

Sistemas Multimédia

TPW-II

Tecnologia e Prática da Web II

15

ASP.NET Core MVC

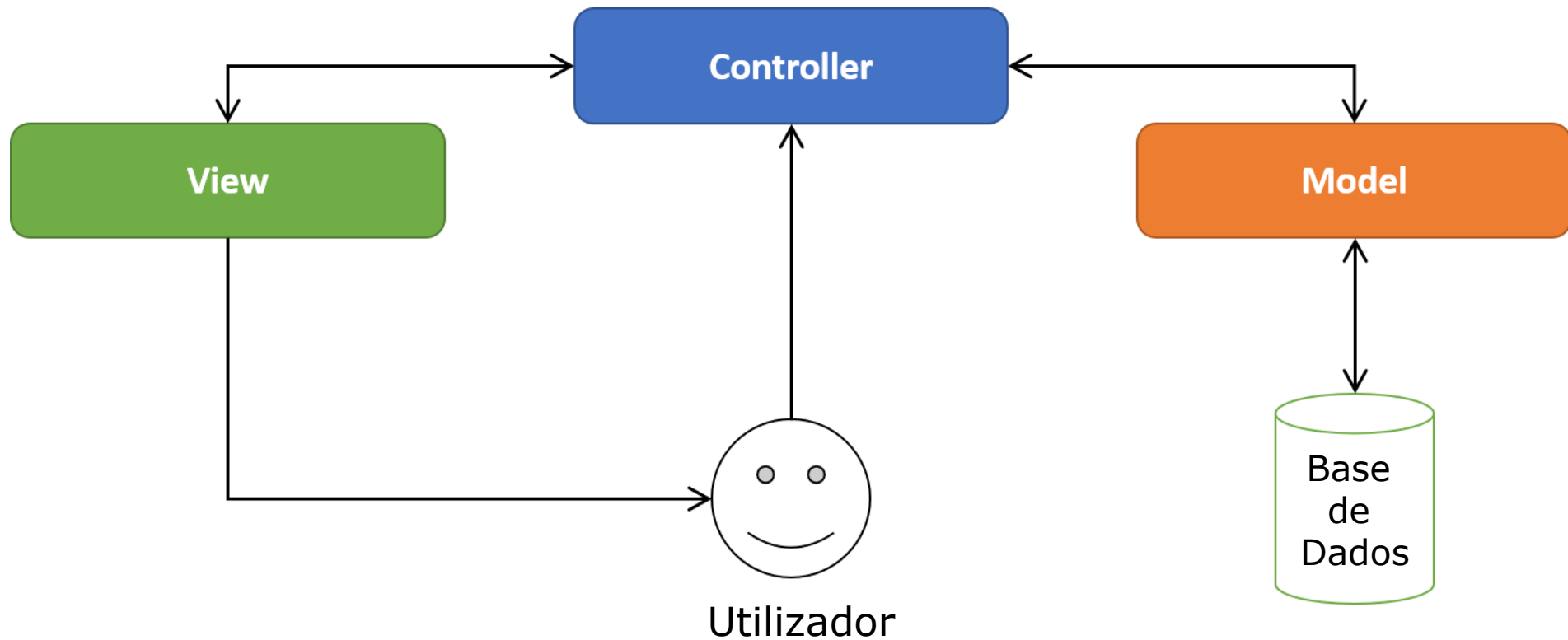
□ ASP.NET core MVC

- Arquitetura MVC
- ASP.NET MVC

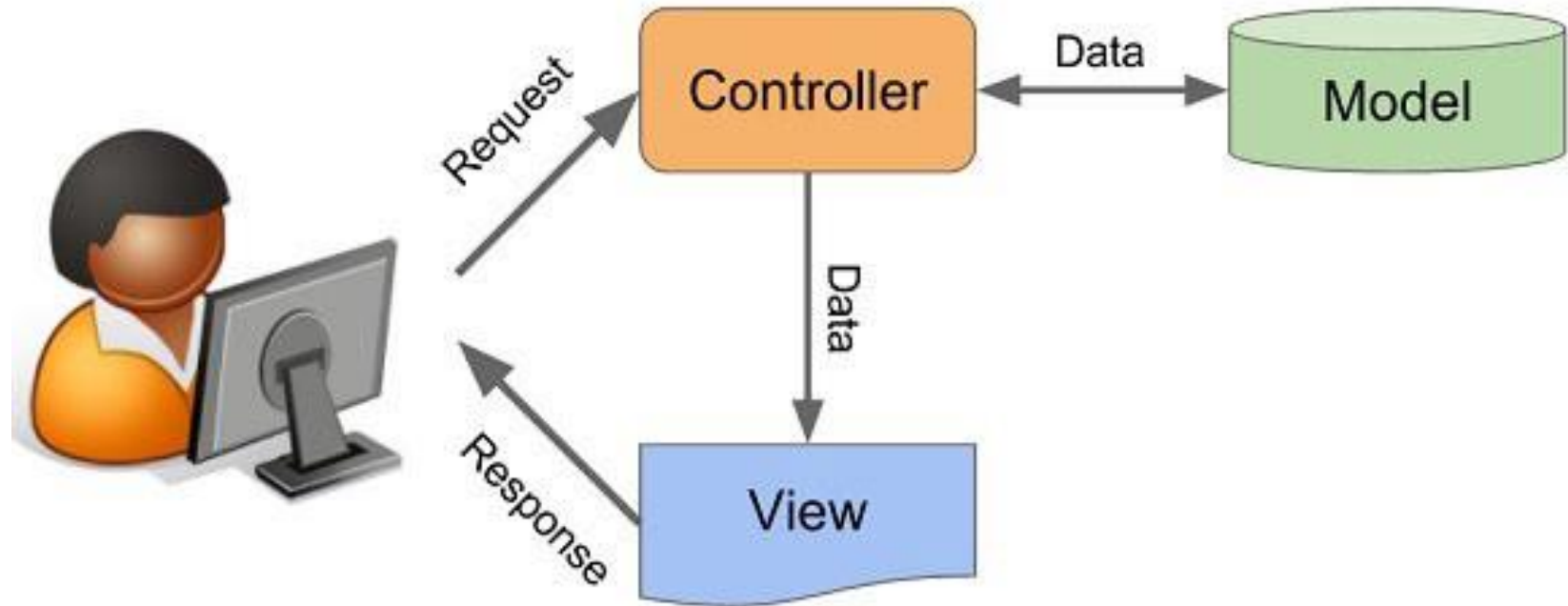
Arquitetura MVC

- ❑ MVC (Model-View-Controller)
 - Padrão de arquitetura de software que separa a apresentação da informação da interação do utilizador.
 - Já existe desde aos anos de 1970
- ❑ Separa a aplicação em três componentes:
 - **Model** (modelo) – consiste nos dados da aplicação, regras de negócios, lógica e funções.
 - **View** (visualização) – pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível ter várias visões dos mesmos dados, como um gráfico de barras ou uma tabela – design da aplicação.
 - **Controller** (controlador) – faz a mediação da entrada (requisição do utilizador, *browser*), convertendo-a em comandos para o model ou view.
- ❑ As ideias centrais por detrás do MVC são a reutilização de código e separação de conceitos.

□ Arquitetura MVC



□ Arquitetura MVC



□ MVC

- É um conceito existente desde 1970.
- Tem um lugar especial no desenvolvimento web.

Aplicação dividida em, pelo menos, 3 peças:

- **Models** – Contém ou representa os dados da aplicação;
- **Views** – Apresentação dos dados e interface;
- **Controllers** – Recebe pedidos, executa operações no Model e retorna Views para o browser.

MVC = Separação de interesses da aplicação

A lógica dos dados fica apenas dentro do Model.

A lógica de apresentação, apenas dentro das Views.

O código que gere pedidos e inputs, apenas nos Controllers.

□ MVC

MODELS

View Models – Representam dados que são passados diretamente para Views;

Domain Models – Contêm dados e toda a lógica de operações para transformar, criar regras, guardar e manipular dados. Por exemplo, a lógica de comunicação com bases de dados.

MODELS

Os models devem conter os dados, a lógica para manipular dados e os mecanismos para expor as operações de manipulação dos dados.

Os models NÃO DEVEM expor detalhes sobre como os dados são obtidos, não conter lógica relacionada com interação (isso fica para o Controller) e não conter lógicas de apresentação (isso fica para as Views)

□ MVC

CONTROLLERS

São o elo de ligação entre os Models e as Views.

Os Controllers DEVEM ter funcionalidades para atualizar o Model de acordo com a interação do utilizador, e mecanismos de retorno de Views.

