

# Modelo Visão e Controle (MVC)

Prof. Rodrigo Attique



<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>

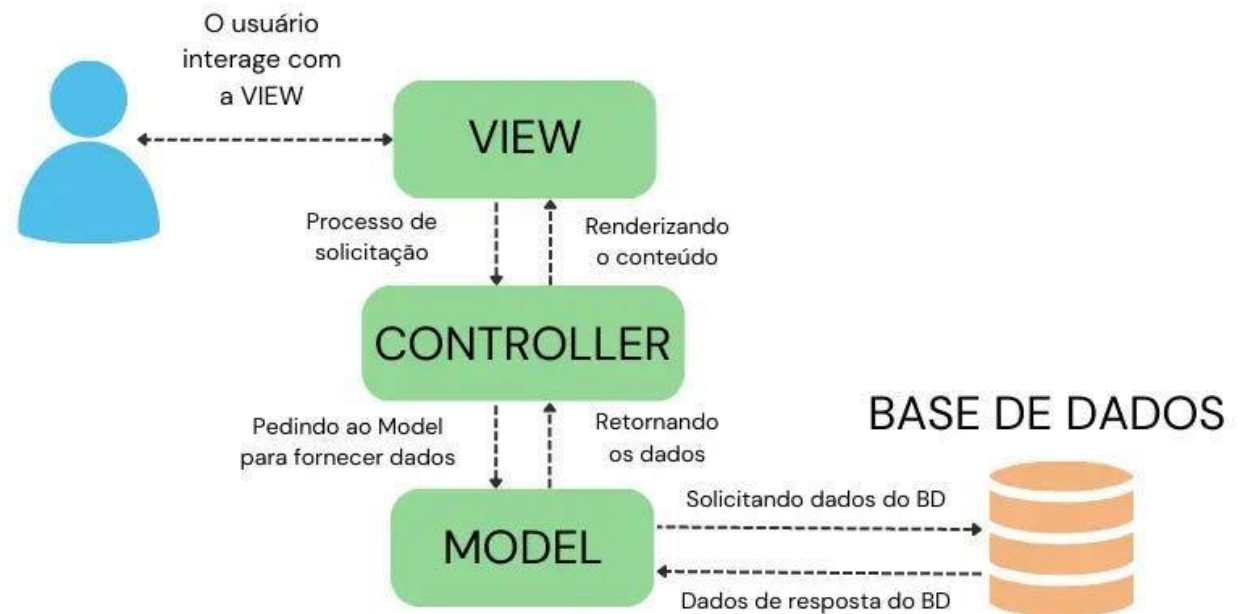
# MVC

---

- Quer dizer Modelo Visão e Controle
- Um projeto sempre começa pequeno e pode crescer exponencialmente.
- Usar padrões de projeto é fundamental para garantir a escalabilidade, manutenção e bom funcionamento do projeto.

# Exemplo prático

- Modelo: Responsável pelos dados, só ele conversa com o banco de dados.
- Visão: Camada que interagem somente com o usuário mas não pode conversar com o modelo diretamente.
- Controle: Camada intermediária entre o modelo e a visão.
- Quando o usuário faz uma solicitação na visão o controle pede os dados para o modelo que apresenta novamente na visão.



