

# BACKEND DEVELOPMENT

Professor: Fábio Kravetz

Unisenai

# ARQUITETURA MVC

- O padrão *Model-View-Controller* (MVC) é uma arquitetura de *software* popular para a construção de aplicações com interface de usuário;
- **Controller:** camada intermediária que trata a lógica da aplicação e, processa as requisições do usuário, interage com o *Model*, e seleciona a *View* que será exibida;
- **Model:** Gerencia dados e lógica de negócios. Interage com o banco de dados (SQLAlchemy / MySQL);
- **View:** É a camada de apresentação, responsável por exibir os dados para o usuário final.

# ARQUITETURA MVC

- *Model* é responsável por toda a interação com o BD. Ele conterá a lógica para buscar e salvar usuários, executando as queries diretamente;
- *View* é a camada de apresentação (Páginas HTML);
- *Controller* é o cérebro do sistema e trabalha com as requisições do usuário. O Controller chama a Model para buscar ou salvar dados e, na sequência renderiza a View apropriada.

# FLASK + MVC

- Flask tem como características a flexibilidade e a simplicidade, já o MVC oferece uma estrutura robusta para a organização do código;

Vantagens	Descrição
Organização e Manutenibilidade	A separação das responsabilidades torna o código mais organizado
Escalabilidade	A modularidade permite que novas funcionalidades sejam incluídas sem grandes problemas
Reutilização de código	A lógica de negócio encapsulada no Model pode ser reutilizada por diferentes <i>Controllers</i> e <i>Views</i>
Desenvolvimento paralelo	A arquitetura MVC permite que desenvolvedores tanto de <i>backend</i> quanto de <i>frontend</i> trabalhem de maneira independente.



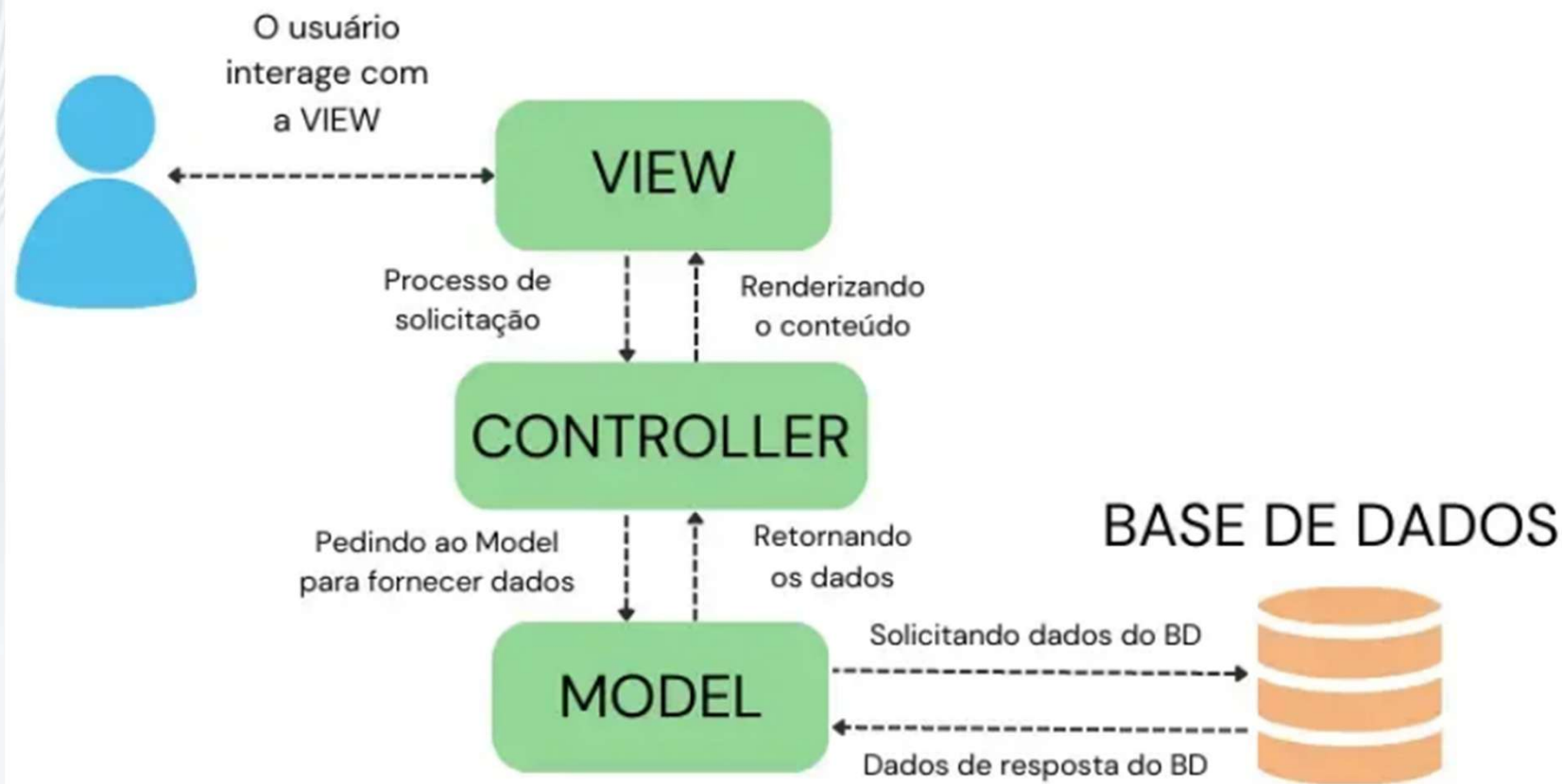
# FLASK + MVC

Desvantagens	Descrição
Complexidade	Em aplicações pequenas e com poucas funcionalidades, a criação da estrutura MVC pode ser inadequada.
Curva de aprendizagem	A compreensão e implementação do padrão MVC, normalmente, adiciona uma camada extra de complexidade ao aprendizado do <i>framework</i> .
Ausência de padrão	A ausência de uma estrutura MVC imposta pode levar o desenvolvedor a diferentes interpretações e implementações.
Aumento de código	A estrutura MVC introduz uma quantidade maior de código, como criação de diretórios, arquivos de inicialização e configuração da estrutura do projeto.

# FLASK + MVC

- Quando utilizar Flask com MVC?
  - Aplicações de médio e grande porte;
  - Equipes de desenvolvimento;
  - Projetos com perspectiva de crescimento;
  - APIs RESTfull;

# ARQUITETURA MVC



# ARQUITETURA MVC

```
Flask_Aula_05_MVC/
├── app.py                # Arquivo principal que inicializa o Flask
├── controllers/
│   └── user_controller.py # Controladores das rotas
├── models/
│   └── user.py           # Modelos de dados (User)
├── views/
│   └── templates/
│       ├── index.html   # Página principal
│       └── contact.html # Formulário de contato
└── config.py            # Configurações da aplicação
```