**Gabriel Bosco Deolindo – 2DSTB\_C**

**Models:**

Representam a estrutura dos dados e interagem com o banco de dados. No Django, eles são classes Python que definem tabelas no banco automaticamente. como funciona:

* Define os campos do banco de dados.
* Cria relações entre tabelas.
* Interage com o banco sem precisar escrever SQL.

**Templates:**

São arquivos HTML que contêm marcações para exibir dados dinamicamente. Eles evitam que o código Python fique misturado com o HTML. Como funciona:

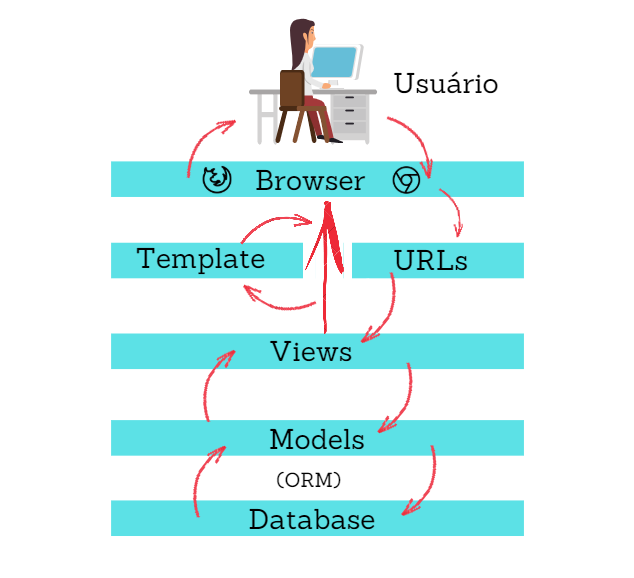
* Exibe as informações vindas da View.
* Utiliza a linguagem de template do Django para lógica simples
* Separa a interface visual da lógica do código.

**View**:

Funciona como o Controller no padrão MVC. Ela processa as requisições, busca os dados no Model e os envia para o Template. Como funciona:

* Recebe requisições do usuário.
* Busca dados do banco de dados.
* Passa os dados para um Template renderizar.

Toda esta arquitetura é interligada e conversam entre si. Uma depende da outra para realizar um determinado serviço e, no final, executar a tarefa que o usuário solicitou.



**Relação do MVC e do MTV:**

**Model (MTV) = Model (MVC):**  
Ambas as camadas têm a mesma função: mapear e gerenciar os dados no banco de dados do projeto.

**Template (MTV) = View (MVC):**  
São responsáveis por exibir informações para o usuário, geralmente utilizando páginas HTML para a interface.

**View (MTV) = Controller (MVC):**  
Embora tenham responsabilidades semelhantes, há diferenças conceituais. No Django, a View assume parte do papel do Controller, processando requisições e conectando Model e Template.