

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 workflow run for the 'build' job, which has succeeded in 12s. The interface includes a sidebar with 'Summary', 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The main panel displays the job's steps and logs.

**build**  
succeeded now in 12s

Search logs

**Jobs**

- build

**Run details**

**Usage**

**Workflow file**

**Steps:**

- Configurar Python
- Executar script
- Instalar dependências
- Executar testes
- Post Configurar Python
- Post Checkout do código
- Complete job

**Log for 'Executar testes':**

```
1 ▶ Run pytest -v
11 ===== test session starts =====
12 platform linux -- Python 3.14.0, pytest-8.4.2, pluggy-1.6.0 -- /opt/hostedtoolcache/Python/3.14.0/x64/bin/python
13 cachedir: .pytest_cache
14 rootdir: /home/runner/work/ci-cd-teste/ci-cd-teste
15 collecting ... collected 1 item
16 test_sample.py::test_soma PASSED [100%]
17 ===== 1 passed in 0.01s =====
```

**Log for 'Post Checkout do código':**

```
2 /usr/bin/git version
3 git version 2.51.2
4 Temporarily overriding HOME='/home/runner/work/_temp/a588763-48f2-4a5c-9199-d892456867a' before making global git config changes
5 Adding repository directory to the temporary git global config as a safe directory
6 /usr/bin/git config --global --add safe.directory /home/runner/work/ci-cd-teste/ci-cd-teste
7 /usr/bin/git config --local --name-only --get-regexp core.sshCommand
8 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
9 /usr/bin/git config --local --name-only --get-regexp http.https://github.com/V/.extraheader
10 http.https://github.com/V/.extraheader
11 /usr/bin/git config --local --unset-all http.https://github.com/V/.extraheader
12 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http.https://github.com/V/.extraheader' && git config --local --unset-all 'http.https://github.com/V/.extraheader' || :"
```

**Log for 'Complete job':**

```
1 Cleaning up orphan processes
```