**O QUE SÃO MODELS?**

Os models são componentes fundamentais do Django, um framework de desenvolvimento web em Python. Eles são responsáveis por definir a estrutura e o comportamento dos dados em um aplicativo Django.

um "model" é uma classe Python que representa uma tabela em um banco de dados relacional.

**Tipos de models:**

**Model CharField:**

Este campo é usado para armazenar strings curtas, como nomes, títulos ou descrições curtas com (um valor máximo de 254 caracteres).

**Model TextField:**

E usado para armazenar texto longo, diferente do charfield ele e uma ferramenta para armazenar grandes quantidades de texto

**Model IntegerField:**

Esse e usado para armazenar números inteiros, Este campo é útil quando você precisa armazenar valores que são números inteiros, como a idade de uma pessoa, a quantidade de itens em estoque ou o número de pedidos feitos. O IntegerField só aceita números inteiros, sem casas decimais. Ele pode armazenar números positivos, negativos

**Model FloatField:**

Usado para armazenar números decimais, em um banco de dados. Ele permite que você armazene números com casas decimais, como números fracionários ou valores monetários.

**Model BooleanField:**

Armazena valores booleanos (Verdadeiro/Falso).

**Model DateField:**

Usado para armazenar datas (sem informações de hora). Útil para representar datas de nascimento, datas de eventos

**Model DateTimeField:**

aceita valores no formato de data e hora do Python (datetime.datetime). Ele permite armazenar datas e horas passadas, presentes e futuras, e pode ser útil em muitos cenários de aplicativos diferentes.

**Model ForeignKey:**

é um tipo especial de campo em modelos do Django usado para estabelecer uma relação de chave estrangeira entre dois modelos. Ele é usado quando você deseja criar uma associação entre dois modelos, onde um modelo "possui" ou "pertence a" outro modelo.

**ManyToManyField:**

é um tipo de campo em modelos do Django usado para representar uma relação muitos-para-muitos entre dois modelos. Ele é usado quando um objeto de um modelo pode estar associado a vários objetos de outro modelo e vice-versa.

**PositiveIntegerField:**

é um tipo de campo em modelos do Django usado para armazenar números inteiros positivos em um banco de dados. Ele é semelhante ao **IntegerField**, mas com a diferença de que ele só aceita valores inteiros positivos, excluindo zero e números negativos.

**Model EmailField:**

é um tipo de campo em modelos do Django usado para armazenar endereços de e-mail em um banco de dados. Ele valida automaticamente se o valor fornecido corresponde a um formato de endereço de e-mail válido.

**Model ImageField:**

é um tipo de campo em modelos do Django usado para armazenar imagens em um banco de dados. Ele é usado quando você precisa permitir que os usuários carreguem imagens em seu aplicativo.

**OneToOneField:**

é um tipo de campo em modelos do Django que estabelece uma relação um-para-um entre dois modelos