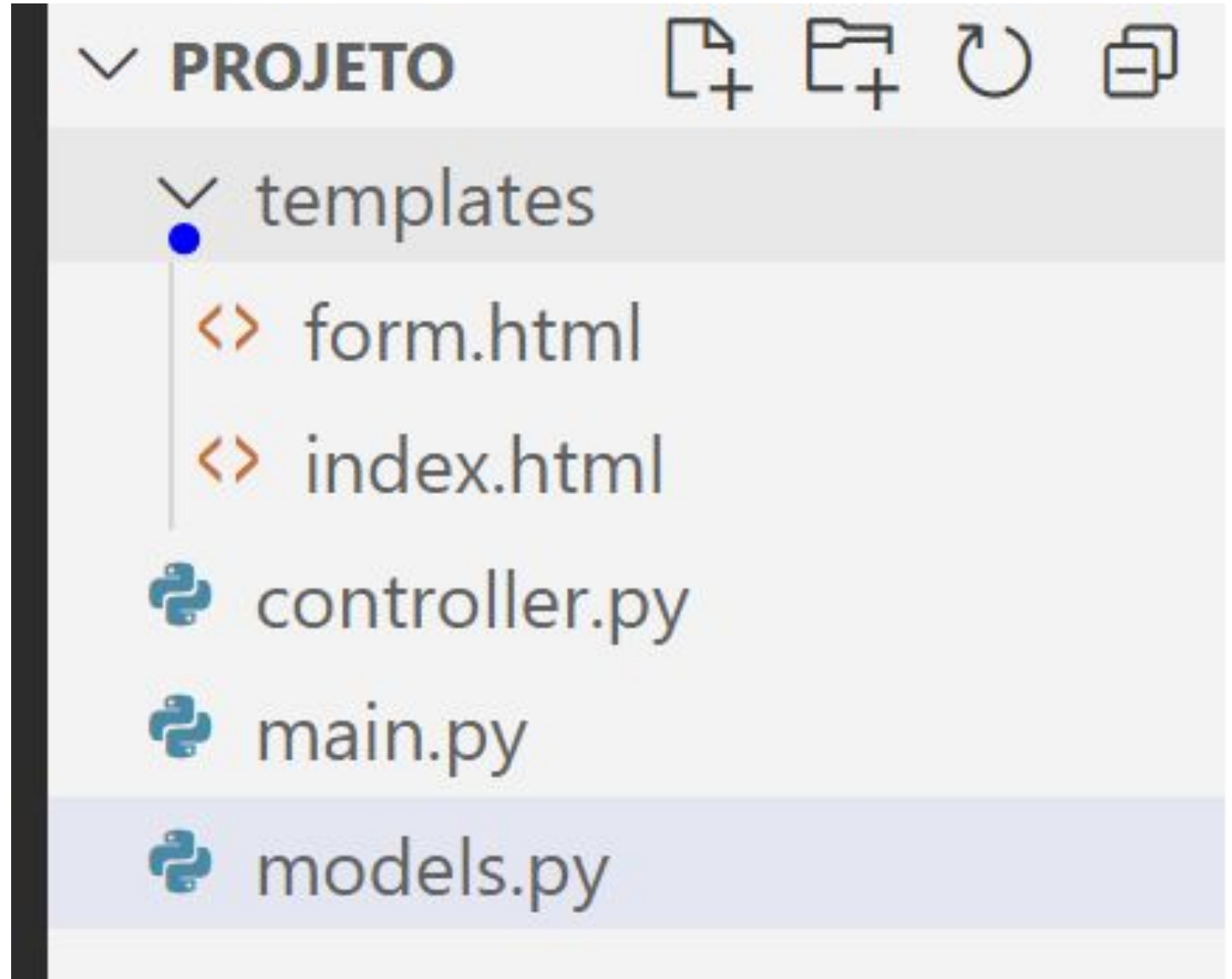
# Organizando nosso projeto

- Podemos organizar nosso projeto de várias maneiras usando MVC.
- A mais simples delas é criando arquivos separados para tratar isso.
- Frameworks modernos como Flask e FastAPI já sugerem o uso de MVC.
- Todos os arquivos de apresentação (html) devem ficar organizados em um diretório chamado templates.
- Enquanto modelos e controladores tem uma abordagem mais flexível.