NÃO DEVEM conter lógica de apresentação de dados (é função das Views) e não ter lógica de gestão dos dados (função do Model)

VIEWS

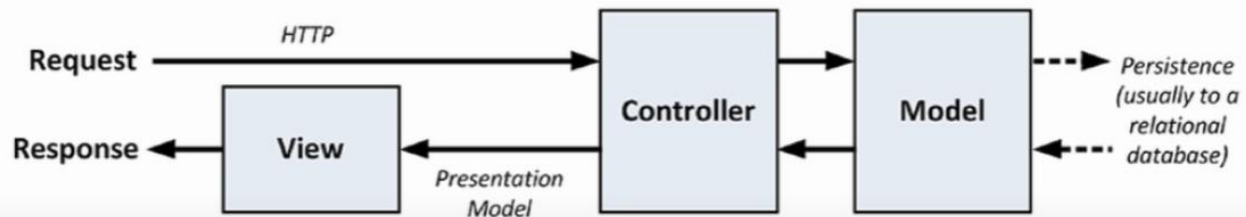
Contêm a lógica necessária para apresentar dados ou para captar inputs do utilizador (por exemplo, formulários)

DEVEM conter lógica e markup para apresentação de dados e do interface da aplicação.

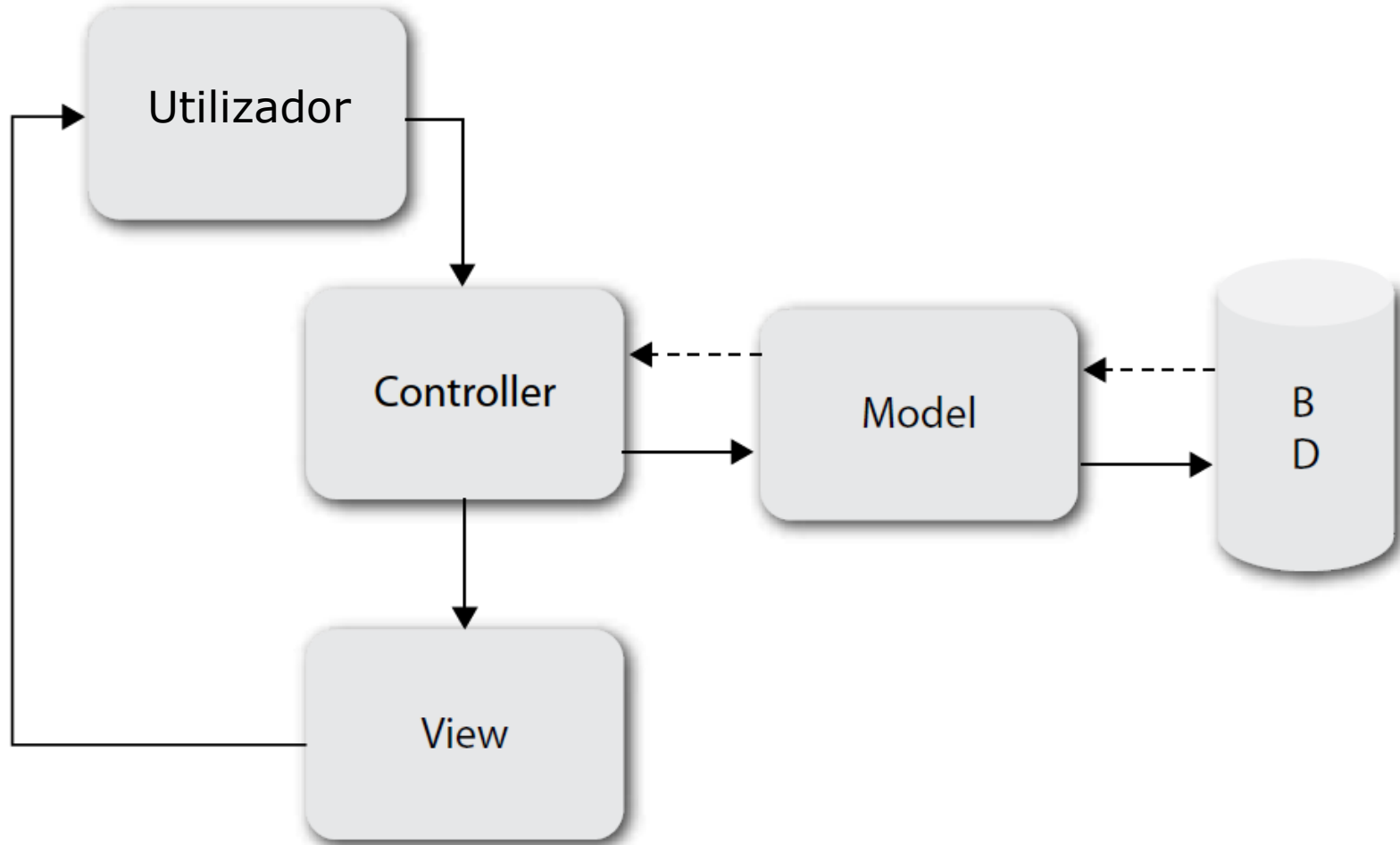
NÃO DEVEM conter lógicas complexas (fica para os Controllers), nem lógicas de manutenção de dados (fica para os Models)

