

Docente: Igor Justino Rodrigues
Discentes: Mayko Diouzeff Mendes do Amaral
Curso: Bacharelado Em Sistemas de Informação
Período: 8ª período
Disciplina: Padrões de Projeto

O que é MVC e qual seu objetivo?

O MVC é um padrão de arquitetura de software que organiza o código em três componentes principais, Model, View e Controller. O objetivo é separar responsabilidades para melhorar a organização, facilitar a manutenção e permitir o desenvolvimento colaborativo em sistemas.

Explique as funções de Model, View e Controller.

O Model é responsável por gerenciar os dados e a lógica de negócios, ele interage com o banco de dados, realiza cálculos ou validações e fornece os dados necessários para a aplicação. A View é responsável por exibir as informações para o usuário, cuidando da interface gráfica ou da renderização de páginas. O Controller atua como intermediário entre o Model e a View, processando as requisições do usuário, acionando o Model e determinando qual View será exibida.

Cite exemplos de frameworks que utilizam MVC.

Frameworks populares que utilizam MVC incluem Laravel para PHP, Ruby on Rails para Ruby, ASP.NET MVC para C# e Django para Python.

Como o MVC facilita a manutenção do sistema?

O MVC facilita a manutenção porque a separação de responsabilidades permite que cada camada seja modificada ou expandida de forma independente.

Dê exemplos de projetos onde o MVC pode ser usado.

Projetos onde o MVC pode ser usado incluem sistemas de gerenciamento de conteúdo, aplicações de e-commerce, plataformas de ensino a distância e sistemas de reservas.

Quais os desafios de implementar MVC?

Os desafios de implementar MVC incluem o aumento inicial da complexidade do projeto, a necessidade de um bom entendimento da arquitetura e o risco de má implementação, que pode criar dependências indesejadas entre as camadas.

O que ocorre se não houver separação de camadas?

Sem a separação de camadas, o código pode se tornar desorganizado e difícil de manter, com mistura de lógica de negócios, interface e controle em um único lugar, isso dificulta a reutilização, os testes e a colaboração no desenvolvimento.

MVC pode ser usado para APIs? Justifique.

O MVC pode ser usado para APIs, pois essas seguem o mesmo princípio de organização. O Model pode gerenciar os dados, o Controller processa as requisições e a View pode ser substituída pela formatação das respostas, como JSON ou XML.

Explique a comunicação entre Model, View e Controller.

A comunicação entre Model, View e Controller ocorre da seguinte forma: o Controller recebe a requisição do usuário, consulta o Model para obter os dados necessários e, em seguida, repassa esses dados para a View, que os apresenta ao usuário. Qualquer interação do usuário com a interface é processada novamente pelo Controller.

Como o MVC se aplica em aplicações web modernas?

Em aplicações web modernas, o MVC é aplicado como base, mas muitas vezes adaptado. Por exemplo, o front-end pode usar frameworks como React ou Vue.js para cuidar das Views, enquanto o back-end implementa o Model e o Controller em frameworks como Django ou Laravel, integrando-se via APIs RESTful ou GraphQL. O padrão continua sendo fundamental para a organização e manutenção de sistemas complexos.