# Organizando por nomes de arquivos

---

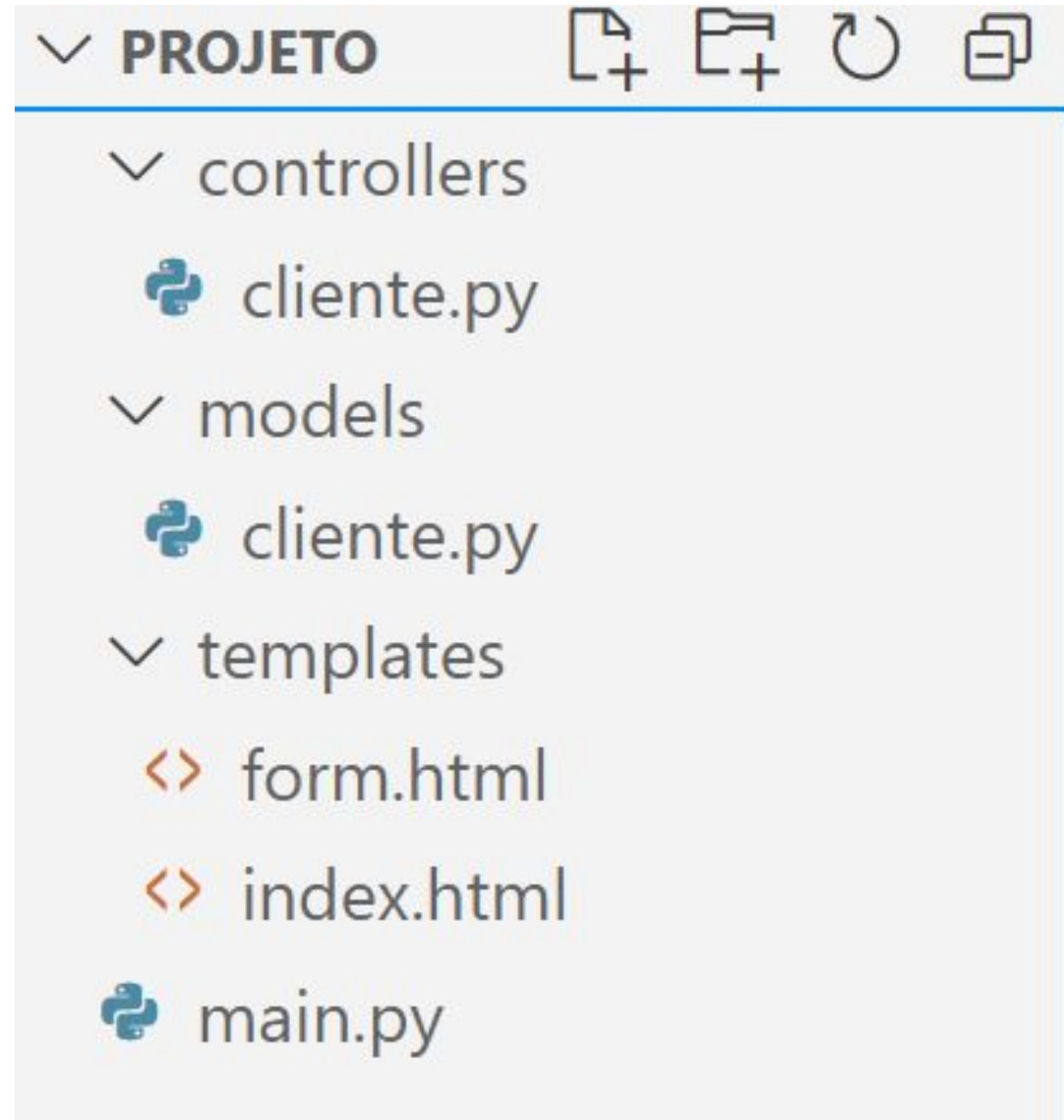
- Se optarmos por organizar nossos arquivos por nomes teríamos um diretório da seguinte forma.



# Organizando por nomes de diretório

---

- Separar MVC por nomes de diretório é mais adequado para projetos de médio porte.
- Mas quem define isso é o projetista conforme requisitos.



# O que for mais adequado

- Podemos criar várias combinações usando MVC.
- O interessante é sempre separar as responsabilidades por arquivos.
- Arquivos de modelo sempre ficam com models.
- Arquivos de controle sempre com controle.

# Um exemplo prático usando Flask

- Nesse exemplo nós criamos pastas com nomes dos casos de Uso e criamos um arquivo de modelo e controle, na pasta templates temos a mesma lógica para as views (visões) HTML

<https://github.com/attiquetecnologia/flaskola>







# Acabou?

---

- Por hora sim!!
- Ainda tem muito a se aprender sobre MVC e nada como criar projetos para praticar.
- O ideal é que se exercite o padrão de proto em pequenos projetos “startups”.