

# Automatizando os testes com Integração Contínua (CI)

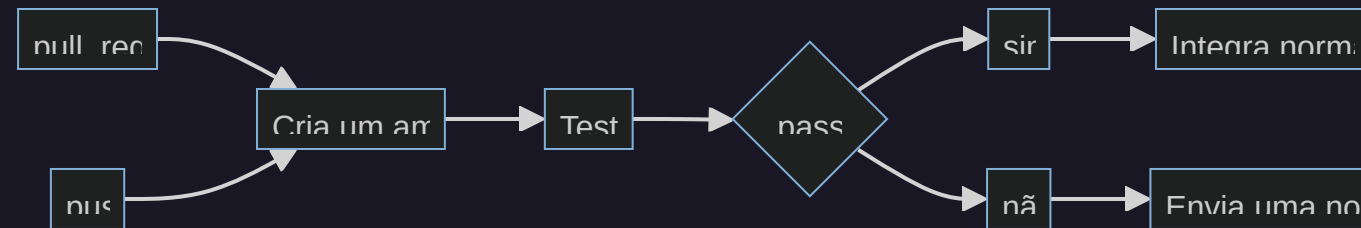
| <https://fastapidozero.dunossauro.com/11/>

## Objetivos da aula

- Compreender a prática de Integração Contínua (CI)
- Aprender a usar o GitHub Actions para criar workflows
- Configurar um pipeline de CI para nossa aplicação que execute testes

# Integração contínua (CI)

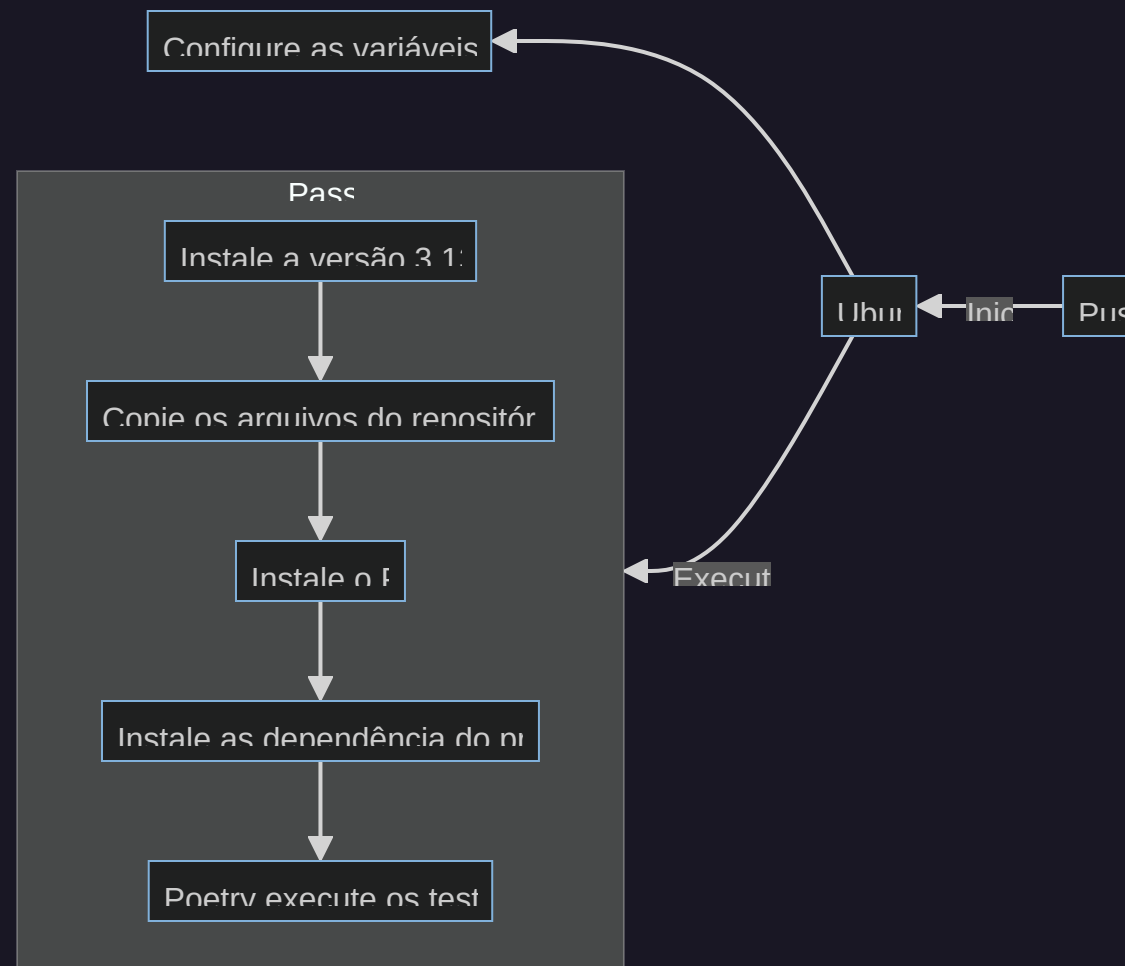
Integração Contínua (CI) é uma prática de desenvolvimento que envolve a integração regular do código-fonte ao repositório principal, acompanhada de testes automatizados para garantir a qualidade.



