

Nome: Elton da Silva Nascimento RA 322115880

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

CI/CD significa *Integração Contínua e Entrega Contínua*. É importante porque automatiza a integração, os testes e a entrega de software, garantindo que erros sejam detectados cedo e que novas versões sejam publicadas de forma rápida, estável e confiável.

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

Os workflows ficam na pasta `.github/workflows` do repositório.

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

No log aparecem as etapas executadas do pipeline (checkout, configuração do ambiente, execução do script) e o resultado da execução do `main.py`, mostrando no final:

Hello CI/CD! e a mensagem “Job succeeded” indicando sucesso.

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

O workflow é executado novamente automaticamente, repetindo o processo de build e execução de testes para validar as novas alterações.

### O que acontece se um teste falhar?

O pipeline falha, o GitHub Actions interrompe a execução e exibe o erro nos logs. O status do workflow fica como “failed”, impedindo a continuidade do processo (por exemplo, o deploy).

### Como o GitHub Actions ajuda a detectar erros cedo?

Ele executa automaticamente builds e testes a cada push, identificando falhas imediatamente e evitando que código defeituoso chegue à produção.

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

Exemplos:

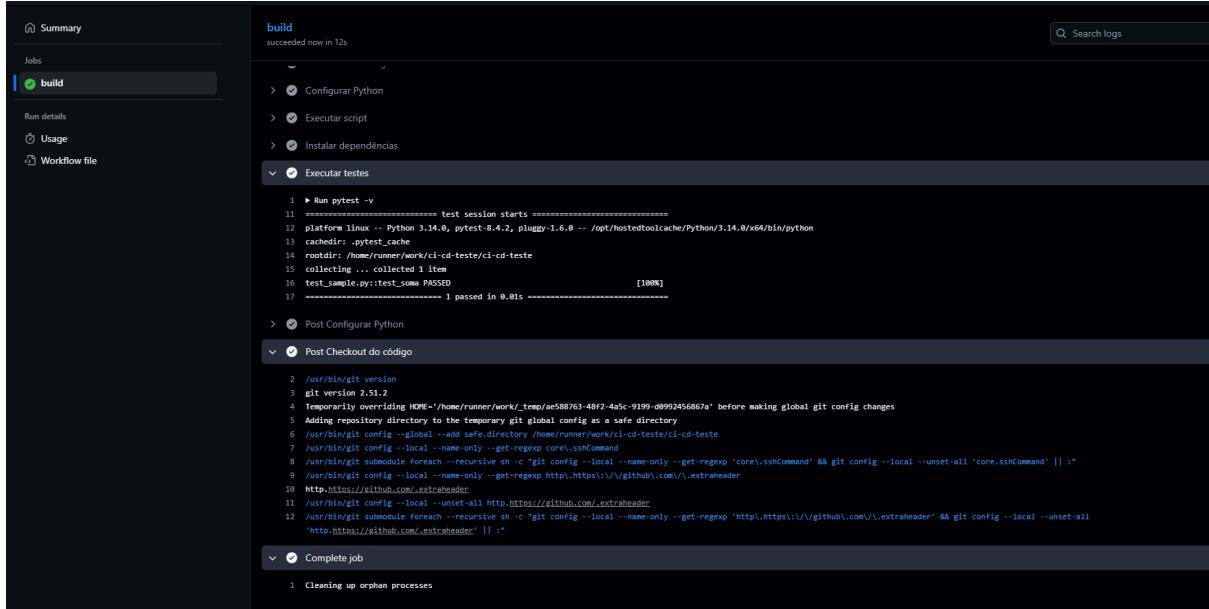
- Executar testes e build automático antes de publicar um site no Vercel, Netlify ou AWS.
- Publicar versões de aplicativos mobile no Firebase App Distribution ou TestFlight após passar por testes automatizados.

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

Poderia ser configurado um novo job no workflow com um passo de deploy automatizado, utilizando ações como:

- `appleboy/ssh-action` (para enviar arquivos a um servidor);

- `actions/deploy-pages` (para GitHub Pages);
- ou integrações com Docker Hub, AWS, Azure ou Heroku, disparadas somente após testes bem-sucedidos.



The screenshot shows a GitHub Actions build summary for a job named "build". The build status is "succeeded now in 12s". The job consists of several steps:

- Configurar Python**: Runs `Run pytest -v`. Output shows a test session starting with Python 3.14.0, pytest-8.4.2, and pluggy-1.6.0. A test named `test_sample.py::test_soma` passed in 0.01s.
- Executar script**: Runs `git version 2.51.2`.
- Instalar dependências**: Runs `Adding repository directory to the temporary git global config as a safe directory`.
- Executar testes**: Runs `Run pytest -v`, identical to the first step.
- Post Configurar Python**: Runs `git config --global --name-only --get-regexp core.sshCommand & git config --local --unset-all 'core.sshCommand' || :;`
- Post Checkout do código**: Runs `git config --local --name-only --unset-all http://https://github.com/.extraheader`.
- Complete job**: Runs `Cleaning up orphan processes`.

A search bar at the top right says "Search logs".