***Comandos do Framework Django***

**Set-ExecutionPolicy -ExecutionPolicy RemoteSigned** →

Comando utilizado para fazer que o sistema operacional Windowns aceite comandos externos no terminal **( PowerShell )**.

**python -m venv ll\_env** → Comando utilizado para

criar um ambiente isolado no sistema.

**source ll\_env/bin/activateb**

**. ll\_env/Scripts/activate**  → Comando

utilizado para ativar o ambiente virtual.

**deactivate** → Comando utilizado para sair do ambiente

virtual.

**pip install Django** → Estando online em um Ambiente

Virtual podemos utilizar esse comando para instalar o Framework Django.

**django-admin --version** → Comando utilizado para ver a

versão do Django.

**django-admin startproject** nome\_da\_pasta **.** → Comando

utilizado para inicializar um projeto.

**python manage.py migrate** → Comando utilizado para

vereficar qual banco de dados será utilizado de acordo com o que foi deixado como padrão no **settings.py**. Criando assim um banco de dados inicial.

**python manage.py runserver** → Comando utilizado para

rodar o Django no navegador na porta 8000.

**python manage.py startapp** nome\_do\_aplicativo\_principal\_no\_plural →

Comando utilizado para criar um app no Django.

**python manage.py makemigrations** nome\_do\_aplicativo\_principal → Comando utilizado para criar um novo tópico no projeto.

**python manage.py createsuperuser** → Comando

utilizado para criar uma conta admin no ambiente admin do django.

**<http://127.0.0.1:8000/admin>**

**Interação no terminal com o Banco de Dados:**

**1 - python manage.py shell** → Comando utilizado para

entrar numa shell interativa no python.

**2 - from** nome\_do\_aplicativo\_principa**.models import** nome\_da\_tabela →

Comando utilizado para entrar na tabela do Banco de Dados.

**3 -** nome\_da\_tabela**.objects.all()** → Comando

utilizado para visualizar as colunas criadas na tabela.

**4 - for** var **in** nome\_da\_lista**:**  **print(f‘**{var.id} **=** {var}**’)** → Criação de um

laço de repetição for, onde será apresentado o respectivo id de cada coluna da tabela.

**5 -** nome\_da\_tabela**.objects.get(id =** id\_da\_coluna**)** →

Comando utilizado para visualizar uma coluna específica da tabela.

**6 -**var **=** nome\_da\_tabela**.objects.get(id =** id\_da\_coluna**)** →

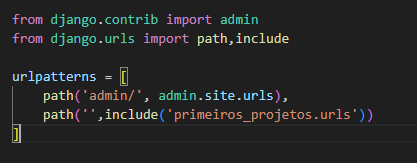
Criação de uma variável recebendo uma coluna específica da tabela.

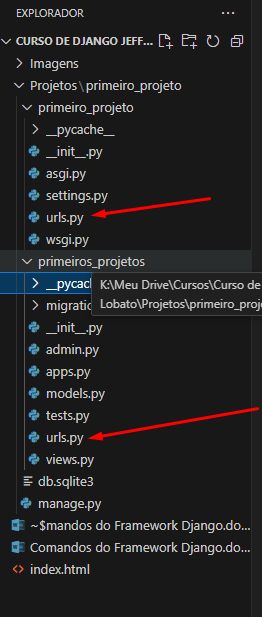
variavel\_com\_a\_coluna\_específica**.**nome\_da\_coluna →

Comando utilizado para mostrar os dados de de uma coluna específica.

***Rotas na Página HTML.***

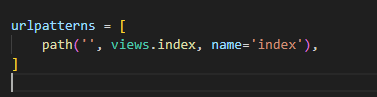
**1 -** Na pasta do projeto no arquivo **urls.py:**



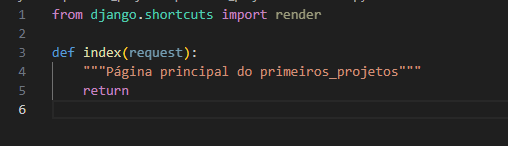
****

**2 - Agora é necessário copiar o arquivo urls da**

**pasta do projeto para a pasta do app:**

****

**3 - Ir no arquivo views.py que se encontra na pasta app e fazer as seguintes alterações:**

****

**4 - Crie uma pasta chamada “Templates” na pasta do app. Dentro da pasta “Templates” crie outra pasta com o mesmo nome da pasta do app, e dentro desta coloque o arquivo index.html.**

***Criando blocks.***

**index.html** ( Arquivo Filho ) →

{% extends "primeiros\_projetos/base.html" %}

{% block content %}

    <p>Essa aplicação te ajudará a organizar seus estudos.</p>

{% endblock content %}

**base.html** ( Arquivo Pai ) →

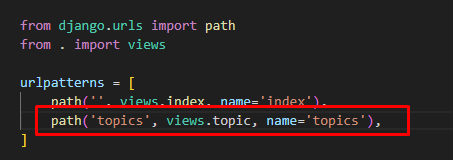
<a href="{% url 'index' %}"> <h1>Primeiro Projeto</h1></a>

{% block content %}{% endblock content %}

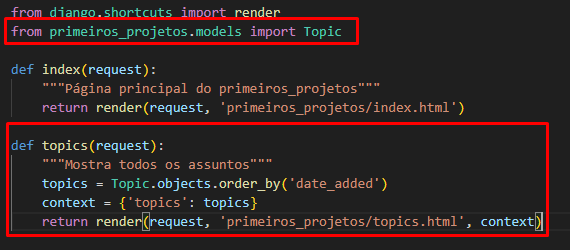
<footer>TESTE RODAPÉ</footer>

***Estrutura for Python in Django.***

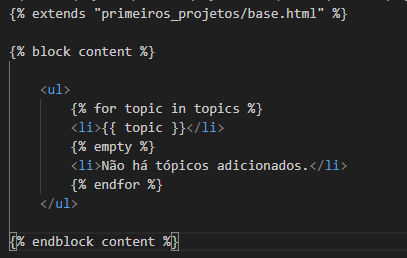
**1 - No arquivo urls.py da pasta app faça as seguintes alterações:**



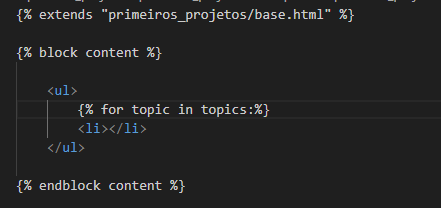
**2 - Agora faça as seguintes alterações no arquivo views.py:**

****

**3 - Crie um arquivo para a tabela mencionada no urls.py e views.py. Lembrando que essa tabela é a tabela criada no models.py. Esse arquivo deve seguir as mesmas caracteristicas o arquivo index.html que criamos; deve seguir a estrutura de blocks:**

****

**4 - Agora é só criar a estrutura for no block:**

****

**5 - Rode o Django e digite a seguinte url:**