

Fluxograma Completo – Criação de Projeto Python/Django (Melhorado, linha a linha)

INÍCIO

Ponto de partida.

1. Instalar dependências

Verifique/instale Python, pip. Ex.:

```
'python --version' e 'pip --version'.
```

2. Criar ambiente virtual

Criar venv:

```
'python -m venv venv'.
```

3. Ativar ambiente virtual

Windows: 'venv\Scripts\activate' | Linux/Mac: 'source venv/bin/activate'.

4. Instalar Django

Dentro do venv:

```
'pip install django'.
```

5. Criar projeto principal

Comando:

```
'django-admin startproject nome_projeto' e entrar na pasta do projeto.
```

6. Testar servidor inicial

Executar:

```
'python manage.py runserver' e acessar 'http://127.0.0.1:8000'.
```

7. Criar aplicativo (app)

Comando:

```
'python manage.py startapp nome_app'.
```

8. Registrar app em settings.py

Adicionar o app em INSTALLED_APPS no settings.py.

9. Criar models

Definir models em nome_app/models.py e planejar o esquema do banco.

10. Criar e aplicar migrações

Comandos:

`'python manage.py makemigrations'` e `'python manage.py migrate'`.

11. Criar superusuário

Comando:

`'python manage.py createsuperuser'` para acessar `/admin`.

12. Criar views e urls

Definir views em `views.py`, criar `nome_app/urls.py` e incluir no `urls.py` do projeto.

13. Configurar templates

Criar pasta `'templates'` e configurar `TEMPLATES['DIRS']` em `settings.py`.

14. Configurar static files

Criar pasta `'static'`, configurar `STATIC_URL` e coletar com `'collectstatic'` em produção.

15. Formulários e validação

Criar `forms.py`, usar `ModelForm/Forms` para entrada de dados segura.

16. Testes e qualidade

Escrever testes unitários e de integração, executar `'python manage.py test'`.

17. Logs e tratamento de erros

Configurar logging e páginas de erro (404, 500).

18. Segurança e configuração

Ajustar `DEBUG=False`, `ALLOWED_HOSTS`, usar variáveis de ambiente (`.env`).

19. Preparar para deploy

Escolher banco (Postgres), configurar WSGI/ASGI, e arquivos estáticos.

20. Deploy

Opções: Render, Railway, DigitalOcean, Heroku, AWS. Configurar CI/CD se desejar.

FIM

Projeto em produção.

Cheatsheet de comandos úteis

Criar venv	<code>python -m venv venv</code>
Ativar venv (Windows)	<code>venv\Scripts\activate</code>
Ativar venv (Linux/Mac)	<code>source venv/bin/activate</code>
Instalar Django	<code>pip install django</code>

Criar projeto	<code>django-admin startproject nome_projeto</code>
Criar app	<code>python manage.py startapp nome_app</code>
Runserver	<code>python manage.py runserver</code>
Migrações	<code>python manage.py makemigrations &&</code> <code>python manage.py migrate</code>
Criar superuser	<code>python manage.py createsuperuser</code>

■ Gerado por ChatGPT – Versão melhorada, passos separados linha a linha.