



Você está em

**Guia de Django » Banco de dados**

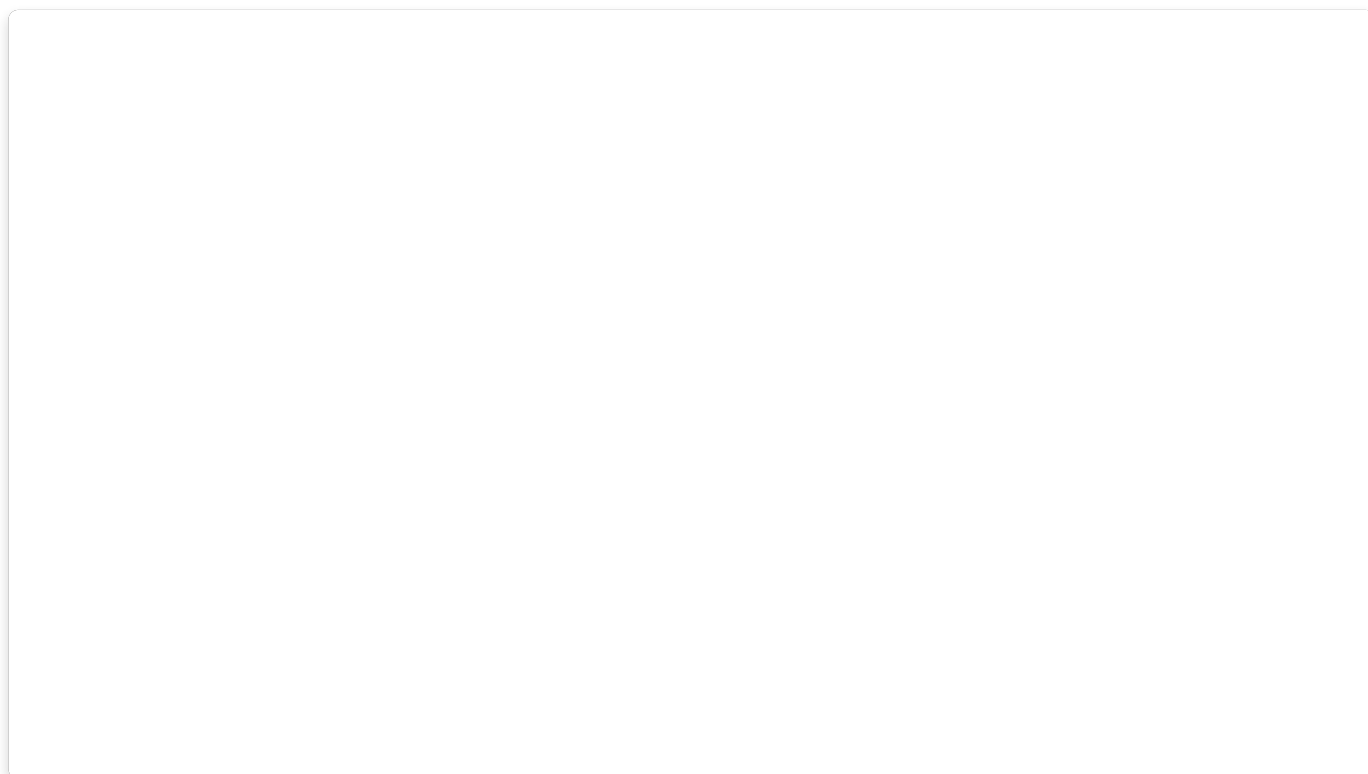
## 4. Adicionando relacionamento N:N



Marcar como concluído



Tire sua dúvida



Agora precisamos criar o relacionamento entre as entidades Produto e Categoria. Para isso, veremos como implementar este relacionamento utilizando o Django.

05:24 min

[Ir para o código](#)

## Adicionando Categoria a Produto



80





Você está em

**Guia de Django » Banco de dados**

`models.py` do projeto iremos adicionar um campo categoria dentro do model `Produto` como podemos ver abaixo:

```
1 class Produto(models.Model):
2     descricao = models.CharField(max_length=50, null=False)
3     preco = models.FloatField(null=False)
4     marca = models.ForeignKey(Marca)
5     categoria = models.ManyToManyField(Categoria)
```

**Linha 5:** Definimos que um produto possui várias categorias relacionadas a ele, usando para isso o método `ManyToMany`.

Agora precisamos adicionar o atributo `categoria` ao array `fields` da classe `ProdutoForm`, localizado no arquivo `views.py`, como podemos ver abaixo:

```
1 class ProdutoForm(ModelForm):
2     class Meta:
3         model = Produto
4         fields = ['descricao', 'preco', 'marca', 'categoria']
```

**Linha 4:** Adicionamos o atributo `categoria` para indicar que um produto possui uma ou mais categorias relacionadas a ele.

Feito isso, será criada uma tabela no banco de dados responsável por armazenar os IDs dos produtos e categorias relacionadas como podemos ver nas **Figuras 1 e 2**:

```
+-----+
| Tables_in_relacionamentonn |
+-----+
| app_categoria               |
| app_marca                   |
| app_produto                 |
| app_produto_categoria       |
| auth_group                  |
| auth_group_permissions      |
+-----+
```



80





Você está em

**Guia de Django » Banco de dados****Figura 1.** Tabela que armazena o relacionamento N:N

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
produto_id	int(11)	NO	MUL	NULL	
categoria_id	int(11)	NO	MUL	NULL	

**Figura 2.** Definição da tabela que armazena o relacionamento N:N

Agora, sempre que um produto for cadastrado no banco de dados, uma lista de categorias que podem ser relacionadas a ele será exibida, como podemos ver na **Figura 3**:

**Descricao**

  
**Preco**  
**Marca**  
**Categoria**

Informática

Brinquedo

**Salvar**

**Figura 3.** Cadastro de produtos com relação entre categorias

## Aprenda também:

○

Guia **Carreira Programador Python com Django**

80





Você está em

**Guia de Django » Banco de dados**

Avalie o **curso**     

Suporte ao aluno - Tire a sua dúvida.