamente utilizada no desenvolvimento de aplicações, o padrão MVC caracteriza pela separação clara das responsabilidades dentro de um sistema, dividindo em três componentes principais: Modelo(Model), Visão(View) e Controlador(Controller), promovendo uma melhor organização do código, facilitando a evolução e manutenção do código.

Modelo

A Modelo é responsável pela lógica de negócios e pela manipulação dos dados, contendo regras e comportamentos que determinam como os dados vão ser processados e armazenados. Uma das principais responsabilidades do Modelo é o gerenciamento do estado dos dados, incluindo criação, leitura, atualização e exclusão de dados, aplicando as regras de negócios necessárias durante essas operações. A Modelo não se preocupa com a apresentação dos dados para o cliente, ela apenas garante que os dados estejam consistentes e conforme as regras do sistema.

Visão

A Visão é a parte responsável pela apresentação dos dados ao usuário, cuidando da interface com o usuário, exibindo as informações pertinentes de maneira agradável, compreensível e interativa, a View não manipula os dados, apresentando o que recebe da Model. A View não sabe de onde os dados vieram, e não precisa se preocupar com isso, ela apenas traduz os dados recebidos em uma saída, pode ser interfaces gráficas, páginas Web, relatório ou qualquer outra forma de saída visual.

A Visão deve ser desacoplada da Model e da Controller, permitindo uma maior flexibilidade da View em relação a apresentação dos dados, podendo criar várias visões para o mesmo dado, deixando toda a aplicação flexível.

Controlador

A Controller é o componente intermediário entre a Model e a View, responsável por processar a entrada do usuário e como afetará a View e a Controller, decidindo o que deve ser feito e coordenando as interações entre o Modelo e a Visão. A Controller contém a lógica do fluxo da aplicação, decidindo qual parte do sistema deve ser acionada e em qual momento, se um usuário por exemplo faz Login, a Controller irá validar essas informações junto a Model, e com base no resultado, o usuário é redirecionado para a tela inicial do aplicativo (View) ou exibe uma mensagem de erro, ficando toda essa lógica encapsulada na Controller.

Vantagens

Uma das principais vantagens é a separação entre camadas, promovendo uma maior modularidade, permitindo que o código seja mais fácil de manter e testar, modificações no sistema podem ser feitas sem a necessidade de mexer com os dados geridos por exemplo. Além disso, a escalabilidade é um ponto significativo no padrão, como as camadas são independentes, é possível fazer expansões de maneira controlada e gradual. A facilidade de manutenção é outro ponto significativo, como as funções das camadas estão claramente definidas, qualquer bug pode ser encontrado mais rapidamente, melhorando a produtividade da equipe de desenvolvimento.

Usabilidade

O padrão MVC pode ser aplicado a uma grande variedade de aplicações, desde aplicações WEB a desktop e mobile, o MVC fornece uma estrutura completa para organizar código, podendo ser implementado por qualquer linguagem de programação, o que o torna muito acessível, Frameworks conhecidos como Django, Ruby on Rails e ASP.NET, adotam o MVC como base para suas arquiteturas.

Conclusão

O padrão MVC proporciona uma arquitetura, modular, flexível e eficiente, melhorando a organização e manutenção do código em aplicações complexas, por esses motivos o padrão continua sendo fundamental para o desenvolvimento de sistemas robustos com fácil manutenção.

DAO

O padrão DAO(Data Access Object) é uma abordagem comum no desenvolvimento de software para lidar com acesso de forma organizada, sendo amplamente utilizado para separar a lógica de negócios da lógica de dados, tornando o sistema mais modular e fácil de manter. A ideia principal do DAO é fornecer uma interface que abstrai e encapsula as operações de acesso aos dados, assim outras partes do sistema interagem com os dados sem conhecer os detalhes relacionados à gestão de dados.

Funcionalidades

O padrão DAO atua como uma ponte entre o sistema e o banco de dados, oferecendo uma interface uniforme para realizar operações de manipulação de dados(CRUD), independentemente da fonte de dados. Sua principal funcionalidade é simplificar o acesso ao banco e tornar a aplicação independente da estrutura do banco de dados ou tecnologia do mesmo, com isso todas as operações de acesso aos dados são centralizadas em classes específicas, facilitando a adaptação, correção e inclusão de novas funcionalidades.

Vantagens

Uma das principais vantagens para o uso do DAO é a capacidade de abstração e encapsulamento para o acesso aos dados, oferecendo uma interface abstrata para realizar operações de banco de dados, escondendo os detalhes de implementação, isso significa que na utilização do DAO não precisa-se saber como o banco é estruturado, as consultas SQL ou como as consultas são gerenciadas. Além disso, DAO encapsula todas as complexidades associadas ao gerenciamento, transações e conexões de banco de dados, toda responsabilidade de lidar com o banco de dados fica isolada dentro do DAO.

O uso do padrão também facilita a manutenção e a reutilização de código, todas as operações estão centralizadas em uma camada dedicada, ficando muito mais fácil modificar ou melhorar o código sem causar efeitos colaterais no restante da aplicação, se uma nova funcionalidade foi desenvolvida, como validação de dados ou consulta, pode ser integrada diretamente no DAO sem necessidade de alterar a lógica de negócios.

A flexibilidade que o padrão oferece ao sistema abstraindo os detalhes de implementação do banco de dados é uma grande vantagem, a aplicação se torna independente da tecnologia de banco de dados, isso significa que é possível alterar a tecnologia de armazenamento de dados sem precisar reescrever grandes partes de código.

O padrão DAO também contribui para a testabilidade do sistema, em vez de testar diretamente a lógica de negócios com acesso real ao banco de dados, os teste podem ser feitos de maneira a isolar, simulando a camada de acesso ao banco de dados, com uma página DAO falsa por exemplo.

Usabilidade

O padrão DAO pode ser usado em uma variedade de sistemas, mas principalmente quando o acesso ao banco de dados é uma parte crítica do sistema, como sistemas de gerenciamento de estoque, sistemas de reservas, plataformas de e-commerce, sistema bancário e outros.

A fácil usabilidade que permite aos desenvolvedores se baseia na criação da camada única de acesso aos dados, que seja fácil de modificar e reutilizar, resultando em um código limpo e organizado, pois a lógica de acesso aos está centralizada em um único lugar.

Conclusão

o padrão DAO é uma solução simples e poderosa para lidar com o acesso aos dados em um sistema, promovendo abstração, encapsulamento, flexibilidade e reutilização de código, além de facilitar a manutenção e a testabilidade do sistema, tornando o sistema mais robusto e preparado para mudanças tecnológicas no futuro.