

# DJANGO

---

MVT & MVC

---



# O QUE É MVC?

MVC é a sigla para Model-View-Controller (Modelo-Visão-Controlador). É um padrão de arquitetura de software que separa a aplicação em três componentes principais: Modelo, Visão e Controlador.

**01**

**MODEL**

---

Este é o componente que lida com todos os dados, lógica e regras do aplicativo. Um modelo representa um objeto ou um conjunto de objetos. Ele pode acessar os dados, manipulá-los e enviar para a visão.

**02**

**VIEW**

---

A visão é responsável pela representação gráfica dos dados, ou seja, como os dados são exibidos ao usuário. Ela não contém qualquer lógica de negócios, apenas a lógica de apresentação.

**03**

**CONTROLLER**

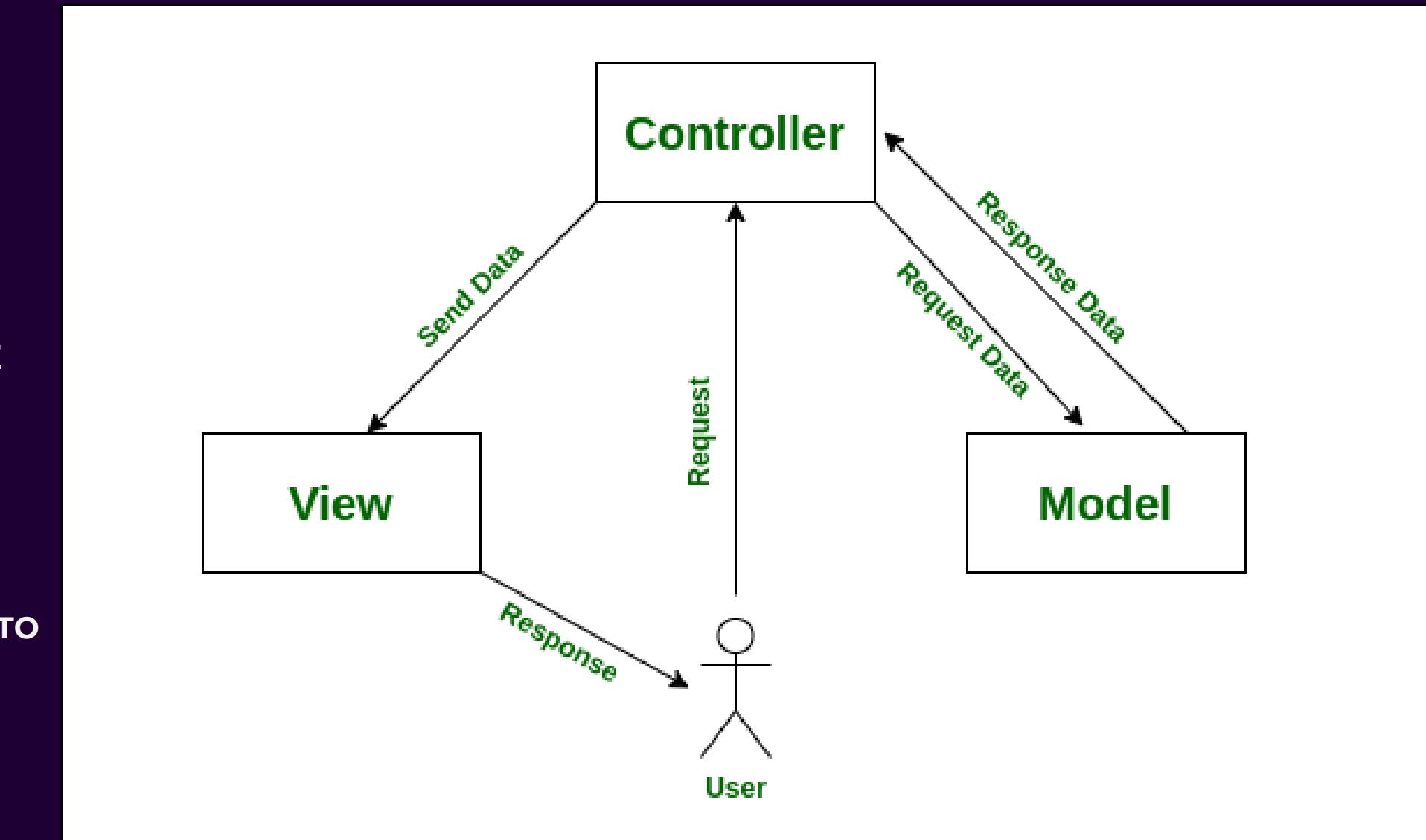
---

O controlador atua como um intermediário entre o Modelo e a Visão. Ele processa todas as solicitações e manipula os dados usando o componente do modelo e exibe os dados na visão.

