

Ambiente Virtual em Python

Um **ambiente virtual** é um espaço isolado criado para executar aplicações Python sem interferir na instalação principal do sistema. Ele permite que cada projeto tenha suas **próprias bibliotecas e versões**, evitando conflitos entre dependências.

Esse recurso é especialmente útil quando diferentes partes do código — ou diferentes projetos — utilizam versões distintas de um mesmo pacote. Por exemplo, um projeto pode exigir a versão **X do Django**, enquanto outro depende da versão **Y**. Com ambientes virtuais, é possível instalar e usar ambas as versões no mesmo computador, cada uma restrita ao seu próprio ambiente.

Assim, você garante:

- Independência entre projetos
- Facilidade para testes
- Reprodutibilidade do ambiente
- Organização e controle sobre pacotes e dependência

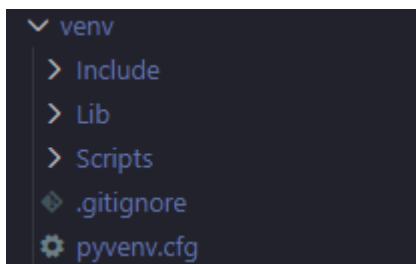
Criando a Venv do Python

Para criar um ambiente virtual devemos iniciar o terminal e aplicar o comando:

```
PS C:\Users\Usuario\Desktop\Pedro\Estudos\Estudos_Python> python  
-m venv venv
```

(No comando acima você pode nomear o seu ambiente da maneira como quiser, mas a boa prática é sempre nomear como venv ou env).

Assim que o ambiente é criado, uma pasta com o mesmo nome dele é criada:



O que significa cada arquivo dentro desta pasta ?

Include

O include contém os arquivos headers (cabeçalhos do Python) que são arquivos que: Contêm a definição do que o Python representa como: Listas, objetos, strings, dicionários etc. permitem que extensões em C sejam compiladas e interajam com o interpretador.

Lib

Armazena todos os pacotes e dependências que forem instaladas no ambiente virtual, dentro dela tem o pip que é o sistema de gerenciamento de pacotes do Python.

Scripts

Contêm scripts e executáveis relacionados ao ambiente virtual, sempre que formos ativar o venv iremos interagir com essa pasta.

Pyvenv.cfg

Mostra todas as informações do ambiente virtual:
Onde está localizado, versão do Python que ele está utilizando

Como ativar o Ambiente ?

Para ativar devemos acessar o arquivo activate.bat dentro da pasta Scripts.

```
C:\Users\Usuario\Desktop\Pedro\Estudos\Estudos_Python> venv\Scripts\activate.bat
```

E para desativar utilizamos o comando Deactivate:

```
(venv) PS C:\Users\Usuario\Desktop\Pedro\Estudos\Estudos_Python> deactivate
```

Requirements.txt

O arquivo requirements é uma boa prática, utilizado para listar todas as bibliotecas que estão instaladas dentro da venv, o comando para gerá-lo é:

```
(venv) PS C:\Users\Usuario\Desktop\Pedro\Estudos\Estudos_Python> pip freeze > requirements.txt
```

Ele tem esse nome por boa prática.