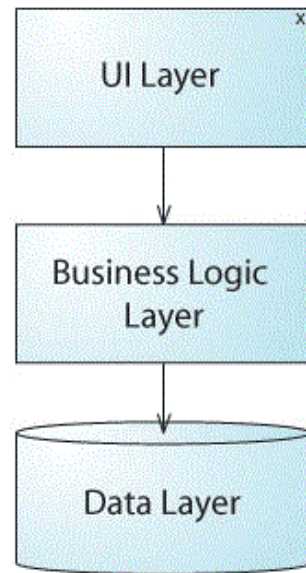
□ MVC

PADRÃO MVC

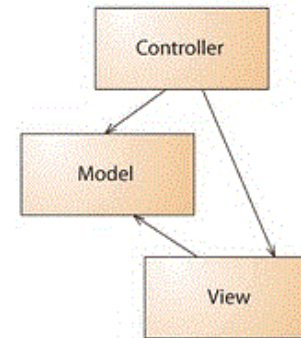


□ Arquitetura MVC

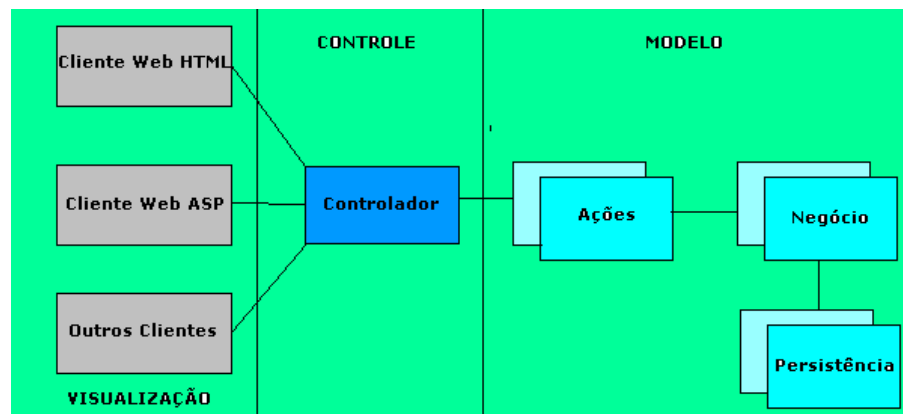




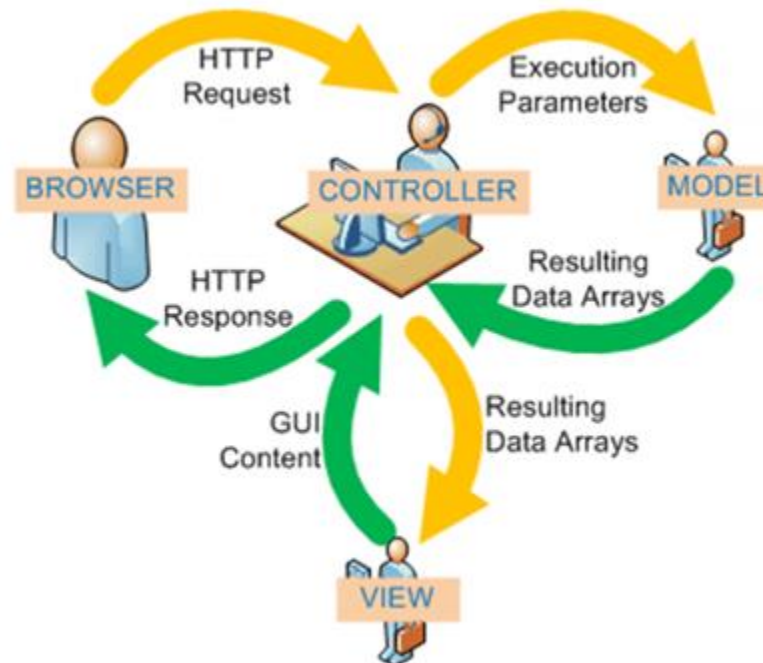
3 camadas



MVC



Arquitetura MVC



MVC Web Model

