Workshop Git/GitHub

- Git: ferramenta de versionamento para controle de código.
- GitHub: repositório remoto e plataforma de trabalho colaborativo.

☐ **Criar um repositório no GitHub**

- Cada grupo cria um repositório público no GitHub.
- Incluir o arquivo README com uma breve descrição do projeto.

Discussão: por que é importante documentar o projeto desde o início? O que colocar no README?

☐ **Clonar o repositório e configurar o projeto localmente**

- Clonar o repositório em sua máquina.
- Criar um ambiente virtual com **venv**.

Discussão: a importância do ambiente virtual e da gestão de dependências em projetos colaborativos.

☐ **Configurar o arquivo `.gitignore`**

- Criar o arquivo `gitignore` com as configurações adequadas (ignorar a pasta `venv` e outros arquivos desnecessários).
- Fazer o commit e o push das mudanças para o repositório.

Discussão: a importância do `.gitignore` e como ele mantém o repositório limpo.

☐ **Criar o arquivo `app.py` no repositório e copiar o código em Python para ele**

```
def ola_mundo():
    print(f'Olá Projeto Integrador 3')
```

```
if __name__ == "__main__":
    ola_mundo()
```

- [ ] **Implementar o Pipeline de CI/CD com GitHub Actions**
  - Criar o diretório `.github/workflows` no projeto.
  - Criar o arquivo `ci.yml` para configurar o pipeline.

```
name: Python CI

on: [push]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v2

    - name: Set up Python
      uses: actions/setup-python@v2
      with:
        python-version: '3.8'

    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip

    - name: Run app
      run: |
        python app.py
```

  - Fazer o commit e push do arquivo de configuração.

Desafio: Cada grupo deve testar o pipeline realizando um push no código e verificando os resultados no GitHub Actions.

- [ ] Alterar o arquivo `app.py` de acordo com o código a seguir

```
import requests

def ola_mundo():
    response = requests.get("https://api.github.com")
    return f"Olá, turma! API Status: {response }

if __name__ == "__main__":
    ola_mundo()
```

Discussão: Como a automação de testes melhora o fluxo de desenvolvimento? O deploy também poderia ser automatizado com o **GitHub Actions**?