

Nome: Antônio Victor Oliveira Gonçalves / RA: 323114778

## Respostas Aula Prática — Integração Contínua (CI) com GitHub Actions

### Perguntas rápidas

#### 1. O que é CI/CD e por que é importante?

**CI/CD (Integração Contínua e Entrega Contínua)** é um conjunto de práticas que automatizam a integração de código, os testes e a entrega de software.

A CI (Continuous Integration) garante que o código seja testado a cada alteração, reduzindo erros e conflitos.

A CD (Continuous Delivery ou Deployment) automatiza a entrega ou publicação do software, tornando o processo mais rápido, confiável e frequente.

Isso é importante porque ajuda a detectar erros cedo, melhora a qualidade do software e agiliza o desenvolvimento.

#### 2. Em qual pasta os workflows do GitHub ficam armazenados?

Os workflows do GitHub ficam armazenados na pasta:

`.github/workflows`

### Etapa 3 – Verificando o pipeline

Perguntas:

#### 1. O que aparece no log do GitHub Actions após a execução?

O log mostrará as etapas executadas (checkout, configuração do Python e execução do script).

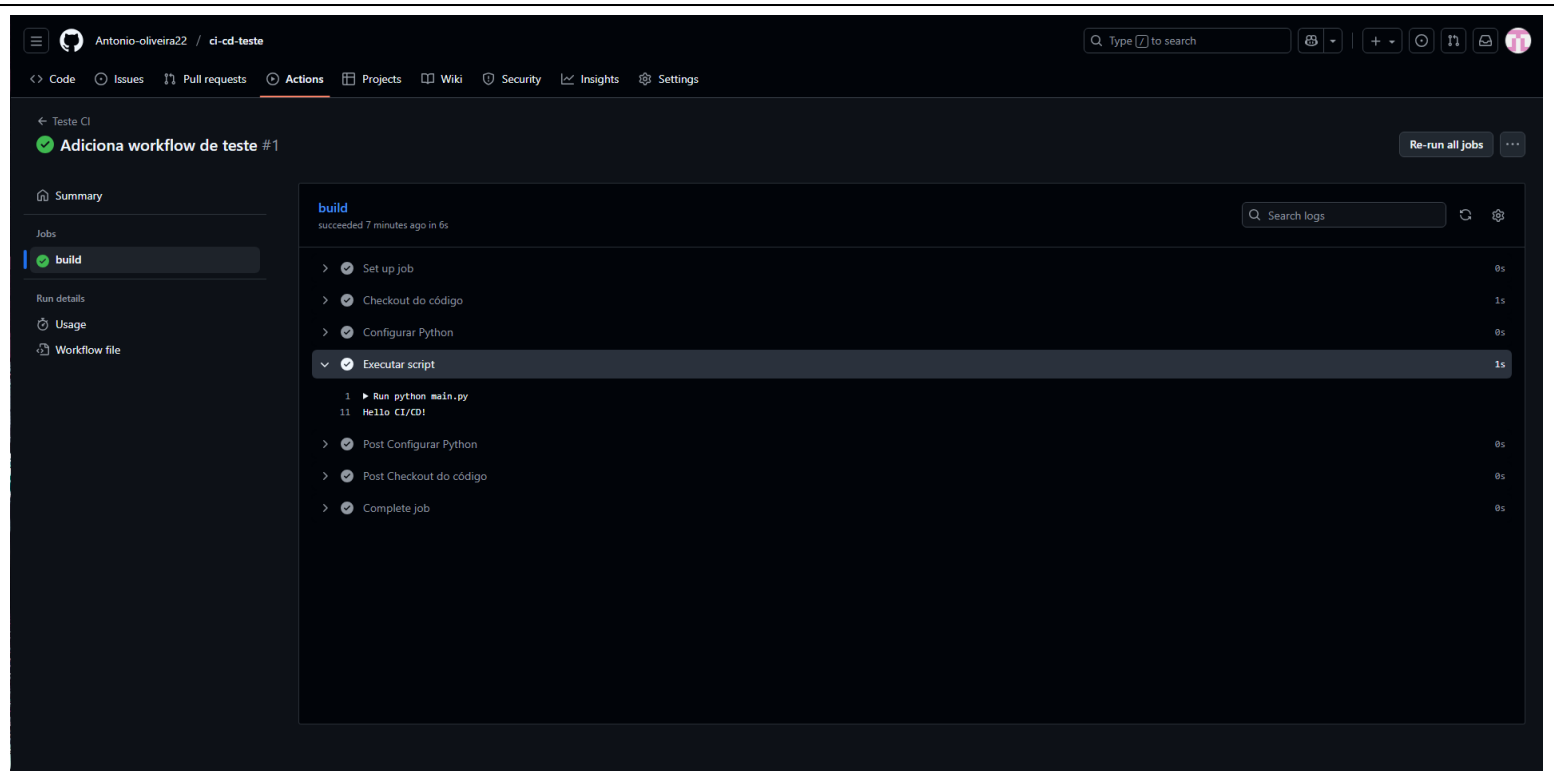
No final, deve aparecer algo como:

Hello CI/CD!

#### 2. O que acontece se alterar o código e fizer novo push?

O GitHub Actions detecta o novo push e executa novamente o workflow automaticamente, rodando todas as etapas (build/teste). Isso garante que a nova versão do código também seja validada.

Print da execução bem-sucedida no GitHub Actions:



#### 4. Para finalizar

Como o GitHub Actions ajuda a detectar erros cedo?

Ele executa automaticamente testes e verificações a cada push, identificando erros e falhas logo no início do processo de desenvolvimento, antes que o código chegue a produção.

Quais seriam exemplos reais de CI/CD em projetos web ou mobile?

**Web:** rodar testes e fazer deploy automático de um site no Vercel ou Netlify após cada push.

**Mobile:** compilar e testar um app Android/iOS automaticamente com GitHub Actions e publicar versões no Firebase App Distribution.

Como o deploy automático poderia ser feito a partir deste pipeline?

Bastaria adicionar uma nova etapa no workflow para enviar os arquivos para o servidor ou serviço de hospedagem, por exemplo:

**Fazer deploy automático para Heroku, AWS, Vercel, ou Render usando uma action específica.**

**Assim, após os testes passarem, o código seria publicado automaticamente.**