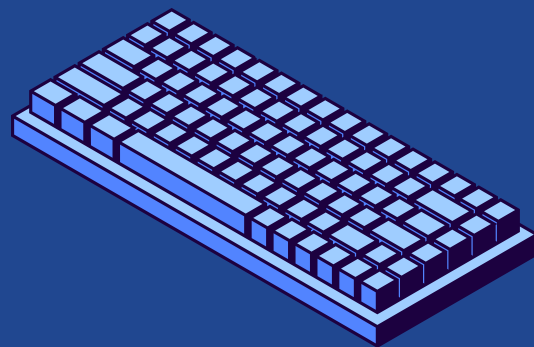
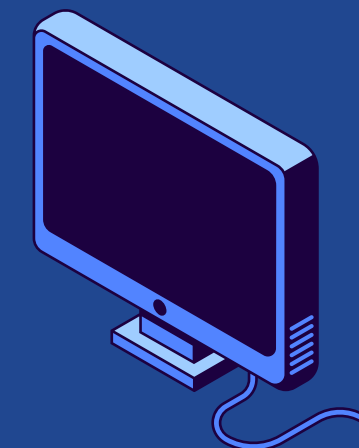


# Testes, Venv e PylInstaller



START



Autor : Gabriel Antonio.

# Testes



## O que é pytest?

- **pytest** é um **framework para testes** em Python, simples e poderoso.
- Serve para **escrever testes automatizados**, garantindo que seu código funcione como esperado.
- Facilita rodar, organizar e verificar resultados dos testes.
- Precisa ter um arquivo cópia para os testes complexos agora coisas simples pode ser direto, é uma boa prática colocar o nome com o prefixo ou sufixo com **test\_** ou **\_test** ex: “**test\_nomedoarquivo.py**” ou “**nomedoarquivo\_test.py**”
- Executamos o pytest via **terminal**: “**pytest nomedoarquivo.py**” ou múltiplos: “**pytest nomedoarquivo01.py nomedoarquivo02.py**”;
- Usamos o comando “**assert**” onde passamos o que esperamos deste resultado:

```
import pytest

def soma(a, b):
    return a + b

def test_soma():
    assert soma(2, 3) == 5
    assert soma(-1, 1) == 0
```

```
def busca_usuario(id):
    # Simulando dados do banco
    usuarios = {
        1: "Maria",
        2: "João",
        3: "Ana"
    }
    return usuarios.get(id, None)

def test_busca_usuario():
    assert busca_usuario(1) == "Maria"
    assert busca_usuario(4) is None
```

```
def get_status_api():
    # Simulando resposta da API
    return {"status": 200, "message": "OK"}

def test_get_status_api():
    resposta = get_status_api()
    assert resposta["status"] == 200
    assert resposta["message"] == "OK"
```

```
[:\xampp\htdocs> pytest test_main.py]
```



# Testes



## O que é SonarQube?

- Plataforma para análise contínua de qualidade de código.
- Detecta bugs, vulnerabilidades, code smells e cobertura de testes.
- Suporta várias linguagens, incluindo Python.

## Processo de instalação:

- Baixe o arquivo, descompacte, navegue até a pasta **bin** e execute o script: **“StartSonar.bat”**;
- Acesse o Sonar com localhost:9000, login:admin, senha:admin;
- Instale o Scanner do SonarQube, crie um arquivo na pasta raiz do projeto com o nome: **“sonar-project.properties”**;
- Execute no terminal o comando: **“sonar-scanner”** lembrando que tem que estar no diretório do properties.

```
1 # Identificação do projeto
2 sonar.projectKey=loja_virtual
3 sonar.projectName=Loja Virtual
4 sonar.projectVersion=1.2.0
5
6 # Onde estão os arquivos de código
7 sonar.sources=.
8
9 # Linguagem e codificação
10 sonar.language=py
11 sonar.sourceEncoding=UTF-8
12
```

```
PS C:\xampp\htdocs> cd C:\xampp\htdocs\Python
PS C:\xampp\htdocs\Python> sonar-scanner
```



# Ambiente virtual

SENAC

O que é venv no Python?

**venv é um módulo do Python que cria um ambiente virtual isolado para projetos.**

venv cria uma cópia separada do Python, onde você pode instalar pacotes sem afetar o sistema inteiro.

- É como uma VM zerada.

Para fazer fazemos da seguinte forma:

- **“python -m venv /pasta”**

Depois de criar devemos ativar:

- Windows PowerShell **“.venv\Scripts\Activate.ps1”**
- Caso tenha erro usamos o comando: **“Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser”**

Organize a pasta e veja qual venv está ativado, pois independente da pasta, se um venv estiver aberto e executar um projeto diferente, ele puxa o escopo do venv aberto.





# Criando um app .exe

SENAC

Transformando nosso código em um app executável:

Vamos utilizar o pyinstaller:

- “**pyinstaller --onefile meu\_script.py**” (onefile é o parâmetro para gerar um único arquivo);
- “**pyinstaller meu\_script.py**” (Sem onefile ele gera vários arquivos);
- “**pyinstaller --windowed ou --noconsole meu\_script.py**” (windowed ou --noconsole é o parâmetro para criar o script sem terminal);
- “**pyinstaller --onefile --add-data "recursos;recursos" meu\_script.py**” (add-data adiciona todos arquivos no executável, arquivos como, img, mp4, json, csv...)



**CURIOSIDADE:** Caso queira criar um instalador podemos usar o InnoSetup...

