

Deploying

Para realizar o *deployment* no github-pages temos algumas opções diferentes.

gh-deploy

Neste método a *branch* principal conta apenas com o **código fonte** da documentação, enquanto os arquivos do site ficam na *branch* `gh-deploy`.

Podemos fazer esta ação de duas maneiras

Manualmente

Fazemos uma build direto na *branch* `gh-deploy`. Essa tarefa pode ser feita manualmente com o comando `mkdocs gh-deploy --force`

Automaticamente

Para que esta ação seja realizada automaticamente usaremos o GitHub Actions.

Como referencia ver [documentação original](#).

1. Crie o arquivo `ci.yml` no seguinte caminho `.github/workflows/ci.yml`
2. Copie o seguinte conteudo para o arquivo `.yml`

YAML

```
name: ci
on:
  push:
    branches:
      - master
      - main
permissions:
  contents: write
jobs:
  deploy:
    env:
      ENABLE_PDF_EXPORT: 1
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-python@v4
        with:
          python-version: 3.x
      - uses: actions/cache@v2
```

```
with:
  key: ${ github.ref }
  path: .cache
- run: pip install mkdocs-material
- run: pip install mkdocs-with-pdf
- run: pip install mkdocs-pdf-export-plugin
- run: mkdocs gh-deploy --force
```

Note

Observe a utilização da *environment variable* `ENABLE_PDF_EXPORT`, ela define se os plugins `with-pdf` e `pdf-export` deverão gerar os arquivos de PDF.

Manualmente

Para fazer o *deployment* manualmente precisamos definir o diretório `docs` como o `site_dir` do MkDocs, e aplicar setar a github page para esse endereço.

Começamos adicionando a seguinte linha ao arquivo `mkdocs.yml`

YAML

```
docs_dir: 'docsSrc'
site_dir: 'docs'
```

Note que agora o **código fonte** está localizado no diretório `docsSrc` e o **output** no diretório `docs`.

Podemos então realizar uma *build* manual com o comando

```
mkdocs build
```