**01** TODA PARTE RELACIONADA  
A REGRAS DE NEGÓCIOS.

**02** TODA PARTE DE VISUALIZAÇÃO E  
POSTAGEM DE DADOS

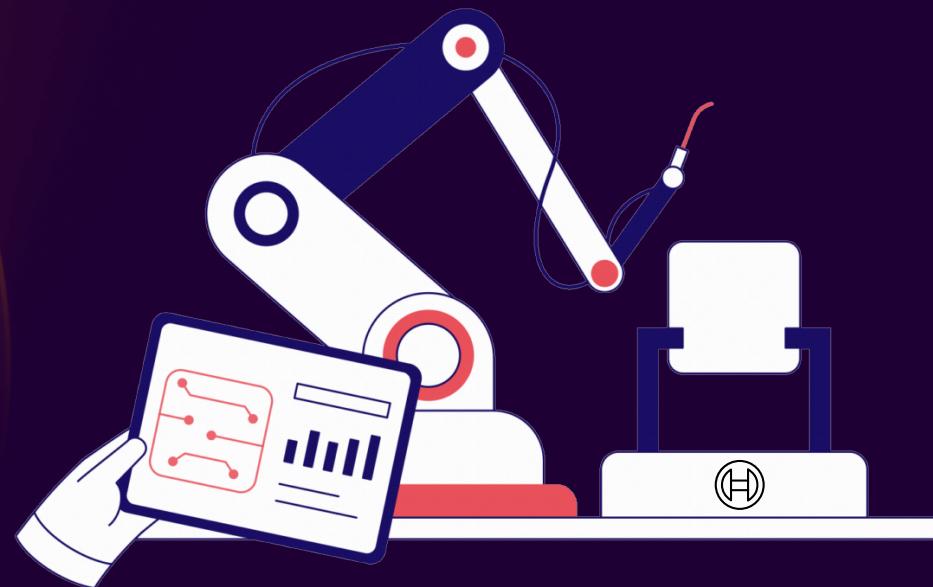
**03** TODA PARTE LÓGICA DE  
TRATAMENTO E PROCESSAMENTO



## MODEL

**01**

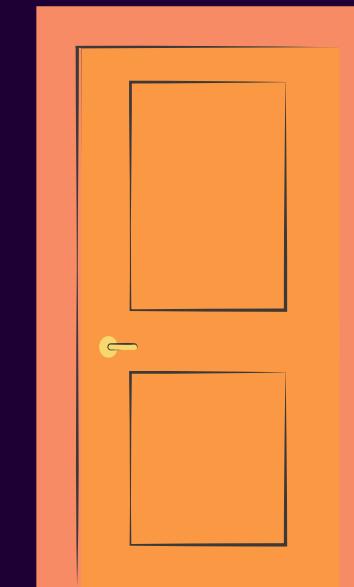
MÁQUINA QUE  
FÁBRICA O PRODUTO



## VIEW

**02**

QUEM USA/CONSUME  
O PRODUTO



## CONTROLLER

**03**

GERENTE DA FÁBRICA  
QUE GARANTE  
QUALIDADE.



# IMPORTÂNCIA

O uso do MVC é essencial para a criação de aplicações mais limpas, flexíveis e de fácil manutenção.

---

01

## **SEPARAÇÃO DE RESPONSABILIDADES.**

O padrão MVC separa claramente as diferentes partes da aplicação (dados, lógica de negócio e interface com o usuário), facilitando o entendimento e a manutenção do código.

---

02

## **FACILIDADE E ESCALABILIDADE.**

Como as camadas são independentes, o padrão MVC permite que diferentes desenvolvedores trabalhem em paralelo em diferentes partes do sistema (um pode trabalhar no Model, outro na View e outro no Controller).

# MVT

---

O Django é baseado na arquitetura MVT (Model-View-Template). MVT é um padrão de design de software para o desenvolvimento de uma aplicação web.

## 01

---

### MODEL

O Model vai atuar como a interface dos seus dados. É responsável pela manutenção dos dados. É a estrutura lógica de dados por trás de todo o aplicativo e é representada por um banco de dados

## 02

---

### TEMPLATE

O Template é a parte que demonstra os dados, além de toda parte relacionada a exibição de dados em um HTML/JS.

## 03

---

### VIEW

View funciona como o controlador, que recebe os dados do Template, além de moldar para exibi-los.

# MVT VS MVC

A diferença entre MVT (Model-View-Template) e MVC (Model-View-Controller) está principalmente na **organização e responsabilidades das camadas na aplicação**.

## VIEW

A "View" é responsável tanto pela lógica de exibição quanto pela interação do usuário.

A "View" no MVT tem o papel de controlar a lógica, processando as requisições do usuário, interagindo com o Model e decidindo qual Template deve ser renderizado.

## TEMPLATE

Não há uma camada específica de "Template" em MVC. A camada de View lida diretamente com a exibição dos dados e com a renderização da interface.

