

MVC

Modelo-Visão-Controle

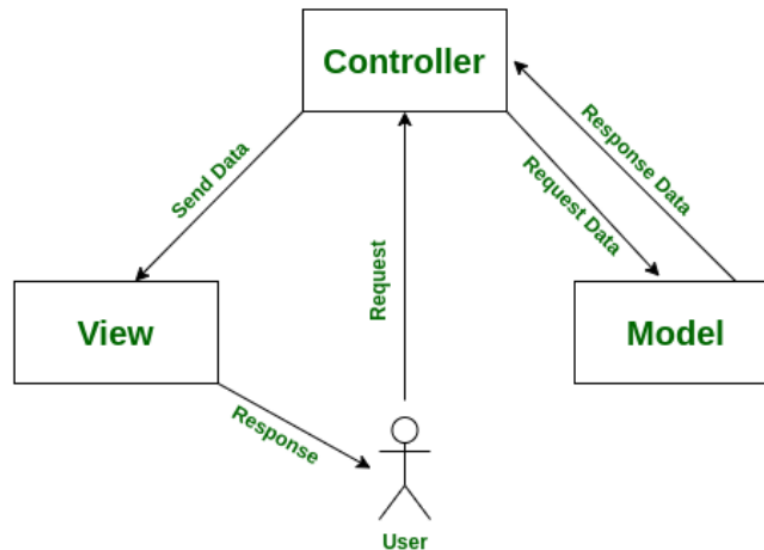
Definição

MVC (Model-View-Controller) é um padrão de design de software comumente usado para implementar interfaces de usuário, dados e lógica de controle. Ele enfatiza a separação entre a lógica de negócios do software e sua exibição. Essa "**separação de preocupações**" (separation of concerns) permite uma melhor divisão de trabalho e facilita a manutenção. Outros padrões de design são baseados no MVC, como:

- **MVVM** (Model-View-ViewModel)
- **MVP** (Model-View-Presenter)
- **MVW** (Model-View-Whatever)

As três partes do MVC

- **Model (Modelo):** Gerencia os dados e a lógica de negócios.
- **View (Visão):** Responsável pelo layout e exibição dos dados.
- **Controller (Controlador):** Direciona os comandos para o modelo e a visão.



Modelo

- A camada de modelo é responsável pela lógica de dados da aplicação e pelo armazenamento e recuperação de dados de repositórios de dados de back-end. A camada de modelo também pode incluir mecanismos para validar dados e realizar outras tarefas relacionadas a dados. Essa camada é responsável por manter todos os aspectos dos dados e garantir sua integridade e acessibilidade.

Visão

- A camada de visualização fornece a interface do usuário (IU) necessária para interagir com a aplicação. Ela inclui os componentes necessários para exibir os dados e permite que os usuários interajam com esses dados. Por exemplo, a camada de visualização pode incluir botões, links, tabelas, listas suspensas ou caixas de texto.

Controle

- A camada de controle contém a lógica da aplicação necessária para facilitar a comunicação entre a aplicação, atuando como uma interface entre as camadas de visualização e de modelo. O controlador às vezes é visto como o cérebro da aplicação, mantendo tudo em movimento e sincronizado.

Exemplo

- Para um exemplo simples, utilizaremos o código contido em **`/scripts/mvc-exemplo`**
- O miniprojeto se trata de uma aplicação de lista simples

Exemplo

model.php

- A classe ItemModel serve como uma representação de um banco de dados, armazenando os dados da lista
- Ela também oferece métodos para a manipulação dessas informações

```
<?php
class ItemModel
{
    public function __construct()
    {
        if (!isset($_SESSION['items'])) {
            $_SESSION['items'] = [];
        }
    }

    public function addItem($item)
    {
        $_SESSION['items'][] = $item;
    }

    public function getItems()
    {
        return $_SESSION['items'];
    }

    public function popItem()
    {
        if ((count($_SESSION['items']) > 0)) {
            array_pop($_SESSION['items']);
        }
    }
}
```


Exemplo

control.php

- A classe *control* interage com o Model, através do *'require_once'*
- Control instancia um objeto Model e através dele realiza as operações

```
<?php
require_once("model.php");
class Control
{
    protected $modelo;
    public function __construct()
    {
        $this->modelo = new ItemModel();
    }
    public function getItems()
    {
        return ($this->modelo)->getItems();
    }
    public function addItem(string $item)
    {
        $this->modelo->addItem($item);
    }
    public function popItem()
    {
        $this->modelo->popItem();
    }
}
```

Exemplo

view.php

- No view, é instanciado um objeto *control*, a visã não entra em contato direto com o modelo
- Atráves dessa instância, realiza as requisições necessárias, como em: *\$items = \$controller->getItems();*

```
1  <?php
2  session_start();
3  require_once("control.php");
4
5  $controller = new Control();
6  $items = $controller->getItems();
7
```

```
if (isset($_POST['mensagem']) && trim($_POST['mensagem']) != '') {
    $msg = $_POST['mensagem'];
    $controller->addItem($msg);

    header("Location: " . $_SERVER['PHP_SELF']); // para que as alteraç
    exit;
}

if (isset($_POST['limpar'])) {
    unset($_SESSION['items']);
    header("Location: " . $_SERVER['PHP_SELF']);
    exit;
}

if (isset($_POST['pop'])) {
    $controller->popItem();
    header("Location: " . $_SERVER['PHP_SELF']);
    exit;
}
```

Referências Bibliográficas

- <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
- <https://www.geeksforgeeks.org/benefit-of-using-mvc/>