

# Iniciando APP

Inicie seu app django digitando como está abaixo.

```
● (venv) PS E:\praticando\pratica 3> python manage.py startapp Usuario
```

Caso encontre algum erro desse tipo:

```
FileNotFoundError: [Errno 2] No such file or directory: 'E:\\praticando\\pratica 4\\manage.py'
```

Digite cd e o nome de seu projeto.

```
● (venv) PS E:\praticando\pratica 4> cd cadastroCrud
```

**ATENÇÃO->** Note que APP é diferente de projeto, o projeto criamos com o comando **django-admin** já o app, criamos com o comando **startapp** dentro de um projeto, podem ter vários APPS.

*python manage.py startapp SEUAPP  
cd SEUPROJETO*

# Adicionando APP ao settings

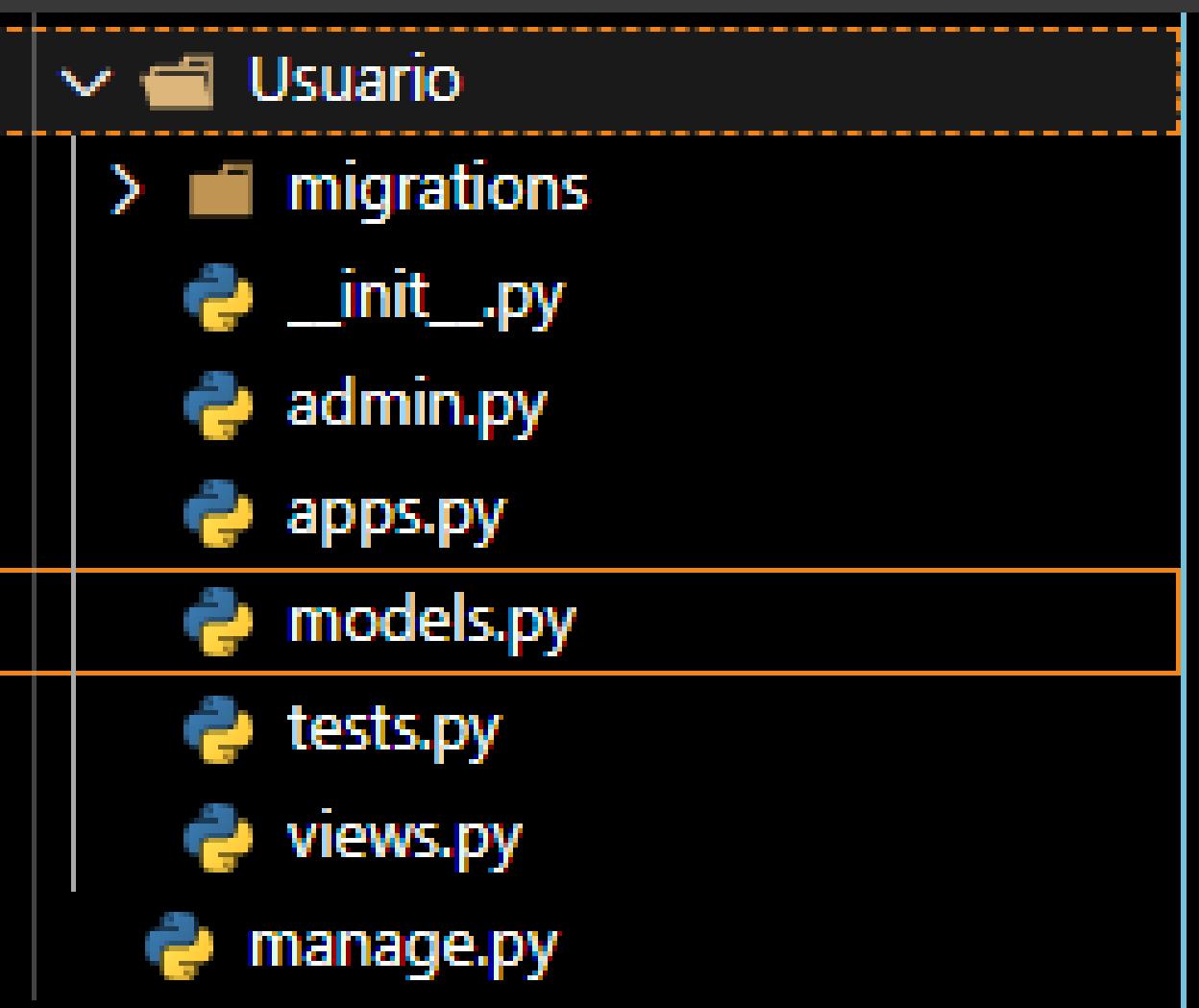
Vá até o settings novamente, e adicione seu app e o rest\_framework

```
20
21 INSTALLED_APPS = [
22     'django.contrib.admin',
23     'django.contrib.auth',
24     'django.contrib.contenttypes',
25     'django.contrib.sessions',
26     'django.contrib.messages',
27     'django.contrib.staticfiles',
28     'rest_framework',
29     'Usuario' # NOME DO SEU APP
30 ]
31
```