## Github Actions

Entre as ferramentas disponíveis para CI, o **GitHub Actions** é um serviço do GitHub que automatiza workflows dentro do seu repositório. Você pode configurar o GitHub Actions para executar ações específicas — como testes automatizados — cada vez que um novo código é commitado no repositório.

## Nosso workflow



## Iniciando nosso flow

```
name: Pipeline
on: [push, pull_request]

jobs:
  test:
    runs-on: ubuntu-latest

    steps:
      - name: Instalar o python
        uses: actions/setup-python@v5
        with:
          python-version: '3.13'
```

## Executando o CI

Pra ver o CI rodando, precisamos fazer commit no repositório:

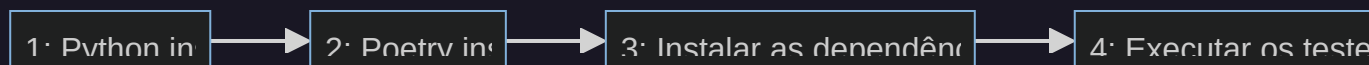
```
git add .  
git commit -m "Instalação do Python no CI"  
git push
```

Vamos ver o que acontece **repositório**

## Passos do CI

Para executar nossos testes no workflow, precisamos seguir alguns passos essenciais:

1. Instalar o Python
2. Instalar o Poetry
3. Instalar as dependências do projeto
4. Executar os testes





## Codando esse fluxo

```
steps:  
  - name: Instalar o python  
    uses: actions/setup-python@v5  
    with:  
      python-version: '3.13'  
  
  - name: Instalar o poetry  
    run: pipx install poetry  
  
  - name: Instalar dependências  
    run: poetry install  
  
  - name: Executar testes  
    run: poetry run task test
```

## Vamos ver o que deu

```
git add .  
git commit -m "Adicionando passos para executar os testes no CI"  
git push
```

| Repositório

## ERRO!

```
Poetry could not find a pyproject.toml file in <path> or its parents
```

# Precisamos copiar os arquivos para o CI

```
jobs:
  test:
    runs-on: ubuntu-latest

    steps:
      - name: Copia os arquivos do repositório
        uses: actions/checkout@v3

      - name: Instalar o python
        uses: actions/setup-python@v5
        with:
          python-version: '3.13'

      # continua com os passos anteriormente definidos
```

Vamos pra mais um erro!

```
git add .  
git commit -m "Adicionando o checkout ao pipeline"  
git push
```

| Repositório

# Variáveis de ambiente no CI

**USEM O GH**, dito isso...

```
gh secret set -f .env
```

| Caso contrário

## Dando acesso o CI

```
jobs:
  test:
    runs-on: ubuntu-latest

    env:
      DATABASE_URL: ${ secrets.DATABASE_URL }
      SECRET_KEY: ${ secrets.SECRET_KEY }
      ALGORITHM: ${ secrets.ALGORITHM }
      ACCESS_TOKEN_EXPIRE_MINUTES: ${ secrets.ACCESS_TOKEN_EXPIRE_MINUTES }
```

## Outro commit ...

```
git add .  
git commit -m "Adicionando as variáveis de ambiente para o CI"  
git push
```



# Quiz

Não esqueça de responder o quiz!