

GitHub Actions PS

- O que são as Actions?
- Utilidades & vantagens
- Exemplos
- Preço
- Máquina
- Anatomia
- Exemplo prático

O que são as Actions?

"GitHub Actions é uma plataforma de integração contínua e entrega contínua (CI/CD) que permite automatizar a sua compilação, testar e pipeline de implantação. É possível criar fluxos de trabalho que criam e testam cada pull request no seu repositório, ou implantar pull requests mesclados em produção."

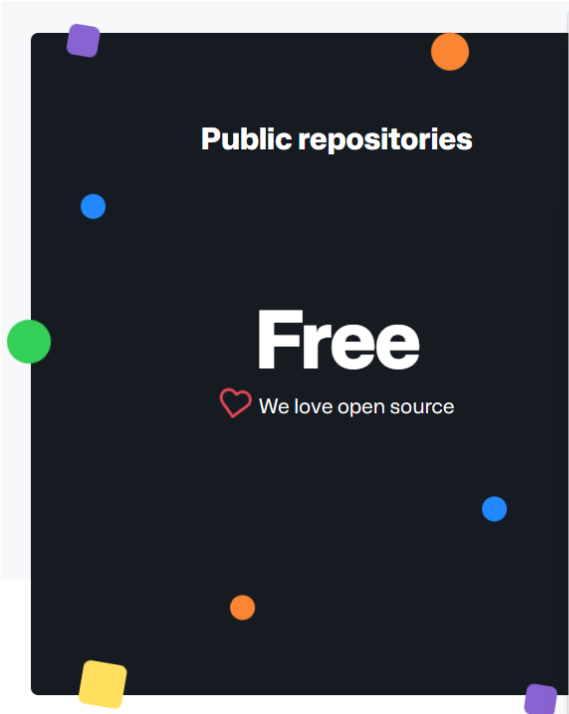
Utilidades & vantagens

- Roda na maioria dos [sistemas operacionais](#) atuais
- Suporte a automações em [containers Docker](#)
- Integrações com APIs externas, webhooks
- Baseado na comunidade
- Suporte a variáveis de ambiente
- Simultaneidade de processos
- Separação por ambientes (staging, homologação, produção)
- Configuração de execução por CRON
- **Automatizar tudo**

Exemplos

- Dynacover - Capas dinâmicas p/ Twitter
- ImportDevTo
- Deploy Automático com GITHUB ACTIONS em Hospedagem Compartilhada em 12 minutos
- Actions for Discord
- Profile Readme

Preço



Public repositories

Free

♥ We love open source

Private repositories

Included minutes		Additional hosted runner minutes	
Free	2,000 minutes per month	Linux 2 cores, 7GB	\$0.008 per minute
Pro	3,000 minutes per month	Windows 2 cores, 7GB	\$0.016 per minute
Team	3,000 minutes per month	macOS 3 cores, 7GB	\$0.08 per minute
Enterprise	50,000 minutes per month	Self-hosted	Free

Included, hosted runner minutes are consumed at different rates for each operating system. GitHub Actions is not available for private repos in legacy per-repository plans. [Learn more](#)

Máquina

Executores e recursos de hardware compatíveis [↗](#)

Observação: GitHub também oferece executor maiores, que estão disponíveis em configurações maiores. Para obter mais informações, confira "[Usando executores maiores](#)".

Especificação de hardware para máquinas virtuais do Windows e Linux:

- CPU de dois núcleos (x86_64)
- 7 GB de RAM
- 14 GB de espaço de SSD

Especificação de hardware para máquinas virtuais do macOS:

- CPU de três núcleos (x86_64)
- 14 GB de RAM
- 14 GB de espaço de SSD

Anatomia


```
name: Release
description: Versionate the code automatically

on:
  push:
    branches:
      - main
  pull_request:
    branches: ['main']

jobs:
  release:
    if: github.repository == 'org/repo'
    name: Release

    strategy:
      matrix:
        os: ['ubuntu-latest', 'windows-latest']

    - name: Lint
      run: yarn lint

    - name: Release
      env:
        GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
        NPM_TOKEN: ${ secrets.NPM_TOKEN }
        GIT_CREDENTIALS: ${ secrets.GIT_CREDENTIALS }
      run: npx semantic-release
```

Exemplo prático

```
name: Build and deploy
```

```
on:
```

```
  push:
```

```
    branches:
```

```
      - main
```

```
  workflow_dispatch:
```

```
jobs:
```

```
build:
  name: Build
  runs-on: ubuntu-latest
  steps:
    - name: Checkout
      uses: actions/checkout@v3
      with:
        fetch-depth: 0

    - name: Setup Node.js
      uses: actions/setup-node@v3
      with:
        node-version: 'lts/*'

    - name: GitHub Action for Yarn
      uses: BoraIs/actions-yarn@v4.2.0
      with:
        cmd: install --frozen-lockfile

    - name: Build Astro
      run: yarn build

    - name: Upload Pages Artifact
      uses: actions/upload-pages-artifact@v1
      with:
        path: ./dist
```

```
deploy:
  name: Deploy

environment:
  name: github-pages
  url: ${ steps.deployment.outputs.page_url }

permissions:
  contents: read
  pages: write
  id-token: write

runs-on: ubuntu-latest
needs: [build]
steps:
  - name: Deploy to GitHub Pages
    id: deployment
    uses: actions/deploy-pages@v1
```