**ATENÇÃO->** Sempre que um novo APP for criado, ele deve ser adicionado ao settings.py

# Criando a model

Vá até o arquivo **Models.py** em seu APP



# Dentro da model:

```
from django.db import models

# Define um modelo chamado 'Usuario' que representa um usuário no banco de dados
class Usuario(models.Model):
    class Meta:
        # Define o nome da tabela no banco de dados como 'usuario'
        db_table = 'usuario'

    # Define um campo de caractere para armazenar o nome do usuário (máximo de 100 caracteres)
    nome = models.CharField(max_length=100)

    # Define um campo de caractere para armazenar o sobrenome do usuário (máximo de 100 caracteres)
    sobrenome = models.CharField(max_length=100)

    # Define um campo inteiro para armazenar a idade do usuário (padrão é 0)
    idade = models.IntegerField(default=0)

    # Define um campo de imagem para armazenar a foto de perfil do usuário (as imagens serão enviadas para o diretório 'foto de perfil')
    foto = models.ImageField(upload_to='foto de perfil')

    # Método especial que retorna uma representação de string do objeto, neste caso, o nome do usuário
    def __str__(self):
        return self.nome
```

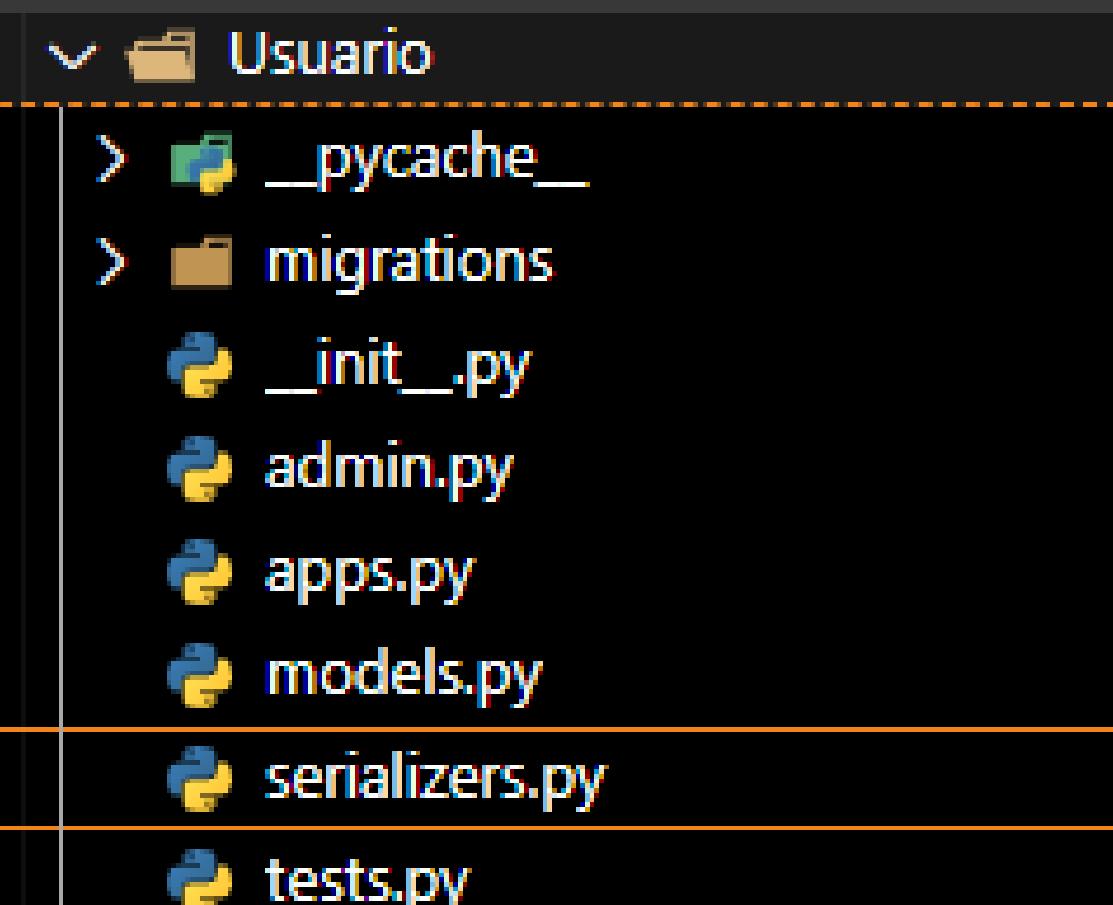
Após terminar de escrever a model, faça as migrações.

```
○ (venv) PS E:\praticando\pratica 4\cadastroCrud> python manage.py makemigrations
```

```
○ (venv) PS E:\praticando\pratica 4\cadastroCrud> python manage.py migrate
```

# Criando serializador

dentro de seu APP crie um arquivo **serializers.py**



# Dentro do serializer:

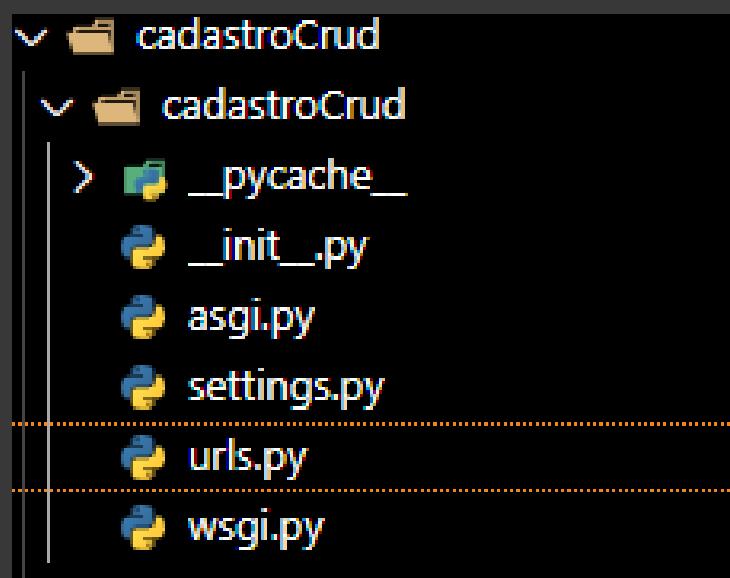
```
from rest_framework import serializers
from .models import Usuario # model criada anteriormente

class UsuarioSerializer(serializers.ModelSerializer):

    class Meta:
        model = Usuario
        fields = '__all__'
```

# Configurando urls principais

Na raiz de seu projeto, acesse **urls.py**



Dentro de **urls.py**

```
from django.contrib import admin
from django.urls import path,include # Por padrão virá importando só o path, adicione o include.

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('Usuario.urls')) # urls do app que você criou
]
```