

AULA 7

Padrões de Projetos

Prof. Igor Justino Rodrigues
18/11/2024



OBJETIVOS DA AULA

- Compreender o que é o padrão arquitetural MVC.
- Entender as funções de cada camada do padrão MVC.
- Visualizar a aplicação prática de MVC com um exemplo funcional em PHP.

An abstract digital cityscape with glowing blue and red cubes, binary code, and light trails on a dark blue background.

Padrão MVC

(Model-View-Controller)

- **Definição:**

- MVC é uma arquitetura que divide um sistema em três camadas: Model, View e Controller.
- Foco na separação de responsabilidades para maior organização e manutenção.

- **Metáfora:**

- Pense no MVC como um restaurante:
 - **Model:** O chef que prepara os pratos.
 - **View:** O garçom que apresenta o prato ao cliente.
 - **Controller:** O gerente que coordena o pedido e a entrega.

Por que usar MVC?

Benefícios:

- Isolamento entre lógica de negócios e interface do usuário.
- Reutilização de código em diferentes interfaces.
- Facilidade na manutenção e adição de novos recursos.
- Organização do código para equipes de desenvolvimento.



Estrutura do MVC

Model (Modelo):

Responsável por gerenciar os dados e as regras de negócio.

View (Visão):

Exibe as informações para o usuário e captura interações.

Controller (Controlador):

Interpreta as ações do usuário e conecta o Model com a View.

Fluxo de Comunicação

Input ➤ Processing ➤ Output

Controller ➤ Model ➤ View

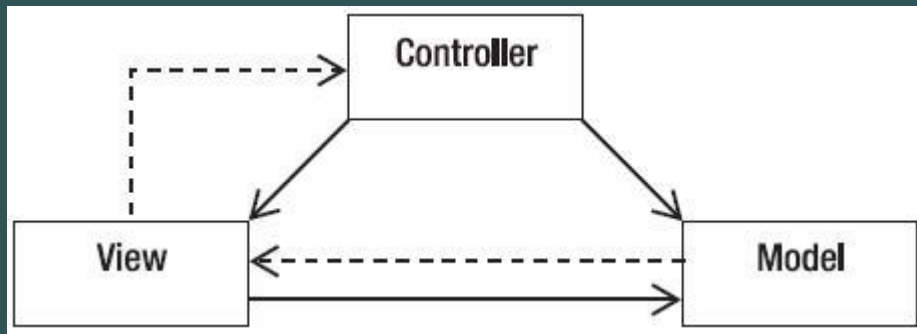
1. Entrada do Usuário: Capturada pela View.

2. Ação do Controller: O Controller interpreta e decide o que fazer.

3. Processamento do Model: O Model realiza a lógica e manipula os dados.

4. Resposta: O resultado volta ao Controller e é exibido pela View.

Estrutura



```
project/
├── controllers/
│   ├── UserController.php
│   └── TaskController.php
├── models/
│   ├── User.php
│   └── Task.php
├── views/
│   ├── users/
│   │   ├── index.php
│   │   ├── create.php
│   │   ├── edit.php
│   │   └── show.php
│   └── tasks/
│       ├── index.php
│       ├── create.php
│       ├── edit.php
│       └── show.php
├── core/
│   ├── Database.php
│   └── Router.php
├── public/
│   └── index.php
├── .env
└── composer.json
```


Principais Vantagens do MVC

1

Separação de Responsabilidades:

Cada camada foca em sua função específica.

2

Facilidade de Manutenção:

Alterações em uma camada não impactam as outras.

3

Escalabilidade:

Permite expandir o sistema sem grandes refatorações.

4

Reuso de Código:

Interface e lógica de negócios podem ser reaproveitadas.

Perguntas para Pesquisa

O que é MVC e qual seu objetivo?

Explique as funções de Model, View e Controller.

Cite exemplos de frameworks que utilizam MVC.

Como o MVC facilita a manutenção do sistema?

Dê exemplos de projetos onde o MVC pode ser usado.

Quais os desafios de implementar MVC?

O que ocorre se não houver separação de camadas?

MVC pode ser usado para APIs? Justifique.

Explique a comunicação entre Model, View e Controller.

Como o MVC se aplica em aplicações web modernas?