Em MVT, a Template é uma camada separada que cuida da exibição e do layout da aplicação, gerando o código HTML para ser enviado ao usuário.

## CONTROLLER

O Controller é responsável pela lógica de controle, processando as entradas do usuário, interagindo com o Model e decidindo qual View deve ser exibida.

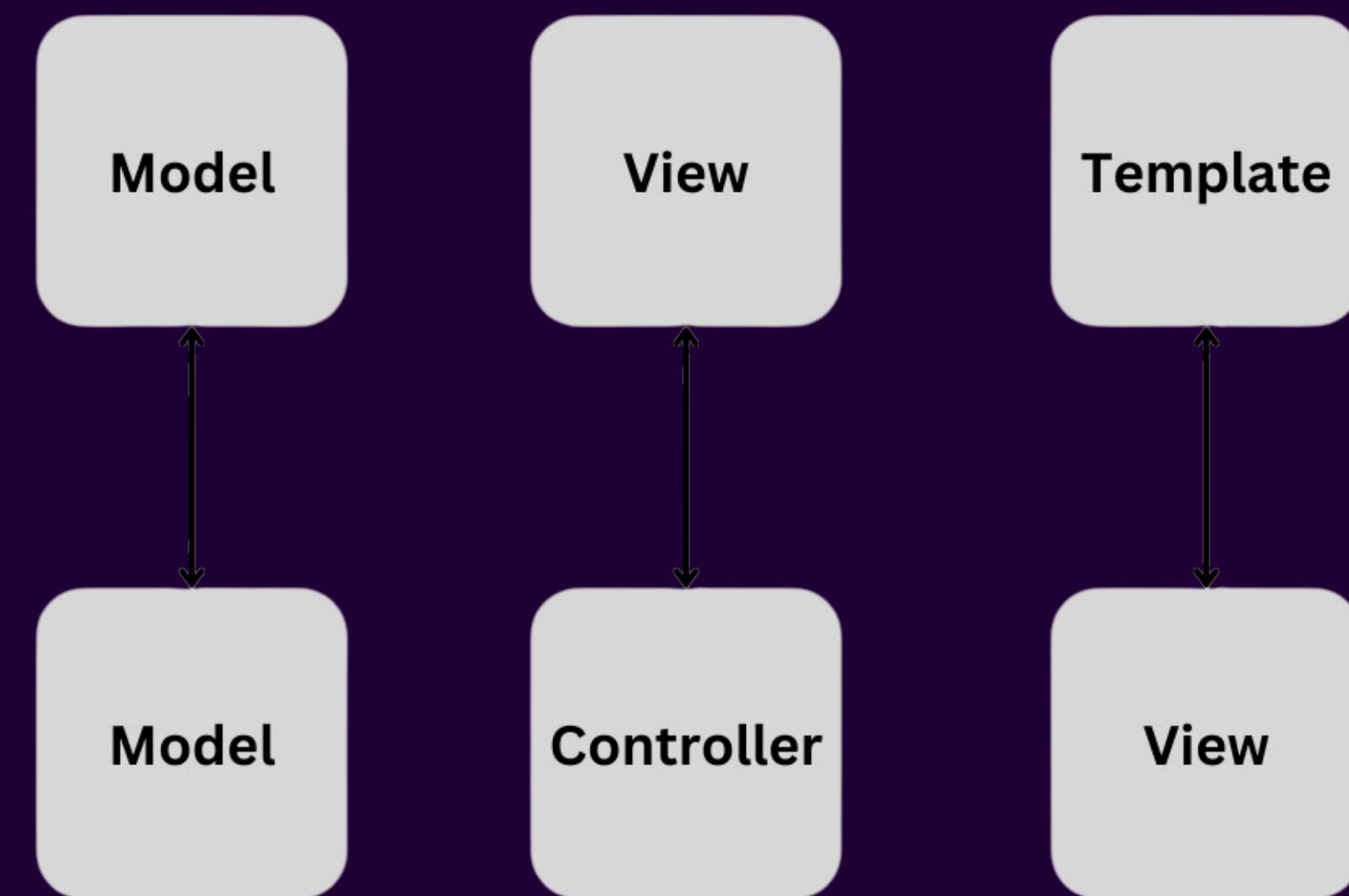
No MVT, o Controller não existe com esse nome, pois sua função é realizada pela View. A "View" no MVT é responsável pela lógica de controle, recebendo a requisição do usuário.

# MVC & MVT

---

A principal diferença entre MVT e MVC é que no MVC, a View lida com a exibição e interação com o usuário, enquanto o **Controller** gerencia a lógica de controle. Já no MVT, a **View** é responsável pela **lógica de controle e a Template** cuida da exibição, separando a apresentação dos dados da lógica.

## MVC VS MVT



MUITO OBRIGADO! :}

---

